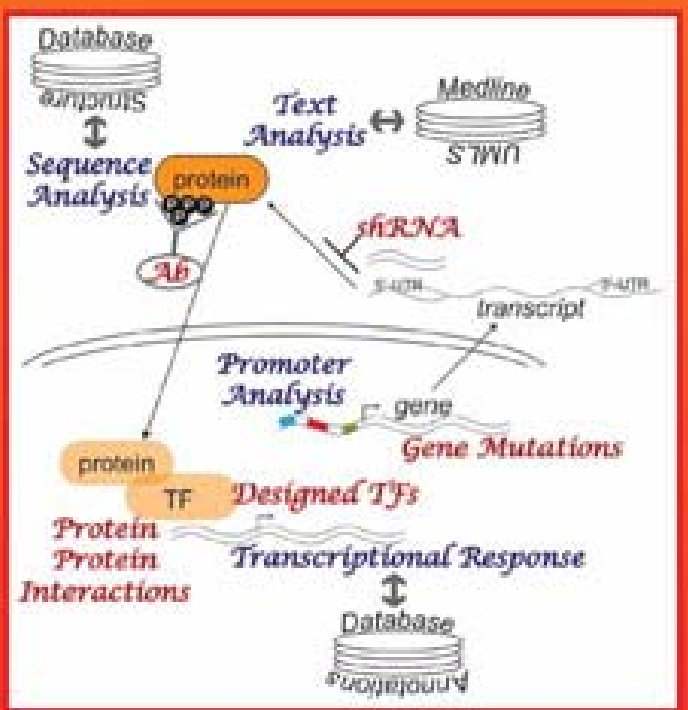
METHODS IN MOLECULAR BIOLOGY™ 408

Gene Function Analysis

Edited by
Michael F. Ochs



HUMANA PRESS

Gene Function Analysis

METHODS IN MOLECULAR BIOLOGY™

John M. Walker, SERIES EDITOR

436. **Avian Influenza Virus,** *edited by Erica Spackman, 2008*
435. **Chromosomal Mutagenesis,** *edited by Greg Davis and Kevin J. Kayser, 2008*
434. **Gene Therapy Protocols:** *Volume 2, Design and Characterization of Gene Transfer Vectors, edited by Joseph M. LeDoux, 2008*
433. **Gene Therapy Protocols:** *Volume 1, Production and In Vivo Applications of Gene Transfer Vectors, edited by Joseph M. LeDoux, 2008*
432. **Organelle Proteomics,** *edited by Delphine Pflieger and Jean Rossier, 2008*
431. **Bacterial Pathogenesis:** *Methods and Protocols, edited by Frank DeLeo and Michael Otto, 2008*
430. **Hematopoietic Stem Cell Protocols,** *edited by Kevin D. Bunting, 2008*
429. **Molecular Beacons:** *Signalling Nucleic Acid Probes, Methods and Protocols, edited by Andreas Marx and Oliver Seitz, 2008*
428. **Clinical Proteomics:** *Methods and Protocols, edited by Antonio Vlahou, 2008*
427. **Plant Embryogenesis,** *edited by Maria Fernanda Suarez and Peter Bozhkov, 2008*
426. **Structural Proteomics:** *High-Throughput Methods, edited by Bostjan Kobe, Mitchell Guss, and Huber Thomas, 2008*
425. **2D PAGE:** *Volume 2, Applications and Protocols, edited by Anton Posch, 2008*
424. **2D PAGE:** *Volume 1, Sample Preparation and Pre-Fractionation, edited by Anton Posch, 2008*
423. **Electroporation Protocols,** *edited by Shulin Li, 2008*
422. **Phylogenomics,** *edited by William J. Murphy, 2008*
421. **Affinity Chromatography:** *Methods and Protocols, Second Edition, edited by Michael Zachariou, 2008*
420. **Drosophila:** *Methods and Protocols, edited by Christian Dahmann, 2008*
419. **Post-Transcriptional Gene Regulation,** edited by *Jeffrey Wilusz, 2008*
418. **Avidin–Biotin Interactions:** *Methods and Applications,* edited by *Robert J. McMahon, 2008*
417. **Tissue Engineering,** Second Edition, edited by *Hannsjörg Hauser and Martin Fussenegger, 2007*
416. **Gene Essentiality:** *Protocols and Bioinformatics,* edited by *Andrei L. Osterman, 2008*
415. **Innate Immunity,** edited by *Jonathan Ewbank and Eric Vivier, 2007*
414. **Apoptosis in Cancer:** *Methods and Protocols,* edited by *Gil Mor and Ayesha Alvero, 2008*
413. **Protein Structure Prediction,** Second Edition, edited by *Mohammed Zaki and Chris Bystroff, 2008*
412. **Neutrophil Methods and Protocols,** edited by *Mark T. Quinn, Frank R. DeLeo, and Gary M. Bokoch, 2007*
411. **Reporter Genes for Mammalian Systems,** edited by *Don Anson, 2007*
410. **Environmental Genomics,** edited by *Cristofre C. Martin, 2007*
409. **Immunoinformatics:** *Predicting Immunogenicity In Silico,* edited by *Darren R. Flower, 2007*
408. **Gene Function Analysis,** edited by *Michael Ochs, 2007*
407. **Stem Cell Assays,** edited by *Vemuri C. Mohan, 2007*
406. **Plant Bioinformatics:** *Methods and Protocols,* edited by *David Edwards, 2007*
405. **Telomerase Inhibition:** *Strategies and Protocols,* edited by *Lucy Andrews and Trygve O. Tollefsbol, 2007*
404. **Topics in Biostatistics,** edited by *Walter T. Ambrosius, 2007*
403. **Patch-Clamp Methods and Protocols,** edited by *Peter Molnar and James J. Hickman, 2007*
402. **PCR Primer Design,** edited by *Anton Yuryev, 2007*
401. **Neuroinformatics,** edited by *Chiquito J. Crasto, 2007*
400. **Methods in Lipid Membranes,** edited by *Alex Dopico, 2007*
399. **Neuroprotection Methods and Protocols,** edited by *Tiziana Borsello, 2007*
398. **Lipid Rafts,** edited by *Thomas J. McIntosh, 2007*
397. **Hedgehog Signaling Protocols,** edited by *Jamila I. Horabin, 2007*
396. **Comparative Genomics,** *Volume 2,* edited by *Nicholas H. Bergman, 2007*
395. **Comparative Genomics,** *Volume 1,* edited by *Nicholas H. Bergman, 2007*
394. **Salmonella:** *Methods and Protocols,* edited by *Heide Schatten and Abe Eisenstark, 2007*
393. **Plant Secondary Metabolites,** edited by *Harinder P. S. Makkar, P. Siddhuraju, and Klaus Becker, 2007*
392. **Molecular Motors:** *Methods and Protocols,* edited by *Ann O. Sperry, 2007*
391. **MRSA Protocols,** edited by *Yinduo Ji, 2007*
390. **Protein Targeting Protocols, Second Edition,** edited by *Mark van der Giezen, 2007*
389. **Pichia Protocols, Second Edition,** edited by *James M. Cregg, 2007*
388. **Baculovirus and Insect Cell Expression Protocols, Second Edition,** edited by *David W. Murhammer, 2007*
387. **Serial Analysis of Gene Expression (SAGE):** *Digital Gene Expression Profiling,* edited by *Kare Lehmann Nielsen, 2007*
386. **Peptide Characterization and Application Protocols,** edited by *Gregg B. Fields, 2007*
385. **Microchip-Based Assay Systems:** *Methods and Applications,* edited by *Pierre N. Floriano, 2007*
384. **Capillary Electrophoresis:** *Methods and Protocols,* edited by *Philippe Schmitt-Kopplin, 2007*
383. **Cancer Genomics and Proteomics:** *Methods and Protocols,* edited by *Paul B. Fisher, 2007*
382. **Microarrays, Second Edition:** *Volume 2, Applications and Data Analysis,* edited by *Jang B. Rampal, 2007*
381. **Microarrays, Second Edition:** *Volume 1, Synthesis Methods,* edited by *Jang B. Rampal, 2007*
380. **Immunological Tolerance:** *Methods and Protocols,* edited by *Paul J. Fairchild, 2007*
379. **Glycovirology Protocols,** edited by *Richard J. Sugrue, 2007*
378. **Monoclonal Antibodies:** *Methods and Protocols,* edited by *Maher Albitar, 2007*
377. **Microarray Data Analysis:** *Methods and Applications,* edited by *Michael J. Korenberg, 2007*
376. **Linkage Disequilibrium and Association Mapping:** *Analysis and Application,* edited by *Andrew R. Collins, 2007*
375. **In Vitro Transcription and Translation Protocols:** *Second Edition,* edited by *Guido Grandi, 2007*
374. **Quantum Dots:** *Applications in Biology,* edited by *Marcel Bruchez and Charles Z. Hotz, 2007*
373. **Pyrosequencing® Protocols,** edited by *Sharon Marsh, 2007*
372. **Mitochondria: Practical Protocols,** edited by *Dario Leister and Johannes Herrmann, 2007*
371. **Biological Aging:** *Methods and Protocols,* edited by *Trygve O. Tollefsbol, 2007*
370. **Adhesion Protein Protocols,** *Second Edition,* edited by *Amanda S. Coutts, 2007*
369. **Electron Microscopy:** *Methods and Protocols, Second Edition,* edited by *John Kuo, 2007*
368. **Cryopreservation and Freeze-Drying Protocols,** *Second Edition,* edited by *John G. Day and Glyn Stacey, 2007*
367. **Mass Spectrometry Data Analysis in Proteomics,** edited by *Rune Matthiesen, 2007*
366. **Cardiac Gene Expression:** *Methods and Protocols,* edited by *Jun Zhang and Gregg Rokosh, 2007*
365. **Protein Phosphatase Protocols:** edited by *Greg Moorhead, 2007*
364. **Macromolecular Crystallography Protocols:** *Volume 2, Structure Determination,* edited by *Sylvie Doublié, 2007*
363. **Macromolecular Crystallography Protocols:** *Volume 1, Preparation and Crystallization of Macromolecules,* edited by *Sylvie Doublié, 2007*
362. **Circadian Rhythms:** *Methods and Protocols,* edited by *Ezio Rosato, 2007*
361. **Target Discovery and Validation Reviews and Protocols:** *Emerging Molecular Targets and Treatment Options, Volume 2,* edited by *Mouldy Sioud, 2007*

METHODS IN MOLECULAR BIOLOGY™

Gene Function Analysis

Edited by

Michael F. Ochs

The Sidney Kimmel Comprehensive Cancer Center,
Johns Hopkins University, Baltimore, MD

HUMANA PRESS TOTOWA, NEW JERSEY

999 Riverview Drive, Suite 208
Totowa, New Jersey 07512

www.humanapress.com

This publication is printed on acid-free paper. ♾
ANSI Z39.48-1984 (American Standards Institute)

Permanence of Paper for Printed Library Materials.
Cover illustration: The cover illustration schematically represents the conversion of a DNA gene to an mRNA transcript and then to a protein, and also shows a protein-protein interaction that leads to a transcriptional response. The computational (blue) and experimental (red) techniques discussed in this book operate at the points indicated. Michael Ochs, 2007

Production Editor: Rhukea J. Hussain
Cover design by Karen Schulz

For additional copies, pricing for bulk purchases, and/or information about other Humana titles, contact Humana at the above address or at any of the following numbers: Tel.: 973-256-1699; Fax: 973-256-8341; E-mail: orders@humanapr.com; or visit our Website: www.humanapress.com

Printed in the United States of America. 10 9 8 7 6 5 4 3 2 1

e-ISBN 13: 978-1-59745-547-3

Library of Congress Control Number: 2007925518

To Ian, you make it fun to be a father.

Preface

This volume of Methods in Molecular Biology focuses on techniques to determine the function of a gene. Traditionally, the function of a gene was determined following cloning, which provided its DNA sequence and an ability to modify this sequence. Experiments were performed that looked for phenotypic changes in a cell line or model organism following modifications to the sequence, knocking out of the gene, or enhancing expression of the gene. In the 1990's, the growing sequence databases and the BLAST algorithm provided additional power by allowing identification of genes with known function that had similar sequences and potentially similar molecular mechanisms. On the experimental side, methods, such as two-hybrid screening that could directly determine the partners of specific proteins and even the domains of interaction, came into widespread use.

With the advent of high-throughput technologies following completion of the human genome project and similar projects in model organisms, the number of genes of interest has expanded and the traditional methods for gene function analysis cannot achieve the throughput necessary for large-scale exploration. Although computational tools such as BLAST remain a good point of departure, it is often the case that a gene that appears interesting in a high-throughput experiment shows no obvious similarity to a gene of known function. In addition, when BLAST does find a similar gene, the process has often only begun. For example, BLAST and family-based derivatives may tell you that a gene of interest is likely to be a kinase, but that does little to tell you its interaction partners or the biological processes in which it plays a role.

This volume brings together a number of techniques that have developed recently for looking at gene function. Computational techniques remain a good point of departure in gene function analysis, as they are inexpensive and can help focus research on those genes that have a high probability of importance. But computational methods can still only predict function, since our knowledge of the detailed biochemical processes that drive cells remains limited in terms of the nonlinear modeling required to predict behavior purely numerically. Computational prediction should therefore always be followed by biochemical and biological techniques to probe the functions of specific targets.

This volume commences, as do most experimental analyses, by looking at computational predictions of gene function. These techniques are divided into two groups, based solely on ease of use. The first set of techniques are straightforward to apply and therefore have moderately low activation energies on the part of the user. The second group are techniques that require moderate programming skills, an ability to create specialized files, or the use of command line interfaces.

The first group of computational techniques (Chapters 1–5) includes methods focused on analysis of gene transcription patterns, promoter analysis, and determination of regions of protein disorder. Microarrays provide global measures of transcriptional output in a number of organisms, and the widespread availability of both data sets and analysis tools make them a logical starting point for gene function analysis. Many of these chapters utilize microarray data for functional inference. In chapter 1, Bidaut provides a method based on the analysis of gene expression data from deletion mutants that permits linking of genes to biological pathways, permitting genes of unknown function to be linked to known pathways. In chapter 2, Kirov and colleagues present an approach using the web-based resource, WebGestalt, to interpret the function of sets of genes through association analysis. In chapter 3, Wang and Ochs analyze microarray data with a modified version of Nonnegative Matrix Factorization to link genes of unknown function to those of known function. In chapter 4, Gonye and colleagues describe a web-based tool, PAINT, that uses promoter analysis to identify gene regulatory networks from microarray data. In chapter 5, Uversky and colleagues describe a web-based tool that predicts protein function by predicting the amount and location of disorder in the structure of a protein.

The second group of computational approaches (Chapters 6–10) includes methods that require greater effort on the part of the reader, but which can also offer greater reward. In chapter 6, Crabtree and colleagues describe the web-based Sybil tool, which allows users to use comparative genomics to identify orthologous sets of genes or proteins, leveraging knowledge from different organisms to predict gene function. In chapter 7, Date uses phylogenetic profiling and the Rosetta stone method to identify functional linkages between proteins, predicting protein interaction partners. In chapter 8, Davuluri provides an approach for predicting the targets of transcription factors and for using this information with ChIP on chip data, linking transcription factors to the genes they regulate. In chapter 9, Osborne and colleagues utilize MetaMap Transfer and the Unified Medical Language System to form relationships between free text in Medline, permitting identification of reported associations between genes. In chapter 10, Ho and colleagues describe an advanced statistical approach to identify genes whose expression is linked, either correlated or anti-correlated, across the conditions in a microarray experiment.

The final portion of this volume (Chapters 11–17) focuses on methods that can experimentally measure and validate gene function, from methods to knock out or reduce the expression of genes, to methods to look for protein interaction partners, and finally to methods that create transcription factors with specialized function. These types of approaches naturally extend the computational methods of the earlier chapters, focused as they are on gene expression, transcriptional regulation, and protein interactions. In chapter 11, Caldwell and

colleagues demonstrate how to use the deletion of genes in the chicken B cell line DT40 to determine gene function. In chapter 12, Zhang and colleagues describe a retroviral-based short hairpin RNA delivery system for knocking down genes in mammalian systems. In chapter 13, Cheng and Chang describe a DNA vector-based short hairpin RNA system for inhibiting gene activities in an inheritable or inducible manner. In chapter 14, Hust and colleagues discuss methods to select antibodies for specific proteins, permitting users to determine where and when proteins are present and active in their systems. In chapters 15 and 16, Tikhmyanova, Serebriiski, and colleagues present two modifications of yeast two-hybrid protein interaction traps that utilize a linked yeast-bacterial approach for refining identification of protein interaction partners. In chapter 17, Thibodeau-Beganny and Joung describe how to use bacterial two-hybrid technology to select Cys2His2 zinc finger domains with specific properties.

Michael F. Ochs

Contents

Contributors

PETER D. ADAMS • *Fox Chase Cancer Center, Philadelphia, PA*
SAMUEL V. ANGIUOLI • *The Institute for Genomic Research, Rockville, MD*
GHISLAIN BIDAUT • *University of Pennsylvania School of Medicine, Philadelphia, PA*
JEAN-MARIE BUERSTEDDE • *GSF, Institute for Molecular Radiobiology, Neuherberg-Munich, Germany*
RANDOLPH B. CALDWELL • *GSF, Institute for Molecular Radiobiology, Neuherberg-Munich, Germany*
PRAVEEN CHAKRAVARTHULA • *Thomas Jefferson University, Philadelphia, PA*
WEN-TSAN CHANG • *National Cheng Kung University Medical College, Tainan, Taiwan*
TSUNG-LIN CHENG • *National Cheng Kung University Medical College, Tainan, Taiwan*
LESLIE COPE • *Johns Hopkins University, Baltimore, MD*
JONATHAN CRABTREE • *The Institute for Genomic Research Rockville, MD*
SHAILESH V. DATE • *University of Pennsylvania School of Medicine, Philadelphia, PA*
RAMANA V. DAVULURI • *OSU Comprehensive Cancer Center, The Ohio State University, Columbus, OH*
MARCEL DETTLING • *Zurcher Hochschule, Winterthur, Swizerland*
STEFAN DÜBEL • *Technical University Braunschweig, Germany*
A. KEITH DUNKER • *School of Medicine, Indiana University, Indianapolis, IN*
PETRA FIEDLER • *GSF, Institute for Molecular Radiobiology, Neuherberg-Munich, Germany*
ERICA A. GOLEMIS • *Fox Chase Cancer Center, Philadelphia, PA*
GREGORY E. GONYE • *Thomas Jefferson University, Philadelphia, PA*
YEN-YI HO • *Johns Hopkins University, Baltimore, MD*
MICHAEL HUST • *Technical University Braunschweig, Germany*
LILIA M. IAKOUCHEVA • *The Rockefeller University New York, NY*
EUGENE A. IZUMCHENKO • *Fox Chase Cancer Center, Philadelphia, PA and Ben Gurion University, Beer Sheva, Israel*
J. KEITH JOUNG • *Massachusetts General Hospital, Charlestown, MA and Department of Pathology, Harvard Medical School, Boston, MA*
WARREN A. KIBBE • *Robert H. Lurie Cancer Center, Northwestern University, Chicago, IL*

STEFAN A. KIROV • *Oak Ridge National Laboratory-University of Tennessee, Oak Ridge, TN*
SIMON LIN • *Robert H. Lurie Cancer Center, Northwestern University, Chicago, IL*
NADIA MILECH • *Telethon Institute for Child Health Research, West Perth, Australia*
ZORAN OBRADOVIC • *Temple University, Philadelphia, PA*
MICHAEL F. OCHS • *The Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University, Baltimore, MD*
JOHN D. OSBORNE • *Robert H. Lurie Cancer Center, Northwestern University, Chicago, IL*
GIOVANNI PARMIGIANI • *Johns Hopkins University, Baltimore, MD*
PREDRAG RADIVOJAC • *School of Informatics, Indiana University, Bloomington, IN*
THOMAS SCHIRRMANN • *Technical University Braunschweig, Germany*
ULRIKE SCHOETZ • *GSF, Institute for Molecular Radiobiology, Neuherberg-Munich, Germany*
JAMES S. SCHWABER • *Thomas Jefferson University, Philadelphia, PA*
ILYA G. SEREBRIISKII • *Fox Chase Cancer Center, Philadelphia, PA*
JAY R. SNODDY • *Vanderbilt University, Nashville, TN*
STACEY THIBODEAU-BEGANNY • *Massachusetts General Hospital, Charlestown, MA*
NADEZHDA Y. TIKHMYANOVA • *Fox Chase Cancer Center, Philadelphia, PA and Drexel University School of Medicine, Philadelphia, PA*
VLADIMIR N. UVERSKY • *School of Medicine, Indiana University, Indianapolis, IN and Russian Academy of Sciences, Moscow Region, Russia*
RAJANIKANTH VADIGEPALLI • *Thomas Jefferson University, Philadelphia, PA*
GUOLI WANG • *Fox Chase Cancer Center, Philadelphia, PA*
OWEN R. WHITE • *The Institute for Genomic Research, Rockville, MD*
JENNIFER R. WORTMAN • *The Institute for Genomic Research Rockville, MD*
XIAOFEN YE • *Fox Chase Cancer Center, Philadelphia, PA*
BING ZHANG • *Vanderbilt University, Nashville, TN*
RUGANG ZHANG • *Fox Chase Cancer Center, Philadelphia, PA*
LIHUA (JULIE) ZHU • *Robert H. Lurie Cancer Center, Northwestern University, Chicago, IL*

I

Computational Methods I

1

Gene Function Inference From Gene Expression of Deletion Mutants

Ghislain Bidaut

Summary

Expression data from knockout mutants is a powerful tool for gene function inference, permitting observation of the phenotype of a deleted gene on the organismal scale. A computational method is demonstrated herein to assess gene function from gene expression measured in deletion mutants using Bayesian decomposition, a matrix factorization technique that permits the extraction of patterns and functional units from the data, i.e., sets of genes belonging to the same pathways shared by sets of knockout mutants. ClutrFree, a cluster visualization program is used to aid in the interpretation of functional units and the assessment of gene functions for a subset of unknown genes.

Key Words: Bayesian decomposition; data dimensionality; gene function discovery; gene-expression analysis; microarray; gene ontologies.

1. Introduction

Assessment of function for genes in yeast *Saccharomyces cerevisiae* is essential for the understanding of molecular function and the annotation of unknown genes in higher eukaryotes ***(1)***. At the date of this writing, more than 60% of genes are reliably annotated in *S. cerevisiae*, which promotes it as one model to compare and model pathways in other genomes. In addition to the genome sequence, a large set of microarray experiments monitoring gene transcription activity under various conditions is available from public repositories. Techniques are now fairly mature, and integrated solutions are provided by microarray makers such as Affymetrix (Santa Clara, CA) and Agilent technologies (Santa Clara, CA), which permit the generation of very reproducible data. On the computing side, microarray management systems and database-based repositories of microarray data permit access, archiving, and tracing of experimental methods.

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs

The critical step in such a high throughput experiment is the process of analysis itself, i.e., converting a high-dimensional data set to a reduced data representation and then to lists of genes of interest. Several techniques (hierarchical clustering, neural networks, and support vector machines, *see* **ref. *2*** for a review) have been proposed, but researchers have not yet reached a consensus on a *de facto* standard to adopt, and this remains an open question. The use of Bayesian decomposition (BD) is demonstrated herein, which has proved to be an effective analysis method to reduce data dimensionality and separate overlapping signals in a variety of data types (spectral data decomposition on chemical shift images ***[2,3]***, microarray gene-expression data such as in the present case, and in the yeast cell cycle ***[4,5]*** and bacterial phylogenetic profiles ***[6]***). In gene-expression analysis, BD has been successfully applied to several data sets in a way that takes the underlying behavior of gene expression into account, i.e., gene products can serve more than one role and therefore genes can be part of multiple functional units.

As an example herein, a deletion mutant data set is analyzed and made available publicly ***(1)***. However, the method can be applied to any large-scale data set containing conditions linked to large changes in the expression of individual genes (e.g., large-scale siRNA studies). Briefly, for this data set, 300 knockout mutants or chemical treatments of yeast *Saccharomyces cerevisiae* were grown in rich media and their gene expression was measured by complementary DNA two-color microarrays. In addition, 63 cultures of wild-type yeast have been grown in rich media to infer the variation of transcription independent of the knockout phenotype in order to generate a gene-specific error model. The data dimensionality has been estimated before by comparing consistency of results for multiple runs of BD with a variable number of patterns ***(5)***. Tools and techniques used were similar to those described herein.

In this chapter, methods for the inference of gene function using BD are detailed, together with advanced visualization software that simplified exploration and interpretation. The chapter covers the setup of the computing environment, the download of data and annotations, their analysis with *BDrun*, and visualization of the results and graphical interpretation with *ClutrFree* ***(7)***.

2. Material

2.1. Software

The procedure is done under a Unix-type operating system, such as MacOS X (Apple Computers Inc., Cupertino, CA) or Solaris (Sun Microsystems Inc., Palo Alto, CA). Likewise, Linux, a free Unix clone for personal computers, is available from several distributors (RedHat Enterprise [RedHat, RTP NC]), Debian Linux [Debian Project], Ubuntu Linux (Canonical Ltd., Douglas, Isle of Man) or preinstalled on new machines.

The components necessary for the analysis are the following:

- The Sun Java virtual machine runtime environment, available from Sun Microsystems (Palo Alto, CA) (http://www.sun.com), which allows to run the tools used in this chapter. This is often installed by default on new computers.
- The BD program (*BDrun*), part of the *BDtools* package available from the Fox Chase Cancer Center Bioinformatics website ***(8)***, under the form of an archive BDtools.tar.gz. This package is extracted with the following command line, which creates a directory "BDtools" containing binaries and documentation:
  $ tar xzf BDtools.tar.gz.
- The ClutrFree program is also available from the Fox Chase Cancer Center Bioinformatics website ***(9)***. This program is available as an executable jar file-clutrfree.jar. To install, download it and save it in the desired location (e.g., /home/ghbidaut/clutrfree/clutrfree.jar). Documentation is available as a pdf file from the same website.

2.2. Data Set and Annotations

2.2.1. Filtered Data Set

The filtered reduced data set is available as a tar archive from the supporting website of this chapter (*see* **ref. *10***).

To extract the archive, the following steps must be executed:

1. The archive "filtered_rosetta_dataset.tar.gz" must be downloaded from the supporting website to the hard drive.
2. The archive is extracted by the following command line:
   $ tar xvf filtered_rosetta_dataset.tar.gz.
3. This creates two file: Fr764_228_ratio.txt and Fr764_228_ratio.unc.

The .txt file contains the gene-expression ratios of experiment over control (this is not a log ratio). The .unc file (at the same format) contains the corresponding uncertainties derived from the gene-specific error model. This is a reduced version of the original data set based on gene variation across experiment: genes showing a variation of at least threefold across experiments, and experiments characterized by a variation of twofold across at least two of the remaining genes were retained, leaving a total of 764 genes across 228 conditions. The file format respects the standard American Standard Code for Information Interchange (ASCII) tab-delimited format used in many analysis packages. Each row represents a gene transcriptional profile across mutants. The format is detailed in **Table 1**.

2.2.2. Annotation of Genes and Conditions

- For gene annotation, the MIPS ontologies (Munich Information Center for Protein Sequences, Munich, Germany) are being used. They are accessible through their website, and a Perl script (automips.pl) to retrieve and format them is provided. The script is downloadable from the supporting website and is invoked using the following command line:

```
$ auto_mips list_of_genes.txt -o annotation.txt.
```

Table 1
Input File Format Used by BDrun

Gene name	Mutant1	Mutant2	Mutant3
Gene_1	Expression_value_1_1	Expression_value_1_2	Expression_value_1_3
Gene_2	Expression_value_2_1	Expression_value_2_2	Expression_value_2_3
Gene_3	Expression_value_3_1	Expression_value_3_2	Expression_value_3_3

Values are tab-delimited. Expression values is a generic term and may be an absolute expression value, or a ratio of experiment/control. Log values are not acceptable as BD performs the factorization on positive, additive distributions. The uncertainties file is the exact same format.

A snapshot of the annotations (April 2005) is also available from the supporting website (annot.txt).

- The list of experiments (conditions in the Rosetta data set) must be supplied for later visualization in ClutrFree. The list is provided on the supporting website as a tab-delimited file (expnames.txt).

3. Methods

First, the approach in **Subheading 3.1.** is discussed. Then pattern recognition in **Subheading 3.2.** is performed, followed by visualization, interpretation, and functional analysis in **Subheading 3.3.**

3.1. Introduction to the BD Algorithm

The BD algorithm is a matrix factorization algorithm that retrieves simultaneously two matrices **A** and **P**, which when multiplied together, reconstruct the expression data **D** under the noise ε:

$$\mathbf{D} = \mathbf{AP} + \varepsilon$$

D is the gene-expression data matrix, and **P** a set of basic vectors in which the data is projected. The **A** matrix is a set of coefficients that allows the reconstruction of **D** through multiplication of **A** and **P**, i.e., the contribution of each basic vector to each gene **(Fig. 1)**. For more details on the underlying mathematics (*see* **ref.** ***11***). Briefly, BD is a Gibbs Sampler that samples the solution space using an atomic prior ***(12)*** and minimizes the χ^2 distance between data **D** and model **A**·**P**. The algorithm operates in two stages: first, the burn-in stage, during which the Markov chain reaches an area of high probability and equilibrates. The second stage is the sampling stage, during which samples are taken to construct a distribution for **A** and **P** elements, leading to a measure of mean and standard deviation for each element.

3.1.1. Application to the Rosetta Compendium

The two matrices **P** and **A** generated by BD from the Rosetta Compendium contain, respectively, a series of patterns and the distribution of those patterns

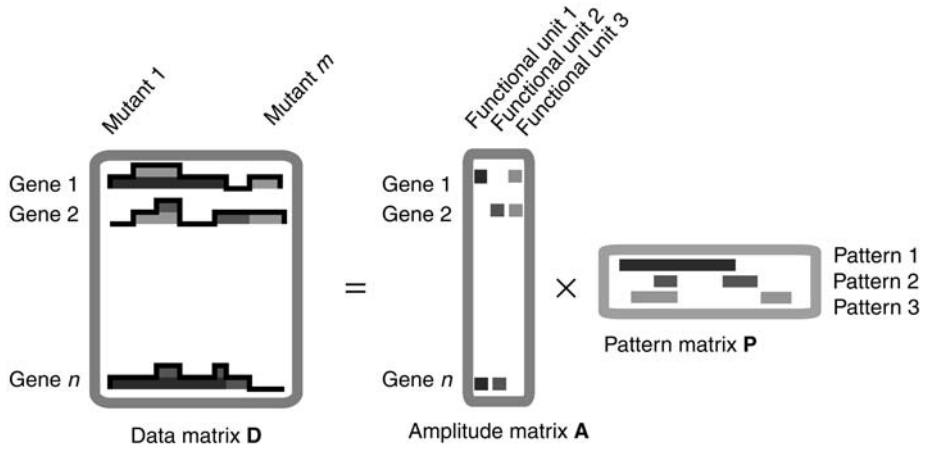


Fig. 1. The data matrix D is decomposed into a pattern matrix **P** and a distribution matrix **A**. These are recombined by multiplication to reconstruct the data matrix **D**. BD generates matrices with elements that are defined by an additive and positive distribution.

in the compendium. Owing to the nature of the prior, BD finds positive additive patterns that are physiologically significant and that can be interpreted in the following way: each pattern describes mutants sharing groups of genes, so-called functional units. Examination of the pattern matrix **P** by row gives the distribution of mutants for a given functional unit, and by columns, how a given mutant is distributed across functional unit. Examination of the distribution matrix **A** by row gives the distribution of a given gene in functional units, and by column, the gene content of each functional unit **(Fig. 1)**. Functional units are groups of genes related to a pattern, and shared by one or more mutant. Genes grouped are part of the same pathway or group of pathways, and genes can be part of several functional units, matching biological borrowing of function (*see* **ref. *13*** for a review of gene function sharing mechanisms in bacteria).

3.1.2. Issue of Dimensionality

The main parameter to set when running BD is the number of patterns to find in the data ***(5)***. Therefore, several BD runs increasing the number of patterns and observing the hierarchical splitting of pathway are performed. When the number of patterns increases, functional units (**A** matrix) grouping several pathways will split into those different pathways, and the patterns that those units relate to will also split (**P** matrix). As the number of patterns reaches 16 ***(5)***, the patterns lose consistency and have a low correlation with the patterns at 15 dimensions, so the optimal number of patterns is predicted as 15. This is because of an increase in degrees of freedom as the number of dimensions needed to explain the data is exceeded.

Table 2
Folder Layout to Use With the Visualization Program ClutrFree

exp_5/	Folder containing the BD result for 5 patterns: the "Fr764_228_ratio.bdo" and "Fr764_228_ratio.gnm" files generated by BDrun for five patterns
exp_6/	Folder containing the BD result for 6 patterns: the "Fr764_228_ratio.bdo" and "Fr764_228_ratio.gnm" files generated by BDrun for six patterns
exp_20/	Folder containing the BD result for 20 patterns
annot.txt	Tab-delimited file with gene annotations
expnames.txt	Tab-delimited file with mutant names

3.2. Applying BD to the Rosetta Compendium With BDrun

3.2.1. Organizing the Data Files

To display the decomposition results with ClutrFree, the data needs to be properly organized at two levels: the BD results level and the gene and experiments (conditions) level. Because the variable that is changed between each BD run is the dimensionality, each BD experiment is to be stored in a separate folder called "exp_dim" ("dim" being the number of patterns for the current experiment, for example, "exp_05" contains the decomposition result for five patterns). ClutrFree can handle arbitrarily named folders but this scheme is followed for clarity and to facilitate the writing of other visualization programs. Other files that have to be included are the condition annotations (the list of knockout mutant names) and the gene annotations generated from the MIPS website. The final file layout is a folder containing the files in **Table 2**.

3.2.2. BDrun and Parameters Selection

BD is run from *BDrun*, a Java graphics interface that permits the specification of parameters and data loading. The program is started by double-clicking on its icon under MacOS X, or by the following command lines:

```
$ cd /path_to_bdtools
$ java -jar BDrun.jar.
```

This brings up the BDrun interface and starts the BDserver computational engine. The BDrun window is organized in three parts **(Fig. 2)**. The right panel contains a series of fields that allows for fixing the parameters (reasonable defaults are provided by the program). The left panel displays messages given during the Markov chain progression and permits the monitoring of the annealing and sampling period by displaying the evolution of the number of atoms ***(12)*** and χ^2 values. The bottom panel permits the operations of loading input files and running the algorithm. Following is the detailed step-by-step procedure to run the analysis.

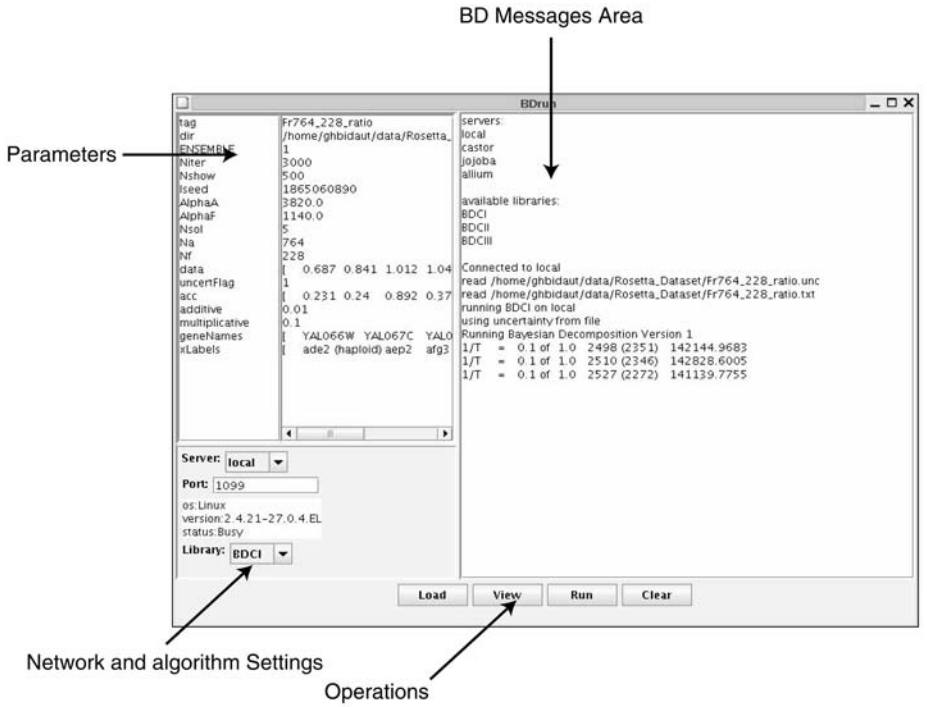


Fig. 2. The main BDrun window, with the parameters to be set on the left and the message window on the right. The sets of parameters on the bottom left allows to run BD on a remote machine, and to select a variant of the algorithm. BD is controlled by the set of buttons at the bottom (operations).

1. The data must be loaded by using the [Load] button. The file Fr764_228_ratio.txt must be chosen and the uncertainty file Fr764_228_ratio.unc must be present in the same directory. If successful, the left textual display gives the message:
   read:/home/ghbidaut/data/Rosetta_Compendium/Fr764_228_ratio.unc
   read:/home/ghbidaut/data/Rosetta_Compendium/Fr764_228_ratio.txt.
   Next, the following parameters need to be reviewed and fixed:
   - tag: this is the current experiment tag, used to identify the output.
   - dir: this is the current working directory.
   - Ensemble: this is the number of Markov chains used simultaneously for searching a solution. For now, the current implementation of BD is limiting this value to one so this cannot be changed by the user.
   - Niter: the number of Markov chain iterations. 3000 is a reasonable value in this analysis.
   - Nshow: this provides the number of steps of the Markov chain between snapshots during sampling. The default value (500) is left as is.
   - Iseed: this is the random seed that will determine the position of the Markov Chain in the solution space. Saving this seed value allows the exact reproducibility of the decomposition.

- alphaA: this is the number of atoms initially present in the atomic domain corresponding to the **A** matrix. BDrun estimates the default number of atoms being proportional to the data set size and the number of patterns, so this default value will be left as it is.
- alphaF: this is the initial number of atoms initially present in the atomic domain corresponding to the **P** matrix. BDrun estimates the default number of atoms being proportional to the data set size and the number of patterns, so this default value will be left as it is.
- Nsol: this parameter fixes the number of basis vectors (patterns) to be inferred by BD. The analysis will be started by setting it initially at three and increase it to 20 during subsequent runs.
- Na: number of genes (rows) in the data set.
- Nb: number of conditions (columns) in the data set.
- data: the data loaded.
- uncertFlag: this flag is set to 1 when an uncertainty file has been found and loaded
- acc: this is the uncertainty data corresponding to each data value.
- additive: when uncertFlag is set to zero, additive and multiplicative values are used to model noise and calculate uncertainty. This is not used here, so leave this to the default value.
- Multiplicative, *see* additive.
- geneName: the gene IDs found in the data set.
- xLabels: the condition IDs found in the data set.
- Server and Port: this is used to run BD on a remote machine (*see* **Note 1**). This is left to the default values of "local" and "1099."
- Library: this allows one to choose among several BD variants. This is set to "BDCI."

2. Once the parameters have been set, BD is run by clicking on the [run] button. If all parameters have been set up correctly, BDrun should display the following messages:

```
running BCDI on local
using uncertainty from file
Running Bayesian Decomposition Version 1.
```

   The Markov Chain random walk is expensive, and one should expect a running time of about 6 h for five patterns/3000 iterations up to 20 h for 20 patterns/3000 iterations on a 3Ghz Pentium4 Linux machine. It is possible to accelerate this process by running multiple instances of BD on remote machines or on a cluster (*see* **Note 1**). To verify the proper progression of the algorithm, the messages specifically can be monitored to the two stages of annealing and sampling in the BDrun message area:

3. During the annealing period, BDrun will display the following messages:
   Running Bayesian Decomposition

```
1/T = 0.1 of 1.0 942 (179) 134654.6845
1/T = 0.1 of 1.0 911 (162) 134515.2276
…
```

4. During the sampling period, BDrun will display the following messages:

```
Sample = 100 of 3000 1573(405) 107000.3768
Sample = 200 of 3000 1548 (398) 106874.0760
…
```

5. Once the decomposition is over, BDrun displays the following message:

```
Random seed was 1141931207
<Chi-Squared> = 106819.8192
<A Atoms> = 1582.8897
< F Atoms > = 405.1083
log[e]Prob(Data) = 75983.9831
Wrote: /home/ghbidaut/data/Rosetta_Dataset/Fr764_228.bdo.
```

6. Two files have been generated by BDrun: Fr764_228.bdo and Fr764_228.gnm. The .bdo file contains the two matrices **A** and **P** along with their corresponding uncertainties. The .gnm file contains additional information that is not used here (a list of gene names used by the BDviewer program).
7. It is then needed to create an appropriate sub directory to store the analysis result. Here the layout is followed as detailed previously, and will name it "exp_05." On MacOS X, the subdirectory can be created by menus, or by the following command line:

```
$ mkdir exp_05.
```

8. The analysis files have to be moved in sub directory created in the previous step. Drag and Drop is possible under MacOS X, or the following command line can be issued from a terminal in Linux or MacOS X.

```
$ mv Fr764_228.bdo Fr764_228.gnm exp_5.
```

9. Back to BDrun, the analysis can restart for six patterns this time, and repeat the same procedure over again, by changing the "Number of Patterns" value from five to six and click on the [run] button again. Once the decomposition is finished, the files Fr764_228.bdo and Fr764_228.gnm has to be moved to the 6_pat directory. This procedure is repeated for up to 20 patterns to obtain all the necessary data.

3.3. Data Visualization and Analysis

ClutrFree is a tool whose goal is to aid in pattern interpretation and visualization through advanced visualization techniques and elaborated tree graphs *(7)*. It features an algorithm for comparison of several experiments generated by the variation of one or several parameters in a given clustering method. In the context of this study, the variable parameter is the number of patterns *n* inferred by BD, to allow estimation of the data dimensionality. Even though ClutrFree can be used with any clustering algorithm, it is well suited to the exploration of the **A** and **P** matrices derived by BD. The pattern window is related to the **P** matrix and the gene window is related to the **A** matrix. The tree display permits the navigation across multiple clustering experiments and the visualization of patterns, stable and unstable, across the experiments. It is constructed by

computing the maximum correlation between patterns (or between the **A** matrix columns for the membership tree).

ClutrFree has the capability to visualize simultaneously multiple functional units and to permit the comparison of the presence/absence of genes across several functional units. In this section, it will be explored how to launch ClutrFree and verify the proper format and organization of the data. The exploration will then be detailed of functional ontologies associated with each pattern to obtain a general idea of the function. Finally **Subheading 3.3.4.** describes in detail the understanding and assignment of gene functions.

3.3.1. Loading the Data in ClutrFree, Verifying Data Integrity, and Basic ClutrFree Window Organization

1. ClutrFree is launched by the following command line wherein the path argument is the root of the layout detailed in **Subheading 3.2.1.**

   ```
   $ java–jar clutrfree/home/ghbidaut/data/Rosetta_Dataset.
   ```

   The second possibility is to double-click on the ClutrFree icon (in MacOS X), then use the menus [File] [Import Data] and select the directory from the dialog. Again, the directory that must be selected is the layout root (Rosetta_Dataset). Once the input directory is selected, ClutrFree displays a progress bar giving feedback on the reading of the data.
2. If everything is properly loaded, a dialog box with the following message is displayed:
   "ClutrFree has successfully loaded 16 experiments. The current displayed experiment has five clusters of length 228." This gives the general topology of the loaded data, the number of experiments compared (here BD was performed with the number of patterns varying between 5 and 20 giving 16 total experiments), and the number of conditions in each cluster (228 conditions from the original data set). This is consistent with the data size.
3. Once the data is loaded, click "OK" on the dialog box. Two windows appear then: the pattern graphics window (Titled ClutrFree: 1:5) and the pattern tree window on top of it. If the pattern tree window is closed, it can be reopened by the menu sequence [Window] [View The Pattern Tree]. The commands available from this window are shown in **Fig. 1**. In addition to the loading data, the [File] menu permits data importation and exportation (graphics or tab-delimited files). The [View] menu allows switching the display between stem-like graphics or plot-like graphics (useful for temporal series), and the [window] menu allows the display of two other windows, the gene table window and the pattern tree window.
4. A gene window is opened by clicking on the [Gene List] button (as many windows can be opened as the user wishes for ease of exploration and pattern comparison). This permits the comparison of a gene list for correlated patterns at different tree levels.

3.3.2. Exploring Patterns and Functional Units: the ClutrFree GUI

The ClutrFree graphic user interface (GUI) has been designed around two windows: the pattern and the gene table window. A global view of the pattern similarities among experiments is displayed in the two tree windows **(Fig. 3)**.

- *The pattern window:* the pattern window shows the pattern inferred by BD (rows of **P**). Horizontal arrows allow navigation across patterns for the same experiments and vertical arrows allow navigation across experiments. The current pattern is represented by a highlighted node in the pattern tree **(Fig. 3A)**. The displayed graph is the contribution of each mutant to the current pattern, displayed in an arbitrary unit. Other features in the graphs include the display of persistence (this value is proportional to the thickness of the blue box behind each plotted point). Graphics can be exported for publication (*see* **Note 3**).
- *The gene-table window:* this window, organized around two tables, displays the distribution of genes (upper table) and ontologies (lower table) across the different patterns (columns of **A**): on one hand, this is used for functional inference on patterns to assess the different pathways present in a pattern and linked to a set of mutants **(Fig. 3C)**. On the other hand, this is used for gene function inference on the basis of known genes with detailed ontologies present in the pattern. The gene table must be used together with the pattern window to understand which gene is linked to which groups of mutants. In the ontology table (lower table), two values are associated for each ontology in each pattern: enrichment (noted $e[x]$, with x being the pattern number) and p-value (noted $p–v[x]$, calculated from a hypergeometric distribution, *see* **ref. *5*** for the mathematical details).
- *The tree windows:* there are two tree windows in ClutrFree, one related to the patterns (pattern tree), showing the relationship of patterns across dimensionalities, and one related to the gene membership (membership tree), constructed from the columns of **A (Fig. 3B**,**D)**. Each tree level represents the patterns from a single BD run, and the levels are sorted from the lower number of patterns (five at the top) to the higher number of patterns (20 at the bottom). Levels are connected by maximum correlation between nodes to infer stable patterns ***(7)***. More information, including ontological information can be displayed in the trees (*see* **Note 4**).

3.3.3. Inference of Pattern Function

To understand the pattern function, the lower part of the gene-table window will be used, the ontology table. This table permits the assessment of a general idea for the pattern function, i.e., which pathways are present in this pattern. This is an essential step before detailed examination of the list of mutants present in the pattern, and the list of genes having the highest contribution to the pattern.

The procedure for pattern function inference is the following: (*see* **Note 2**).

1. The ontologies for 15 patterns are displayed, the level that is believed to be optimal for analysis (*see* **Subheading 3.1.2.**). This is done by pressing the [NSol+1]

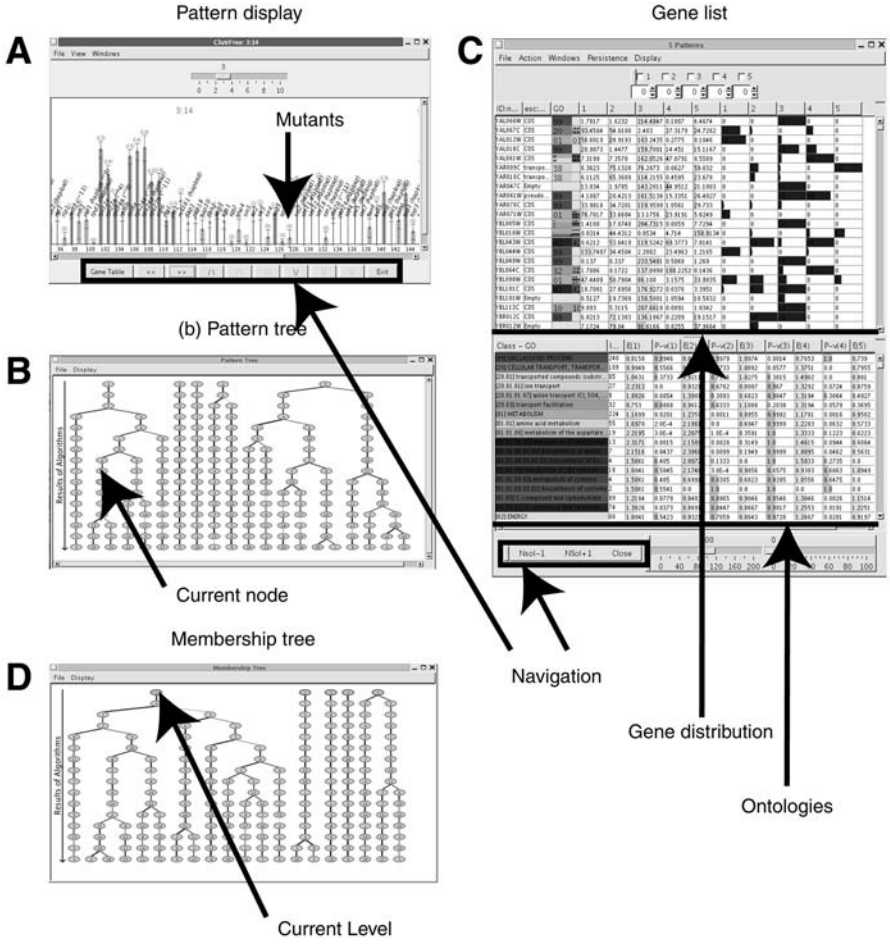


Fig. 3. The complete ClutrFree GUI: in **(A)**, the pattern window showing individual patterns (lines from the **P** matrix) and featuring arrows for navigation. In **(B)**, the pattern tree displaying relationship between patterns across dimensionality. The lowest dimensionality (5) is at the top of the tree, and the highest (20) at the bottom. In this tree, each line represents an experiment. **(C)** The gene list window is divided in two parts: on the upper half, the distribution of genes (**A** matrix) is displayed, each column related to the corresponding pattern in **A**. On the lower half, ontologies enrichment is displayed, and arrows permit navigation between experiments. **(D)** The windows displays the tree built on the **A** matrix columns.

button on the gene table window until the window title displays “Gene Table: 15 patterns.”

2. The filter threshold is set up on the bottom of the gene table (the right slider) to 20. This removes ontologies with 20 or less instances from the ontology table.

3. The most significant ontologies are explored for each pattern by sorting them in order of decreasing enrichment value. This is done by clicking on the column header (alternatively, the ontologies can be sorted by increasing *p*-values by pressing shift+clicking on the column header).
4. The five most enriched ontologies are visualized for all the 15 patterns of the current level. The obtained functions are shown in **Table 3** for patterns 5, 9, 11, 14, and 15 (*see* **Note 2**).

The last step of this procedure is the linkage of mutants present in the pattern of genes of the ontologies found. This permits a finer understanding of the pathways present in the pattern. This is done by examination of the mutants that are *absent* or *present* from the corresponding pattern because those absent mutants are the ones that are affecting the pathways in the functional unit owing to the gene knockout. There is no clearly established cutoff for considering a mutant to be present/absent from a pattern but a rule of thumb is that mutant with an amplitude less than 10% of the maximum amplitude in the pattern can be considered *absent* from it (and correspondingly, a mutant with an amplitude greater or equal to 10% of the maximum amplitude will be considered *present*). *See* **Fig. 4** for an example.

In the following, it is assumed that the files from the website have been loaded into ClutrFree. If BD is rerun, the pattern numbers may change (*see* **Note 2**). The detailed step-by-step procedure is as follows:

1. The Pattern window is brought up and the patterns obtained are displayed with BD set for 15 solutions by pressing the "\/" key as much as necessary until the 15 pattern experiment appears.
2. Pattern 1 is selected by pressing ">>" or "<<" accordingly. The pattern window title must display "ClutrFree: 1:15." It is now displaying pattern 1. Mutant names are collected and characterized by an amplitude lower than 10% of the maximum amplitude. For example, in pattern 14, the only missing mutant is snn6Δ.
3. Once the mutant list is established, each mutant annotation is examined: for pattern 14, all mutants are present, although snn6Δ shows a weaker signal for technical reasons. Now the previous result obtained with the ontologies can be corroborated and the pattern function is confirmed, which is the overall base of processes necessary for survival, so that all mutants have this pattern. For pattern 11, it is found that most mutants are absent. Two mutants with genes of known function, Gas1Δ and Fks1Δ, are present in the pattern. Those mutants (Gas1Δ and Fks1Δ) are known to disrupt cell wall maintenance. Other mutants found in the pattern (Erg2Δ, She4Δ, as well as YER044cΔ) are affecting ergosterol biosynthesis, known for affecting cell wall maintenance.
4. This process is repeated for all patterns by navigating in the pattern window using the arrows keys. Results are summarized in **Table 3** (*see* the "Missing mutants" fields). Additional notes for patterns 9 and 15 are provided (*see* **Note 5**).

Table 3
Some Identified Patterns (5, 9, 11, 14, and 15) and Their Five More Represented Ontologies Present in More Than 20 Genes

Pattern 5		Pattern 9		Pattern 11		Pattern 14		Pattern 15	
Ontology	E(x)	Ontology	E(x)	Ontology	E(x)	Ontology	E(x)	Ontology	E(x)
[18.02.01]: Enzymatic activity regulation/ enzyme regulator	3.9206	[34.11.03.07]: Pheromone response, mating-type determination, sex-specific proteins	5.928	[14.07]: protein modification	3.9696	[14.07]: protein modification	1.4262	[34.11.03.07]: Pheromone response, mating-type determination, sex-specific proteins	8.1617
[18.02]: Target of regulation	3.4306	[34.11.03]: Chemoperception and response	4.1167	[30]:Cellular communication/ signal transduction mechanism	2.7567	[14]:Protein fate (folding, modification, and destination)	1.3275	[34.11.03]: Chemoperception and response	5.5072
[18]:Protein activity regulation	3.0494	[34.11]:Cellular sensing and response	3.9159	[14]:Protein fate (folding, modification, and destination)	2.7304	[20.03]: Transport facilitation	1.2763	[34.11]:Cellular sensing and response	5.2386
[10.03.01]: Mitotic cell cycle and cell cycle control	2.7444	[34]:Interaction with the cellular environment	2.7887	[43.01.03.05]: Budding, cell polarity, and filament formation	2.5446	[11]: Transcription	1.2377	[30]:Cellular communication/ signal transduction mechanism	4.4746

[10.03]:Cell cycle	2.6	[30]:Cellular communication/ signal transduction mechanism	2.5729	[01.03]: Nucleotide metabolism	1.8903	[10.03.01]: Mitotic cell cycle and cell cycle control	1.2318	[34]:Interaction with the cellular environment	3.9839
Present mutants: pll2aΔ, Rpl27aΔ, Rpl34aΔ, Rpl6bΔ, Rp18aΔ, Rps24aΔ, Rps24aΔ (haploid), and Rps27bΔ		Missing mutants: fus3, fus3/kss1, ste11, ste12, and tec1		Present mutants Gas1Δ, Fks1Δ, YER083cΔ, (cell wall) Erg2Δ, She4Δ, YER044cΔ (ergosterol biosynthesis)		Missing mutants: snn6		Missing mutants: fus3, fus3/kss1, ste11, and ste12	
Pattern function: ribosomal proteins		Pattern function: cellular signaling pathway for √filamentation		Pattern function: absence of cell wall maintenance		Pattern function: common processes necessary for survival		Pattern function: cellular signaling pathway for mating	

Row 5 contains the missing mutants found in the pattern, except for pattern 11 wherein present mutant was listed instead, owing to the particularity of the phenotype of the mutant in this pattern (loss of cell wall). The last row contains the general function attributed to the pattern, inferred from absent/present mutants and enriched gene ontologies.

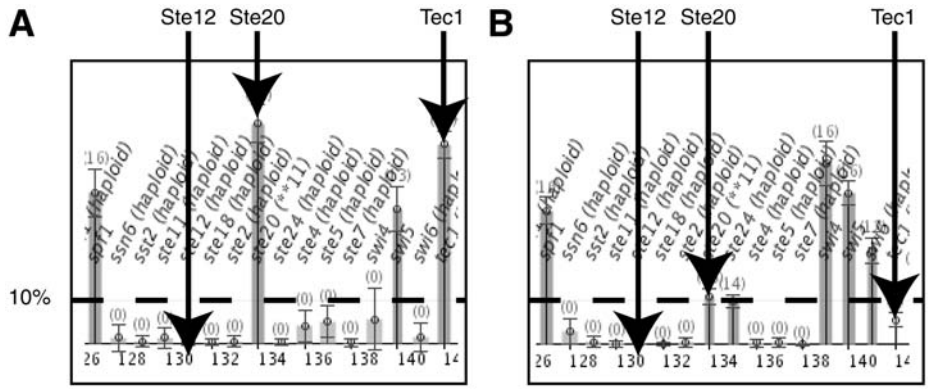


Fig. 4. This is a comparison of patterns 9 and 15, and an illustration of mutant is considered absent or present in the pattern. The contribution is being compared of mutants Ste12Δ, Ste20Δ, and Tec1Δ to patterns 9 and 15: Ste12Δ is considered absent from pattern 15 whereas Ste12Δ, Ste20Δ, and Tec1Δ are all considered absent from pattern 9.

3.3.4. Function Inference for Unknown Genes

Once the pattern function has been confirmed from statistically enriched ontologies and mutants absent from the patterns, (or by statistically depleted ontologies and mutants presents in the pattern), one can use that information to infer functions for individual genes that remain unclassified. This is done by using the information of present/absent mutants from the pattern display. Herein the focus will be on the cell wall patterns (pattern 11) and the ribosomal pattern (pattern 5) already analyzed in **Subheading 3.3.3.**

1. For each explored pattern, a list of mutants is established having a significant contribution to the pattern or that are absent from the pattern.
2. For pattern 11, the following mutants were found to be present: Gas1Δ, Fks1Δ, treatments with tunicamycin and glucosamine (known to disrupt cell wall); and Erg2Δ, She4Δ, and YER044cΔ (linked to ergosterol biosynthesis, so indirectly to cell wall maintenance). A mutant knocked out an unknown ORF: YER083c.
3. For pattern 5, the following mutants were found to be present: Rpll2aΔ, Rpl27aΔ, Rpl34aΔ, Rpl6bΔ, Rp18aΔ, Rps24aΔ, Rps24aΔ (haploid), and Rps27bΔ. Mutants knocking out unknown ORF have also been found (YOR078wΔ, YMR269wΔ, and YHR034cΔ).
4. For each pattern, for which the function has been determined, the pattern function can be attributed to the unknown ORFS in the pattern: for pattern 11, it is conjectured that the gene YER083c is linked to cell wall maintenance. For pattern 5, it is conjectured that the genes YOR078wΔ, YMR269wΔ, and YHR034cΔ are linked to ribosomal functions.

4. Notes

1. BDrun has a distributed network mode wherein a series of nodes running a BDrun server can be set up. A BDrun launched in client mode will send BD jobs to idle nodes on the network that run a BDrun server. This allows running BD on idle machines and leaving the user machine resources free. The user can configure the machines he wishes to use by editing the following file provided with BDrun: /path_to_bdtools/BDservers.list.
2. The pattern indices can be completely different if the analyses are run, because BD assigns patterns a random order number. Also, the values obtained for ontological enhancement may be slightly different than the one shown herein, owing to the gene amplitude values found by BD, and to the ongoing reannotation of the yeast gene ontologies in MIPS.
3. ClutrFree offers the possibility to export pattern graphics (menu [File] [Export Current Graphics] from the ClutrFree main window) or tree graphics for further editing with publishing tools (menu [File] [Export Current Tree] from one of the tree window). Supported formats include joint picture exchange (JPEG), tagged image file format (TIFF), portable network graphics (PNG), simple vector graphics (SVG), and dot for tree exportation. Tree exported in dot formats can be further displayed with the GraphViz package (AT&T Research, Florham Park, NJ, www.graphviz.org).
4. In ClutrFree, trees can be rendered with more information if needed. It is possible to include correlation value (menu [Display] [Correlation]), as well as displaying ontological enrichment in each tree node (menu [Display] [Ontologies]). The display parameters can be set at [Display] [Options].
5. By following the procedure listed in **Subheading 3.3.3.**, functions can be assigned to patterns 9 and 15. They are the two most enriched patterns for the ontology 34.11.03.07: pheromone response, mating-type determination, and sex-specific proteins. To observe which genes are the highest contributors to pattern 9 and 15, they are sorted by value in the upper gene table. For pattern 9, it is observed that 11 of the top 15 genes are transposable elements, involved in filamentation. The hypothesis is that the patterns 9 and 15 represent two reproductive modes in yeast; filamentation and the mating response (*see* **ref. *14***). To confirm this hypothesis, the contribution of tree mutants is examined in interest to those patterns; Ste12Δ, Ste20Δ, and Tec1Δ. In pattern 15, it is found that Ste12Δ is absent, whereas Ste20Δ and Tec1Δ have a strong signal, which is in accordance with the mating pathway activated through Ste12 but which can bypass Ste20 through a G protein complex ***(14)***. In pattern 9, Ste12Δ is still absent, and Ste20Δ and Tec1Δ are showing weak signals. This measurement is in accordance with the set of genes known to trigger filamentation, which includes Ste12, Ste20, and Tec1. *See* **ref. *5*** for a detailed discussion of this issue.

Acknowledgments

The author would like to thank Dr. Michael Ochs, for the development of the BD Method and contribution to *ClutrFree*, and Bill Speier, who is the author of *BDrun*.

References

1. Hughes, T. R., Marton, M. J., Jones, A. R., et al. (2000) Functional discovery through a compendium of expression profiles. *Cell* **102,** 109–126.
2. Ochs, M. F. and Godwin, A. K. (2003) Microarrays in cancer: research and applications. *Biotechniques* **(Suppl),** 4–15.
3. Ochs, M. F., Stoyanova, R. S., Arias-Mendoza, F., and Brown, T. R. (1999) A new method for spectral decomposition using a bilinear Bayesian approach. *J. Magn. Reson*. **137(1),** 161–176.
4. Moloshok, T. D., Klevecz, R. R., Grant, J. D., Manion, F. J., Speier, W. F., IV., and Ochs, M. F. (2002) Application of Bayesian decomposition for analysing microarray data. *Bioinformatics* **18(4),** 566–575.
5. Bidaut, G., Suhre, K., Claverie, J. M., and Ochs, M. F. (2006) Determination of strongly overlapping signaling activity from microarray data. *BMC Bioinformatics* **7,** 99.
6. Bidaut, G., Suhre, K., Claverie, J. M., and Ochs, M. F. (2005) Bayesian decomposition analysis of bacterial phylogenomic profiles. *Am. J. Pharmacogenomics* **5(1),** 63–70.
7. Bidaut, G. and Ochs, M. F. (2004) ClutrFree: cluster tree visualization and interpretation. *Bioinformatics* **20(16),** 2869–2871.
8. Fox Chase Cancer Center bioinformatics Bayesian Decomposition page: http://bioinformatics.fccc.edu/software/Proprietary/bd/bd.shtml.
9. Fox Chase Cancer Center bioinformatics ClutrFree page: http://bioinformatics.fccc.edu/software/OpenSource/ClutrFree/clutrfree.shtml.
10. Supporting website hosted at the Sidney Kimmel Comprehensive Cancer Center at Johns Hopkins. Website: http://www.cancerbiostats.onc.jhmi.edu/supportmmbchapter/index.html.
11. Ochs, M. F. (2003) Bayesian Decomposition, in *The Analysis of Gene Expression Data: Methods and Software*, (Parmigiani, G., Garrett, E., Irizarry, R., and Zeger, S., eds.), Springer Verlag, New York.
12. Sibisi, S. and Skilling, J. (1997) Prior distributions on measure space. *J. Royal Statist. Soc. B* **59(1),** 217–235.
13. Ochman, H., Lawrence, J. G., and Groisman, E. A. (2000) Lateral gene transfer and the nature of bacterial innovation. *Nature* **405,** 299–304.
14. Schwartz, M. A. and Madhani, H. D. (2004) Principles of MAP kinase signaling specificity in *Saccharomyces cerevisiae*. *Annu. Rev. Genet.* **38,** 725–748.

2

Association Analysis for Large-Scale Gene Set Data

Stefan A. Kirov, Bing Zhang, and Jay R. Snoddy

Summary

High-throughput experiments in biology often produce sets of genes of potential interests. Some of those gene sets might be of considerable size. Therefore, computer-assisted analysis is necessary for the biological interpretation of the gene sets, and for creating working hypotheses, which can be tested experimentally. One obvious way to analyze gene set data is to associate the genes with a particular biological feature, for example, a given pathway. Statistical analysis could be used to evaluate if a gene set is truly associated with a feature. Over the past few years many tools that perform such analysis have been created. In this chapter, using WebGestalt as an example, it will be explained in detail how to associate gene sets with functional annotations, pathways, publication records, and protein domains.

Key Words: Association analysis; data interpretation; gene expression; gene set; WebGestalt; genome-scale; high-throughput analysis.

1. Introduction

Because of the first large-scale expression analysis in 1995 ***(1)***, numerous studies have tried to correlate the observed expression patterns with other significant biological data, such as phenotypes, regulatory sequences, pathways, and so on. Such types of correlation analysis could potentially reveal mechanisms that are associated with the observed expression patterns. The results from large-scale biological experiments, such as expression analysis is often complex. In many cases, it will not be possible to infer the aforementioned associations by manual analysis because of the data size and complexity. An overview of the microarray technology and some of the computer-assisted inference analyses is reviewed by Stoughton ***(2)***.

A large number of studies use gene ontology (GO) annotation ***(3)*** to assist in the analysis of gene expression data. For example, Bono et al. used GO to reconstruct metabolic pathways ***(4)***. The GO consortium ***(3)*** provides a powerful

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

way to associate genes with the existing knowledge on some of the genes' major characteristics, such as function and cellular localization. GO has a controlled vocabulary that is understandable to both human and computer, which makes it extremely useful for associative inference analysis. Biocarta and Kyto Encyclopedia of Genes (KEGG) ***(5)*** pathways are also routinely used in expression data analysis. Lin et al. ***(6)*** used GO, Biocarta, and KEGG to identify regulatory networks involved in cancer progression; Kluger et al. ***(7)*** used GO, Biocarta, and KEGG in combination with expression patterns to create a matrix capable of discriminating the developmental choice of hematopoietic cells. An alternative to GO, KEGG, and Biocarta is the PANTHER project ***(8)***, which relies on its own ontologies and pathway data. Other ontologies are in their developmental stage as a part of the Open Biomedical Ontologies (OBO) project and might expand the inference analysis that is described in this chapter. ***(9)***.

Finding other associations, such as transcription factor-binding sites, 3′-UTR signals, and so on, is of very high interest, yet the existing knowledge in this area is too limited for high-throughput analysis. The recent development of different high-throughput techniques such as chip–chip ***(10,11)*** and genome-wide DNase footprinting ***(12,13)*** might lead to the accumulation of the critical volume of data, necessary for transcription factor-binding sites association studies. As gene set interpretation is becoming a critical step in high-throughput biological studies, many bioinformatics tools have been developed for this purpose. **Table 1** lists some common software packages for gene set functional association analysis. Note that this list is not exhaustive. In this chapter, the application of WebGestalt ***(14)*** to the management and association analysis of large-scale gene set data will be illustrated. This analysis usually includes three steps: (1) identifiers (IDs) conversion, (2) gene set management, and (3) gene set analysis. Some distinct functions of other software packages will also be discussed.

2. Materials

Typically, any personal computer (Linux, Mac [Apple, Inc., Cupertino, CA] Windows [Microsoft, Inc., Redmond, WA], and so on) with a recent Internet browser should be sufficient. Certain analyses generate comma-separated files, which are best viewed with a spreadsheet application, such as Open Office, koffice, Microsoft Excel, and so on. PDF reader is required in order to read some of the tool documentation. High-speed Internet connection (such as T1, Digital Subscriber Line [DSL], or cable modem) is highly desirable. The example gene sets consists of four sets: "lymph node," "cerebellum," "cerebrum" ***(15)***, and "brain embryo imprint" ***(16)***. These gene sets can be downloaded from http://bioinfo.vanderbilt.edu/mp/gene_sets. The "lymph node," "cerebellum," and "cerebrum" sets include genes that are overexpressed in corresponding tissues ***(15)***. The "brain embryo imprint" set is

Table 1
Tools That Can Perform Gene Set Functional Association Analysis

Tool name	Web address	Status	Reference
GOStat	http://gostat.wehi.edu.au/	Live	***22***
CLENCH	http://www.personal.psu.edu/faculty/n/h/nhs109/Clench/	Live	***23***
GoToolBox	http://crfb.univ-mrs.fr/GOToolBox/index.php	Live	***24***
DAVID/EASEOnline	http://apps1.niaid.nih.gov/david/	Live	***25***
GoMiner	http://discover.nci.nih.gov/gominer/	Live	***26***
FatiGO	http://www.fatigo.org/	Live	***27***
GOTM/WebGestalt	htttp://genereg.ornl.gov/gotm	Live	***15,28***
Onto-Express	http://vortex.cs.wayne.edu/projects.htm	Live	***29***
GeneMerge	http://www.oeb.harvard.edu/hartl/lab/publications/GeneMerge/GeneMerge.html	Live	***30***
FuncAssociate	http://llama.med.harvard.edu/cgi/func/funcassociate	Live	***31***
GoSurfer	http://biosun1.harvard.edu/complab/gosurfer/	Live	***32***
EGOn	http://www.genetools.no/	Live	***33***
OntologyTraverser	http://franklin.imgen.bcm.tmc.edu/rho/services/OntologyTraverser/	Live	***34***
PANTHER	http://www.pantherdb.org/	Live	***8***
FuncSpec	http://funspec.med.utoronto.ca/	Live	***35***
GoTermFinder	http://bair-server.lesc.doc.ic.ac.uk:8088/cgi-bin/GOTermFinder.pl	Live	***36***
MAPPFinder	http://www.genmapp.org/MAPPFinder.html	Live	***37***
GFINDER	http://promoter.bioing.polimi.it/gfinder/	Live	***38***

based on a study associating expression patterns in the adult mouse brain with the development of the mouse embryo ***(16)***.

Websites of interest and related to this work include:

- WebGestalt: http://genereg.ornl.gov/webgestalt.
- GOTree Machine: http://genereg.ornl.gov/gotm.
- GeneKeyDB: http://genereg.ornl.gov/gkdb or http://sourceforge.net/projects/genekeydb.
- Entrez gene: http://ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene.
- UCSC genome browser: http://genome.ucsc.edu/cgi-bin/hgGateway.
- CGAP: http://cgap.nci.nih.gov/.
- SourceForge: http://www.sf.net.
- Biocarta (Biocarta, Inc., San Diego, CA): http://www.biocarta.com/index.asp.
- KEGG: http://www.genome.ad.jp/kegg/.
- GO consortium: http://www.geneontology.org/.
- OBO: http://obo.sourceforge.net/.
- PANTHER: www.pantherdb.org.
- STRING: http://string.embl.de/.
- DAVID: http://david.abcc.ncifcrf.gov/.
- GSB: http://www.cisreg.ca/gsb/.
- EnsMART: http://www.ensembl.org/Multi/martview.

Additional important sites and mailing lists are described in **Note 1**.

3. Methods

The methods that are described require minimal set of computer skills: using spreadsheet and simple text editing software and accessing the Internet. The steps needed to define a gene set based on expression or other high-throughput experimental design are out of the scope of this chapter and could require high computer/biostatistics proficiency and access to specialized software. As a starting point it is assumed that an interesting gene set (based on experimental or other evidence) is already compiled.

3.1. ID Conversion and Gene Set Management

3.1.1. Gene Set Upload and ID Conversion

Usually commercial microarrays designate the genes printed on the chip using their own IDs. This contributes to the already significant set of existing gene designators. Therefore, often the first step that needs to be undertaken is the conversion of gene IDs. Although it might seem trivial, this step could lead to misleading results if not performed correctly (*see* **Note 2**). It is strongly advised against the use of gene symbols (*see* **Note 3**). It would be best if the input gene IDs are Entrez gene ID (formerly Locuslink ID)

(17). Swissprot ID and Ensembl gene stable ID are also recommended as a second option.

Depending on the technical skills of the user, conversion choices would vary. Installing a local copy of the Ensembl MART database ***(18)***, BioMART (http://www.biomart.org/), or GeneKeyDB ***(19)*** locally, and directly querying the database might be the best choice, but will be much more challenging than using web-based tools, such as EnsMART, WebGestalt, and GSB ***(20)***. Both WebGestalt and GSB provide data set management modules, but GSB has better gene set management functionality. It allows gene set sharing, comment addition, and set synchronization. WebGestalt has analysis modules and a slightly larger choice of chip IDs. WebGestalt uses GeneKeyDB ***(21)*** to find the relevant data, whereas GSB is based on Ensembl. Several tools listed in **Table 1**, such as PANTHER and DAVID, also have integrated gene set conversion and management functionality. If advanced gene set management is important, then either GSB or WebGestalt would be the appropriate choice for this step. Only WebGestalt is described herein whereas an overview for other tools exists at http://bioinfo.vanderbilt.edu/mpaper/tc.html.

1. Register in WebGestalt. This is necessary both to protect and store each users' gene sets. Storage is temporary. Gene sets are erased following each database update, but users are warned by email at least a week before the gene sets are removed. The updates usually occur at 30–60 d periods.
2. Prepare a text file (e.g., in notepad if Windows is used, TextWrangler [Bare Bones Software, Bedford, MA] for MacOS, and Emacs in *nix/Mac OS) with one identifier per line. Do not include a header row. If Microsoft Word is used make sure that in the "Save as type" field one chooses "Plain text." As an alternative one can use the example gene sets (*see* **Subheading 2.**). The description following is based on the example files.
3. Access the gene set uploading/conversion form by selecting "Form file." Fill in the form and press the UPLOAD button. At this point the data set is stored in the database and can be retrieved at a later time (*see* **Note 4** on data set storage).
4. Check for possible inaccuracies, for example:
   - The gene list size in the file should be the same or bigger (in case some genes are not in the supporting database) than the one reported by the conversion tool (*see* **Notes 2** and **3** for explanation).
   - If the number of genes is far smaller than the input, there might be a problem in the upload process.

3.1.2. Gene Set Manipulations

3.1.2.1. Retrieval of Orthologous Gene Sets

Currently WebGestalt works only with human and mouse orthologs. To get the orthologs:

1. Retrieve the gene set uploaded in **Subheading 3.1.1.** To do so login to WebGestalt, select "lymph node" from the pull-down menu next to RETRIEVE button and press RETRIEVE.
2. Press the GET <SPECIES> ORTHOLOGS button. In this case the button would read "get mouse orthologs," because the starting gene set is human.
3. Press SAVE in the new window that opens and follow the normal gene set description step.

3.1.2.2. Boolean Operations: Unions, Intersections, and Differences

Comparing the composition of different gene sets is important in order to compare genes' behavior under different conditions. In this case the overlap of gene sets enriched in cerebellum vs cerebrum will be compared.

Create the intersection of gene set "cerebellum" (A) and gene set "cerebrum" (B):

1. Login to WebGestalt. Unless gene sets "cerebellum" or "cerebrum" have been uploaded previously, upload them at this point. This will make both sets available from the pull-down menus next to BOOLEAN OPERATIONS button.
2. Select gene set A (cerebrum) and B (cerebellum).
3. Select intersection and press BOOLEAN OPERATIONS to create the new gene set.
4. Press SAVE and enter the new set name and description.

3.1.2.3. Boolean Operations for Sets From Different Species

1. Upload organism 1 gene set (set A) following the method in **Subheading 3.1.2.1**.
2. Upload organism 2 gene set (set B) following the method in **Subheading 3.1.2.1**.
3. Using set A follow **steps 1–3** as described in **Subheading 3.1.2.1.**, which will create gene set AB.
4. Create set BA as in **step 3**, using set B instead of A.
5. Check if sets AB and BA have the same number of genes.

3.2. Information Retrieval

The following data can be currently collected in WebGestalt: gene IDs/accession numbers (EntrezGene, RefSeq, UniGene, UniProt, and Ensembl gene stable id), gene names and symbols, cytogenetic and physical mapping data, protein domains, Online Mendelian Inheritance in Man (OMIM), genome reference into function, publications, GO, KEGG, Biocarta, and phenotype.

1. Check the checkboxes denoting the type of data one needs.
2. Press the "Information retrieval" button and save the results (and remember the location of the file). Some browsers are configured to automatically open the results through an Excel plugin (this behavior can be changed through the browser "Helper application" preferences or the operating system (OS) folder settings).

Some fields would contain more than one entry. For example, a gene might have alternative transcripts, in which case there would be more than one RefSeq accession number, which would be separated by three right slashes (///). Another exception is the chromosome coordinates column wherein the fields (chromosome number, start, end, and orientation) are separated with three colons (:::).

3.3. GO Association Analysis (GOTree Machine and the WebGestalt GO Module)

All software packages listed in **Table 1** could analyze associations between gene sets and GO categories. There are several important considerations when choosing the tool that might be most appropriate for the analysis: scope of the analysis (available organisms, GO levels), statistical approach (including reference gene set options), visualization options (interactive vs noninteractive), and update schedule (as almost all tools use database integration). There are other additional features that are not critical for the analysis, but might be of some benefit to the user (*see* the detailed comparison at http://bioinfo.vanderbilt.edu/mpaper/tc.html).

One could use the following steps to accomplish the analysis through the GOTree module in WebGestalt (*see* **Note 5** on the differences between GOTree Machine (GOTM) and WebGestalt GO module and **Note 6** on some restrictions):

1. Retrieve the "brain embryo imprint" gene set.
2. Press the GOTREE button.
3. Choose appropriate reference set and statistical approach (to observe some important considerations on this choice, *see* **Note 7**). In this case, "WEBGESTALT_MG_U74AV2" and "Hypergeometric test." are selected. It may take a few minutes before the calculations are done.
4. Once the analysis process is completed, a new button CHECK GOTREE will appear. After pressing the button, an expandable GOTree will appear, with significantly enriched GO categories highlighted.
5. Press the direct acyclic graph (DAG) button **(Fig. 1)** for an enriched GO DAG, the BAR CHART button for a bar chart at the specified annotation level, or the EXPORT GOTREE button for a text output of the complete tree.
6. To change the GO level and the main tree for a bar chart branch use the pull-down menu under the "Bar chart" button (*see* explanation in **Note 8**).
7. To export the intersection of the input gene set and a specific category follow one of these steps:
   a. *From the bar chart:* press BAR CHART (opens new window), choose an interesting category and press the bar. This opens a new window with genes from the gene set being analyzed that are also in the selected category.

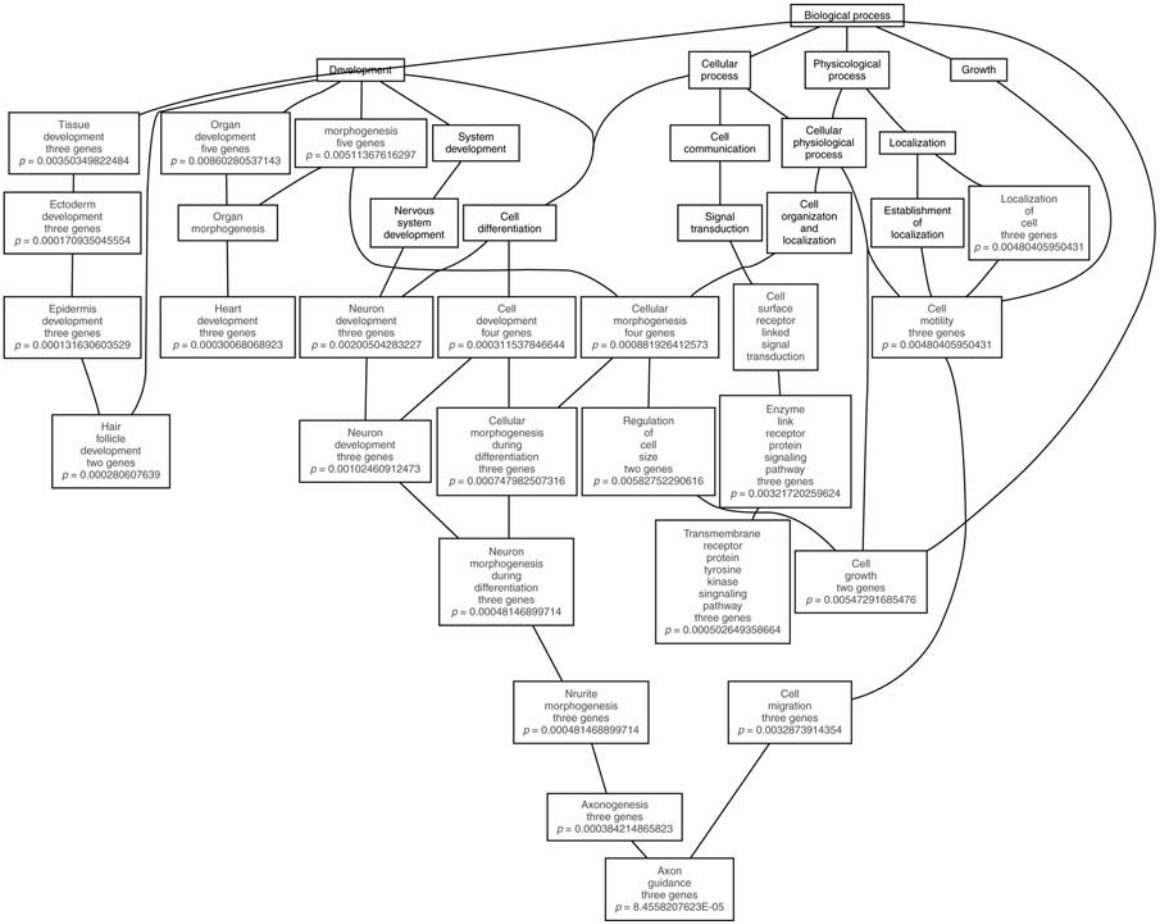


Fig. 1. Functional associations for gene set “brain embryo imprint.” Part of the DAG is shown owing to the size of the graph.

b. *From DAG:* press ENRICHED DAG (opens new window) and press one of the DAG boxes, containing the interesting GO category.
c. *By GO category keyword (might yield many or none gene sets):* type a key word (e.g., "neuron" or "development") in the box above KEY WORD SEARCH and press KEY WORD SEARCH button. This opens a new window with multiple categories, each showing the genes that were contained in the initial gene set (brain embryo imprint).
d. *From GO category identifier:* type the GO category name (e.g., "neuron differentiation") in the box above GO TERM SEARCH and press the button. Unlike the previous method (**step 7c**) this search will produce one or zero (if there is no intersection between the category and the initial gene set or if the category is not correctly identified) gene sets.

As shown in **Fig. 1**, there is strong association between the input gene set and several categories, most notably neuron development and nervous system development. Discovering all of these associations manually would have been extremely difficult.

3.4. Pathway Association Analysis

Some of the tools in **Table 1** can be used for finding associations among gene sets and known biological pathways. The primary sources of data for this analysis are KEGG and BioCarta. One exception is the PANTHER analysis tools, which are supported by their own database. The general considerations when choosing the appropriate tool are similar to GO analysis tools.

There are three steps: (1) uploading the gene set, (2) selecting (or uploading) a reference gene set, and (3) performing the analysis.

1. Retrieve the "lymph node" gene set as in **Subheading 3.1.2.2.**
2. Press the BioCarta or KEGG button.
3. Select a reference set by choosing "human" (all genes in the human genome).
4. At this point, a list of pathways is available. Pathways with a desired *p*-value (default value of 0.01) are highlighted in red. Clicking the name of the pathways will cause redirection to the source website (KEGG or BioCarta) and display the pathway maps. Genes from the original gene set are highlighted in the KEGG pathway map.
5. Export/Save the image (*see* **Fig. 2**).

3.5. Other Types of Associative Analysis

Other associations also could be of some value. Among these are protein domains, genome reference into function, and Pubmed. The association analysis for this type of data is performed in the same way as the pathway analysis in WebGestalt (*see* **Subheading 3.4.**), after pressing the corresponding button.

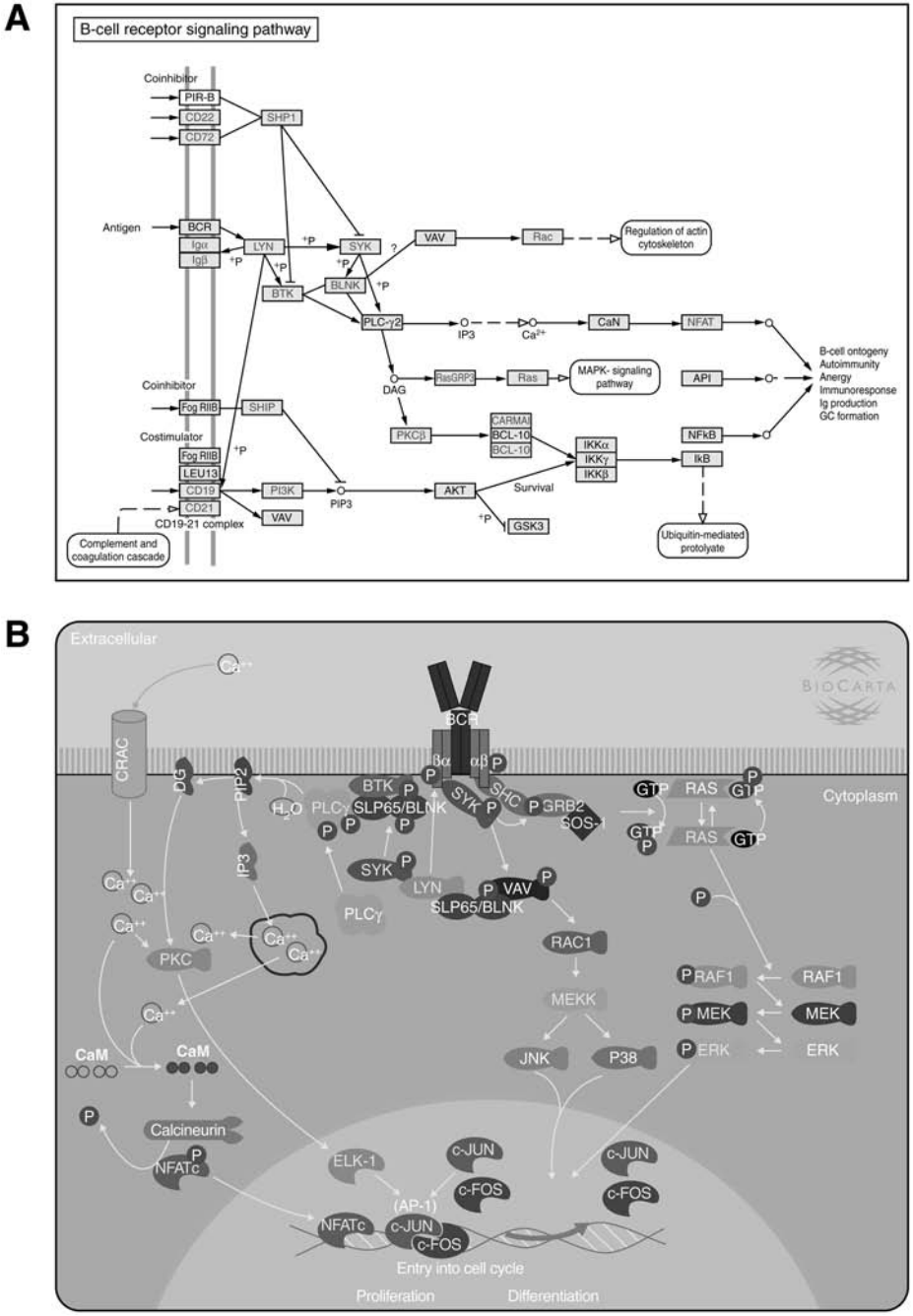


Fig. 2. Pathway analysis: **(A)** KEGG and **(B)** Biocarta.

3.6. Tissue Expression and Chromosomal Distribution of the Gene Set

Associations are not quantified in these two analyses. The tissue-expression pattern for the whole gene set is visualized based on the CGAP project publicly available data. The chromosomal distribution function is visualized based on the UCSC annotation.

1. Press the "Tissue Expression Bar Chart" or "Chromosomal distribution Chart" button (this will open a new window).
2. Click the tissue/chromosome bar representing the tissue/chromosome of interest to retrieve genes associated with the tissue/chromosome (this also opens a new window).
3. For the tissue expression analysis, the significance of enrichment for each gene in the gene list is provided (*see* **Note 9** for the evaluation of the significance of enrichment). One could also click on each member of the list to see its tissue distribution.
4. Click "SAVE" button to create a new gene set.

3.7. Advanced Batch Mode Data Mining With GeneKeyDB

Often it is important to find associations for which there is no tool available. In such case installing and querying one of the genome annotation databases is necessary. As a rule, most annotation databases are bulky. Two systems, GeneKeyDB and BioMART, provide a lighter solution to this problem. Designing custom-associative analysis based on either GeneKeyDB or BioMART requires higher level of computer proficiency and will not be discussed herein. A guide on installing GeneKeyDB locally is available at http://bioinfo.vanderbilt.edu/genekeydb/mirroring.

4. Notes

1. *Mailing lists and forums:* mailing lists and forums can be used to request help, new features, and to submit bugs. Forums and mailing lists can be accessed through sourceforge:
   Sourceforge projects:
   GenekeyDB: http://sourceforge.net/projects/genekeydb/.
   Mailing lists:
   GeneKeyDB: genekeydb-faqs@lists.sourceforge.net.
   WebGestalt: geneset-wg-faq@lists.sourceforge.net.
   GOTM: geneset-gotm-faq @lists.sourceforge.net.
2. *ID conversion:* the difficulties in ID conversion arise from the fact that the relationships among different IDs are not always one-to-one. For example, one unigene identifier can map to more than one Entrez gene ID. Therefore, mapping to Entrez gene ID through unigene might be unreliable and would produce unreliable final results unless the many-to-one relationships are taken into account. The same is true for other IDs as well, such as oligo microarray IDs.

3. *Gene symbols usage:* gene symbols are not reliable IDs in most cases. Often different research groups refer to the same gene with different gene symbols, and even worse to different genes with the same symbol. A search for gene symbol *TRAF2* matches two human genes—7186 and 10010 (EntrezGene IDs). In recent years different organizations have started the process of standardization of gene symbols (e.g., HUGO), but symbols are still unreliable and should be avoided whenever possible to prevent confusion and conflicts during identifier conversion.
4. *Database updates and gene set stability:* gene annotation changes as new information is incorporated in the publicly available databases. The process of compiling new knowledge and adding it to a database is known as "release" or "build." WebGestalt indirectly uses many such databases through GeneKeyDB. GeneKeyDB needs to be synchronized against each of its sources because older information in one of the source can create conflicts. After the update process the users' gene sets need to be reuploaded because the identifier mapping might have changed.
5. *GOTM vs WebGestalt GOTree module:* unlike GOTM web service, the WebGestalt GOTree module does not have its own gene set management and is dependent on the gene sets deposited to the user WebGestalt account. Therefore, any analysis will be lost when the analysis window is closed. The WebGestalt module is also restricted to include only the organisms, which are accessible through WebGestalt (currently human and mouse). Using the full GOTM service may be more appropriate in several cases.
6. *WebGestalt restrictions:* owing to the resource limitation, GOTM and the GOTree module of Webgestalt current restrictions are 500. It is planned to raise the limit in the near future (besides, this would be very useful as currently the author is constantly driven to other tools owing to this limit). Local installation of GeneKeyDB and GOTM can also provide the user with the ability to select his own limit, but describing the installation process for GOTM is out of the scope of this chapter.

   In the authors'experience GOTM/WebGestalt will work with Internet Explorer 4 or higher, Mozilla (Mozilla Foundation, Mountain View, CA), Safari (Apple, Inc., Cupertino, CA), and Konqueror (http://www.konquerer.org). If one experiences problems with other browser, one can switch to one of these aforementioned and alert the authors about the problem. Currently, there is no awareness of any bugs. The updating process might cause some unexpected behavior because of the unexpected change in the source files (e.g., Entrez Gene), if such circumstances occur, please contact the webmaster.

   Occasionally, the servers can experience difficulties with a high volume of requests, which will slow down the analysis considerably. Make sure the Internet connection is not too slow and try again later, and if one observes the same problems for a gene set of a reasonable size, one can alert the authors directly or through one of the mailing lists. To check the Internet connection: Ping www.google.com (type "ping www.google.com" in a console). Response time of less than 120 ms and no more than 5% lost packets is sufficiently good.
7. *Choosing a reference data set and statistical analysis—caveats:* care should be taken to choose an appropriate reference data set. For example, if the whole

genome is used as a reference list, but the data set is derived from a subset of the human genome, then the different ontological distribution will skew the statistical analysis. One safeguard is to always use the reference set for the microarray used to generate the expression data (if this is a gene expression study). If this chip is not available a request can be sent for it to be added, or upload it on own. It is useful to pick a convenient name, such as mychip_reference. Try not to include spaces and special symbols in the gene set name. Occasionally this will lead to a problem. Another option one can change is the statistical method used to analyze the data. Currently there are two options: Fisher and hypergeometric tests.

8. *GOTree module visualization options:* DAG and bar chart views are graphic interchange format (GIF) interactive pictures. Unlike DAG, the bar chart will work only at one of the GOtree structure levels in one of the three main branches. By default WebGestalt chooses level 4 in Biological processes.
9. *Tissue enrichment:* the gene expression profile is derived from CGAP publicly available data. A gene was considered for inclusion in the enriched set if it was overrepresented with $p < 0.01$ (with Bonferroni correction).

Acknowledgment

We would like to thank Suzanne Baktash for the technical help she provided in preparing this manuscript.

References

1. Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*. **270(5235),** 467–470.
2. Stoughton, R. B. (2005) Applications of DNA microarrays in biology. *Annu. Rev. Biochem.* **74,** 53–82.
3. Ashburner, M., Ball, C. A., Blake, J. A., et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet*. **25(1),** 25–29.
4. Bono, H., Nikaido, I., Kasukawa, T., Hayashizaki, Y., and Okazaki, Y. (2003) Comprehensive analysis of the mouse metabolome based on the transcriptome. *Genome Res*. **13(6B),** 1345–1349.
5. Kanehisa, M., Goto, S., Kawashima, S., Okunu, Y., and Hattori, M. (2004) The KEGG resource for deciphering the genome. *Nucleic Acids Res*. **32(Database issue),** D277–D280.
6. Lin, B., White, J. T., Lu, W., et al. (2005) Evidence for the presence of disease-perturbed networks in prostate cancer cells by genomic and proteomic analyses: a systems approach to disease. *Cancer Res*. **65(8),** 3081–3091.
7. Kluger, Y., Tuck, D. P., Chang, J. T., et al. (2004) Lineage specificity of gene expression patterns. *Proc. Natl. Acad. Sci. USA* **101(17),** 6508–6513.
8. Mi, H., Lazareva-Ulitsky, B., Loo, R., et al. (2005) The PANTHER database of protein families, subfamilies, functions and pathways. *Nucleic Acids Res*. **33(Database issue),** D284–D288.

9. OBO_Team, *Open Biomedical Ontologies Foundry*. (http://obofoundry.org/).
10. Ren, B., Robert, F., Wyrick, J. J., et al. (2000) Genome-wide location and function of DNA binding proteins. *Science* **290(5500),** 2306–2309.
11. Iyer, V. R., Horak, C. E., Scafe, C. S., Bostein, D., Synder, M., and Brown, P. O. (2001) Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature* **409(6819),** 533–538.
12. Dorschner, M. O., Hawrylycz, M., Humbert, R., et al. (2004) High-throughput localization of functional elements by quantitative chromatin profiling. *Nat. Methods*. **1(3),** 219–225.
13. Crawford, G. E., Holt, I. E., Whittle, J., et al. (2006) Genome-wide mapping of DNase hypersensitive sites using massively parallel signature sequencing (MPSS). *Genome Res*. **16(1),** 123–131.
14. Zhang, B., Kirov, S., and Snoddy, J. (2005) WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic Acids Res.* **33,** W741–W748.
15. Zhang, B., Kirov, S., and Snoddy, J. (2005) WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic Acids Res*. **33(Web Server issue),** W741–W748.
16. Zapala, M. A., Hovatta, I., Ellison, J. A., et al. (2005) Adult mouse brain gene expression patterns bear an embryologic imprint. *Proc. Natl. Acad. Sci. USA* **102(29),** 10,357–10,362.
17. Maglott, D., Ostell, J., Pruitt, K. D., and Tatusova, T. (2005) Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res*. **33(Database issue),** D54–D58.
18. Kasprzyk, A., Keefe, D., Smedley, D., et al. (2004) EnsMart: a generic system for fast and flexible access to biological data. *Genome Res.* **14(1),** 160–169.
19. Kirov, S. A., Peng, X., Baker, E., Schmoyer, D., Zhang, B., and Snoddy, J. (2005) GeneKeyDB: a lightweight, gene-centric, relational database to support data mining environments. *BMC Bioinformatics* **6(1),** 72.
20. Yusuf, D., Lim, J. S., and Wasserman, W. W. (2005) The Gene Set Builder: collation, curation, and distribution of sets of genes. *BMC Bioinformatics* **6,** 305.
21. Kirov, S. A., Peng, X., Baker, E., Schmoyer, D., Zhang, B., and Snoddy, J. (2005) GeneKeyDB: A lightweight, gene-centric, relational database to support data mining environments. *BMC Bioinformatics* **6**.
22. Beissbarth, T. and Speed, T. P. (2004) GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics* **20(9),** 1464–1465.
23. Shah, N. H. and Fedoroff, N. V. (2004) CLENCH: a program for calculating Cluster ENriCHment using the Gene Ontology. *Bioinformatics* **20(7),** 1196–1197.
24. Martin, D., Brun, C., Remy, E., Mouren, O., Thiaffry, D., and Jacq, B. (2004) GOToolBox: functional analysis of gene data sets based on Gene Ontology. *Genome Biol*. **5(12),** R101.
25. Dennis, G., Jr., Sherman, B. T., Hosack, D. A., et al. (2003) DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biol*. **4(5),** P3.
26. Zeeberg, B. R., Feng, W., Wang, G., et al. (2003) GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biol*. **4(4),** R28.

27. Al-Shahrour, F., Diaz-Uriarte, R., and Dopazo, J. (2004) FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics* **20(4),** 578–580.
28. Zhang, B., Schmoyer, D., Kirov, S., and Snoddy, J. (2004) GOTree Machine (GOTM): a web-based platform for interpreting sets of interesting genes using Gene Ontology hierarchies. *BMC Bioinformatics* **5,** 16.
29. Draghici, S., Kulaeva, O., Hoff, B., Petrov, A., Shams, S., and Tainsky, M. A. (2003) Onto-Tools, the toolkit of the modern biologist: Onto-Express, Onto-Compare, Onto-Design and Onto-Translate. *Nucleic Acids Res*. **31(13),** 3775–3781.
30. Castillo-Davis, C. I. and Hartl, D. L. (2003) GeneMerge—post-genomic analysis, data mining, and hypothesis testing. *Bioinformatics* **19(7),** 891–892.
31. Berriz, G. F., King, O. D., Bryant, B., Sander, C., and Roth, F. P. (2003) Characterizing gene sets with FuncAssociate. *Bioinformatics* **19(18),** 2502–2504.
32. Zhong, S., Storch, K. F., Lipan, O., et al. (2004) GoSurfer: a graphical interactive tool for comparative analysis of large gene sets in Gene Ontology space. *Appl. Bioinformatics* **3(4),** 261–264.
33. EGOn Beisvag, V., et al. (2006) Gene Tools—application for functional annotation and statistical hypothesis testing. *BMC Bioinformatics* **7,** p. 470.
34. Young, A., Whitehouse, N., Cho, J., and Shaw, C. (2005) OntologyTraverser: an R package for GO analysis. *Bioinformatics* **21(2),** 275–276.
35. Robinson, M. D., Grigull, J., Mohammad, N., and Hughes, T. R. (2002) FunSpec: a web-based cluster interpreter for yeast. *BMC Bioinformatics* **3,** 35.
36. Boyle, E. I., Weng, S., Gollub, J., et al. (2004) GO:TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics* **20(18),** 3710–3715.
37. Doniger, S. W., Salomonis, N., Dahlguist, K. D., Vranizan, K., Lawlor, S. C., and Conklin, B. R. (2003) MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biol*. **4(1),** R7.
38. Masseroli, M., Martucci, D., and Pinciroli, F. (2004) GFINDer: Genome Function INtegrated Discoverer through dynamic annotation, statistical analysis, and mining. *Nucleic Acids Res*. **32(Web Server issue),** W293–W300.

3

Estimating Gene Function With Least Squares Nonnegative Matrix Factorization

Guoli Wang and Michael F. Ochs

Summary

Nonnegative matrix factorization is a machine learning algorithm that has extracted information from data in a number of fields, including imaging and spectral analysis, text mining, and microarray data analysis. One limitation with the method for linking genes through microarray data in order to estimate gene function is the high variance observed in transcription levels between different genes. Least squares nonnegative matrix factorization uses estimates of the uncertainties on the mRNA levels for each gene in each condition, to guide the algorithm to a local minimum in normalized χ^2, rather than a Euclidean distance or divergence between the reconstructed data and the data itself. Herein, application of this method to microarray data is demonstrated in order to predict gene function.

Key Words: Clustering; least squares; microarray data analysis; nonnegative matrix factorization (NMF); pattern recognition; machine learning.

1. Introduction

Nonnegative matrix factorization (NMF) was introduced by Lee and Seung for image decomposition ***(1)***. Because of benefits in both interpretation and implementation, NMF was soon adopted in other research, including text mining ***(2)***, spectral decomposition ***(3)***, multiple sequence alignment ***(4)***, and neurophysiology ***(5)***. The application of NMF to microarray data analysis showed that it could be superior to clustering techniques for prediction of gene function ***(6,7)***. One issue that has limited application of NMF in many areas is that the patterns found within the data are diffuse, leading to attempts to limit the distributions through sparse matrix methods (e.g., *see* **ref.** ***8***). In addition, because measurements on mRNA levels of different genes show large differences in variance, a method that utilizes variance estimates was recently introduced to improve predictions of gene

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

function ***(9)***. Herein the authors present the methodology and demonstrate how to apply it to estimate gene function.

NMF aims to solve a problem in which a data matrix (**D**) can be decomposed by

$$\mathbf{D} = \mathbf{M} + \varepsilon = \mathbf{AP} + \varepsilon \tag{1}$$

and **M** represents a reconstruction of the data from two new matrices, **A** (amplitude) and **P** (pattern), and ε is the error on each element of **D**. For microarray data, the matrix **D** provides the estimates of mRNA levels for genes, such that each column corresponds to the estimate for a single condition, and each row represents levels for a single gene. A row of **D** corresponds to the processed intensity for a single gene across all conditions. **A** and **P** are the decomposed matrices, which define the assignment of genes to patterns (**A**) and the behavior of patterns across condition (**P**). Therefore, each row of matrix **P** can be viewed as representing an expression pattern, and each column of matrix **A** can be viewed as representing the amplitude distribution of each gene in the corresponding expression pattern. Therefore, genes linked within a column are linked to a behavior represented by the row **P**, and these genes can be expected to be linked to one or more biological behaviors. By comparing genes of unknown function in these groups to genes of known function, the function of the unknown genes can be predicted. Similarly, by noting the genes that have a behavior related to biological processes (such as genes expressed at a specific phase of the cell cycle), the biological role of these genes can be predicted.

The key issue to determine in applying an NMF approach to a problem is the cost function that will guide the analysis to the desired result. The cost function determines how the algorithm measures the difference between the data (**D**) and the estimation of the data, **M**. For instance, if two genes varied simultaneously but with different amplitudes, a Pearson correlation would be more useful than a measure that took into account differences in levels of expression, such as Euclidean distance. The change in cost function is the primary improvement in least squares nonnegative matrix factorization (LS-NMF) for microarray data, as LS-NMF minimizes

$$E_{\mathrm{s}} = \left\| \frac{\mathbf{D}-\mathbf{M}}{\sigma} \right\|^2 = \sum_{ij} \left(\frac{\mathbf{D}_{\mathrm{ij}} - \mathbf{M}_{\mathrm{ij}}}{\sigma_{\mathrm{ij}}} \right)^2 = \sum_{ij} \left(\frac{\mathbf{D}_{\mathrm{ij}} - \sum_k \mathbf{A}_{\mathrm{ik}} \mathbf{P}_{\mathrm{kj}}}{\sigma_{\mathrm{ij}}} \right)^2 \tag{2}$$

the normalized χ^2 measure, instead of

$$E_{\mathrm{e}} = \|\mathbf{D}-\mathbf{M}\|^2 = \sum_{ij} (\mathbf{D}_{\mathrm{ij}} - \mathbf{M}_{\mathrm{ij}})^2 = \sum_{ij} \left(\mathbf{D}_{\mathrm{ij}} - \sum_k \mathbf{A}_{\mathrm{ik}} \mathbf{P}_{\mathrm{kj}} \right)^2 \tag{3}$$

the Euclidean distance. The inclusion of the gene and array specific standard deviation (σ_{ij}) improves the recovery of functional information ***(9)***.

2. Materials

1. The source code for LS-NMF including a graphical user interface for loading input files and visualizing output files can be downloaded from the Fox Chase Cancer Center Bioinformatics Group at http://bioinformatics.fccc.edu/software/OpenSource/LS-NMF/java/LS-NMF.shtml (*see* **Note 1**).
2. The ClutrFree visualization and gene oncology analysis tool is available from http://bioinformatics.fccc.edu/software/OpenSource/ClutrFree/clutrFree.shtml.
3. The sample data set and associated gene ontology (GO) annotations can be downloaded from http://bioinformatics.fccc.edu/papers/methodsLS-NMF/data_GO.zip.
4. An updated version of the automated sequence annotation pipeline (ASAP II) is available at http://bioinformatics.fccc.edu/software/OpenSource/ASAP/ASAP.shtml; however, it does require considerable systems administration skills to implement. Users might instead gather GO and other annotations for ClutrFree using different systems, such as OntoExpress ***(10)***.
5. It is often useful to access the organism specific database for the particular data set. Here GeneDB and the *Schizosaccharomyces pombe* database is used, http://www.genedb.org/genedb/pombe/.

3. Methods

The procedure for LS-NMF simulation on microarray data sets involves three steps:

1. Preprocessing microarray data into proper format for LS-NMF analysis (*see* **Note 2**).
2. Setting parameters for LS-NMF simulation, and running the simulation with the set of the parameters.
3. Interpreting the simulation results.

In order to go through the whole procedure in detail, a sample microarray data set that is a reduced version of the *S. pombe* cell cycle experiment is provided ***(11)***. Every step in the implementation is applied specifically on this sample data set, so readers can follow the description below step-by-step. For different data, the steps are the same. It is recommended that, users new to bioinformatics tools apply the process first to the sample data set to learn the procedures.

3.1. Downloading Files and Preparing for Analysis

Each of the files noted in the **Subheading 2.** should be downloaded (with the exception of the ASAP system). This can be done with a typical web browser on any system. The files should be handled in the following manner.

1. Download the LS-NMFrun.zip file and place it in a directory (i.e. [Apple Inc., Cupertino, CA] folder) LS-NMF. Unzip this file (double-click on LS-NMFrun.zip on a Macintosh or Windows [Microsoft Inc., Redmond, WA] computer, give the command gunzip LS-NMFrun.zip on a Linux or Solaris [Sun Microsystem Inc., Santa Clara, CA] computer).
2. Download the LS-NMF_DATA.zip file and place it in a directory SampleData. Unzip this file as well. The reduced data is made up of ratio values for 169 genes across 20 time-points.
3. Download the ClutrFree tool. This can be placed in any directory, as it is an executable jar file.
4. If one wishes to setup the ASAP, download the system and follow the installation instructions.

3.2. Preprocessing the Data

The important advantage of the LS-NMF algorithm is its nonnegative constraints, which matches the biology of mRNA levels (no negative quantities), whereas reduces the mathematical space required to be searched to identify **A** and **P** matrices that can explain the observed data. As many researchers provide log-transformed data, it is necessary to transform such data into ratios. The transformation to use depends on the original log-transform, but most typically it is 2^{logratio} as the $\log_2$ ratios are most commonly used. Such transformations can be done using a spreadsheet program. For original data, it is merely necessary to generate ratios for two color arrays or use expression estimates from Affymetrix (Santa Clara, CA), such as provided by robust multichip analysis ***(12)***.

The input data format used by the downloaded LS-NMF package is the same as most commonly used microarray analysis tools, such as the Multiexperiment Viewer (MEV) tool ***(13)***, i.e., matrix **D** and σ are stored tab-delimited in files. Robust multichip analysis provides this format as a standard output. Matrix **D** should be stored in a file named FILENAME.txt, and matrix σ is stored in FILENAME.unc. The format has the first row in FILENAME.txt and FILENAME.unc as a header, which labels the conditions, one column for each condition. The first column of each file provides the gene ID (e.g., probeset ID, gene name, and so on) (*see* **Note 3**).

The sample data set is already in this format, with the upper left portion of the tab-delimited cdc25-sep1.txt file appearing as

	time0	time1	time2	time3	time4
SPAC222.09	1.1886973	1.4043094	1.2545819	0.9742178	0.8308763
SPAC977.10	1.4588974	1.4858006	1.6240937	1.5055444	1.7027408
SPAC821.06	1.0014285	1.0467643	1.0772699	1.0653352	1.1393371
SPAC821.09	1.3300012	1.6415249	2.7272928	1.8456596	1.8935258
SPAC821.11	1.0598825	0.9647703	0.93459004	0.9713597	1.0335108
SPAC23C4.13	1.0806911	1.2341665	1.4772284	1.5164454	1.3917305

with the first row giving the conditions (here, time-points) and the first column giving gene names. The other data points then provide measurements of gene expression for each gene in each condition.

3.3. Setting Parameters and Running the Simulation

LS-NMF requires four parameters for a simulation. The first parameter is the flag to enforce use of uncertainty estimates (*see* **Note 4**). The second parameter is the dimensionality of the factorization. As there is no available method yet to choose the dimensionality *a priori*, different values are usually tested in order to find an optimal dimensionality for analysis, so the dimensionality parameter is usually a range. The third parameter provides the number of repeated LS-NMF simulations for each specific dimensionality (*see* **Note 5**). The fourth parameter is the maximum number of update steps, which terminates the simulation after the specified number of steps whether or not the simulation has reached convergence.

These parameters are set using the interface of PattRun (*see* **Note 6**). At this point the PattRun algorithm should be started:

1. On a Macintosh or Windows computer double-click on the PattRun.jar icon. This will launch the interface. On a Linux or Solaris computer, use the command java–jar PattRun.jar. The user interface will appear.
2. Click on the Load button, and choose the file cdc25-sep1.txt using the file chooser.
3. The interface will now appear as in **Fig. 1**, with the left hand column showing the default parameters for the simulation.
4. Change the parameters for the LS-NMF simulation. Note the arrows in **Fig. 2**, and change the numbers by clicking on the values and typing in the new values. Use

uncertFlag	1
StartRank	6
EndRank	6
Nchains	20

   which will set LS-NMF to use the uncertainty values in cdc25-sep1.unc, to try only six dimensions (herein to speed the process the correct number is identified), and to do 20 simulations. In addition, set the value of Niter further up in the interface to 5000.
5. The simulation can now be run, which is done simply by pressing the Run button. The display will note that the run is beginning, and it will update the status after the completion of each simulation.

3.4. Interpreting the Results

How to interpret decomposed matrices from an NMF simulation depends on how the original data matrix **D** is organized, and LS-NMF should be interpreted in the same way. Usually, matrix **D** provides the estimates of transcriptional

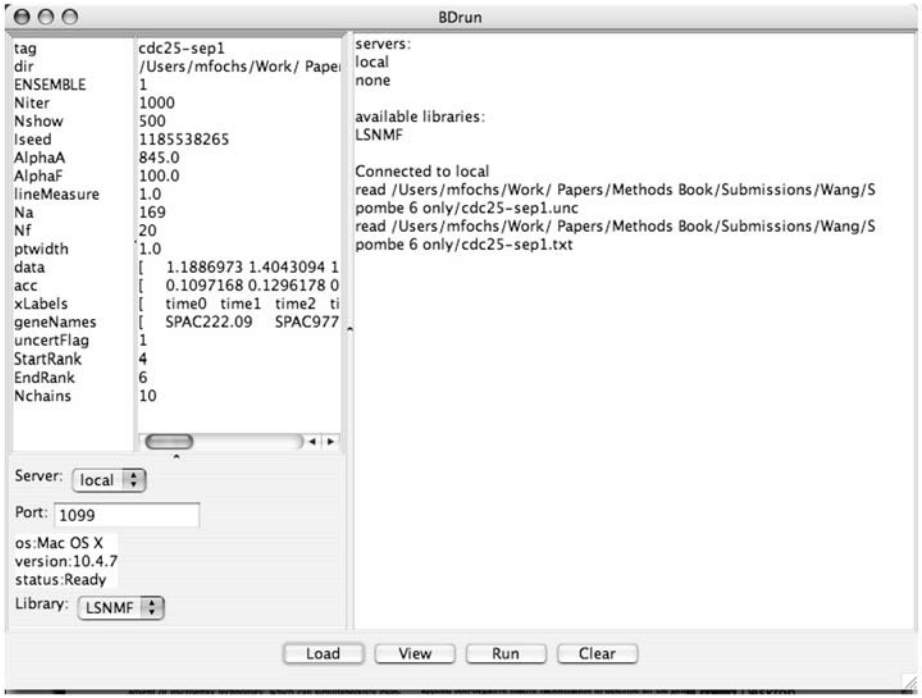


Fig. 1. The PattRun interface following loading of the data. If one has downloaded the version with BD software as well as LS-NMF software, one will need to choose LS-NMF from the pop-up menu in the bottom left.

levels of genes, such that each column corresponds to the estimate for a single condition, with each matrix element in a column corresponding to the processed intensity for a single gene (or probe) in that condition. A row of **D** corresponds to the processed intensity for a single gene across all conditions. If **D** has dimension of $I \times J$, **A** has dimensions of $I \times K$, and **P** has dimensions of $K \times J$, where K is the dimensionality. Given the factorization $\mathbf{D} \sim \mathbf{AP}$, matrix **P** can be used to group the J conditions into K patterns, with each condition being placed into at least one pattern corresponding to the most highly expressed metagenes in that condition. So condition j is placed in pattern i if the P_{ij} is among the largest entries in column j. There are many ways to define the largest entries; Z-score is used in the original LS-NMF work ***(9)***. On the other hand, the **A** matrix can also be used to group the I genes into K clusters, which means each gene is placed into at least one cluster corresponding to the most significant metaconditions (or metasamples) in that gene. There are always dual views about the decomposition ***(6)***, but the view of decomposition in this metagene view is most common.

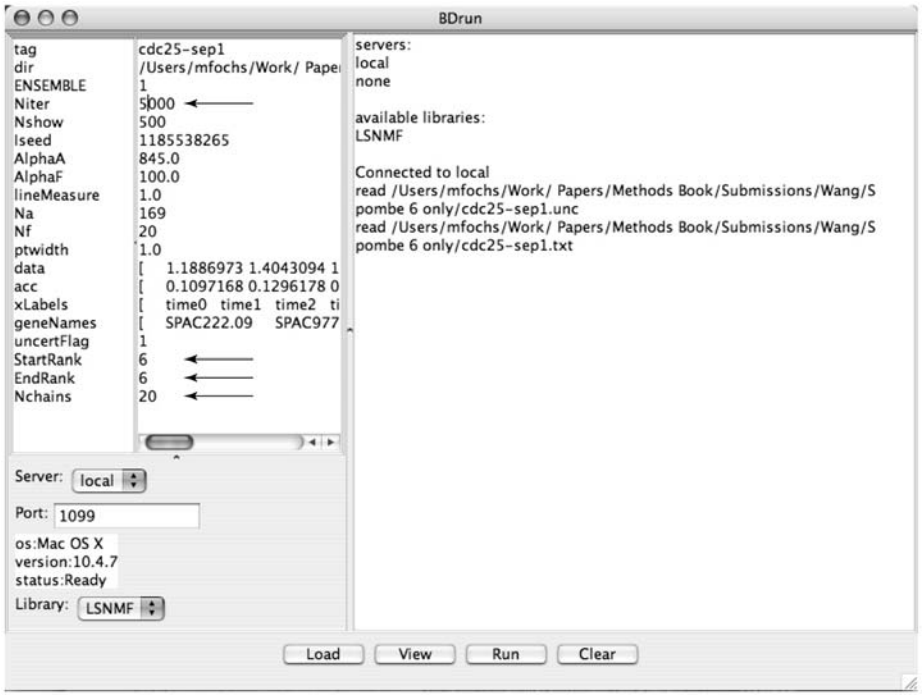


Fig. 2. The PattRun interface after setting of the parameters for running LS-NMF. The arrows show the location of parameters to set before running the simulations.

In order to get comparable decomposed matrices, amplitude (**A**) and pattern (**P**) matrices need to be normalized. The normalization chosen herein sets each row of **P** to unit amplitude (i.e., the sum of all elements of the row is 1). The columns of **A** matrix must then be scaled inversely to leave **M** unchanged (*see* **Eq. 1**). There are many freely available tools that can be used to interpret the results from LS-NMF simulation, herein ClutrFree is used ***(14)***, which is described in Chapter 1 in this work. For mouse and human data, WebGestalt ***(15)*** is a useful web-based system that is described in Chapter 2 in this work. The graphical version of LS-NMF generates output files appropriate to use with ClutrFree (*see* **Note 7**).

3.4.1. Choosing a Single Best Simulation for Analysis

PattRun with LS-NMF will generate a series of results stored in directories (folders) in the directory containing the original data. Each directory will contain one simulation result. The Viewer button on the PattRun window can be used to view these results one at a time. When it is pushed the first

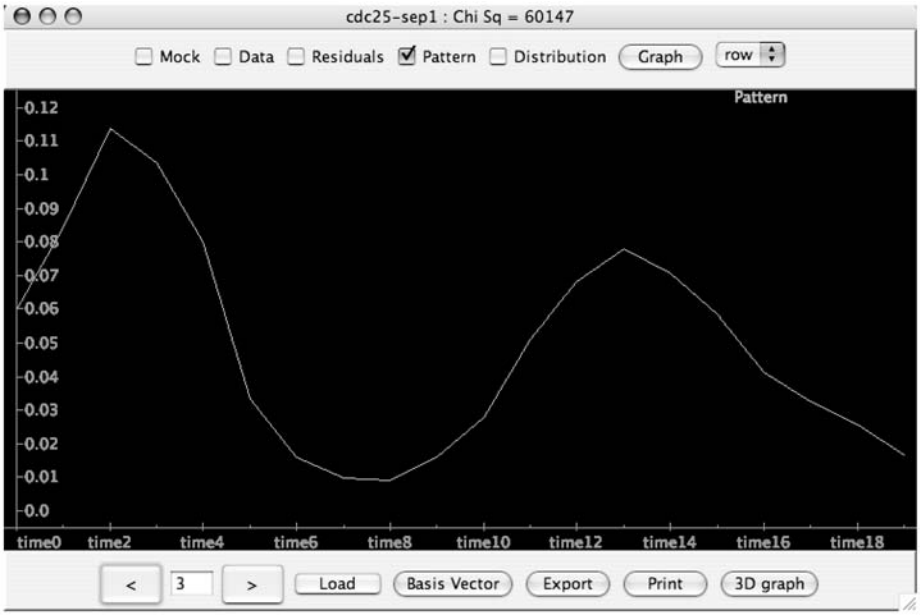


Fig. 3. The PattRun-viewer interface with the pattern shown. The viewer allows the user to look at the data, the reconstructed data (mock), the residuals, the patterns, and the distributions. In addition, using the basis vector button allows users to look for genes strongly tied to one or more patterns.

time, it will load the most recent simulation for visualization. Different files can be visualized by pressing the Load button.

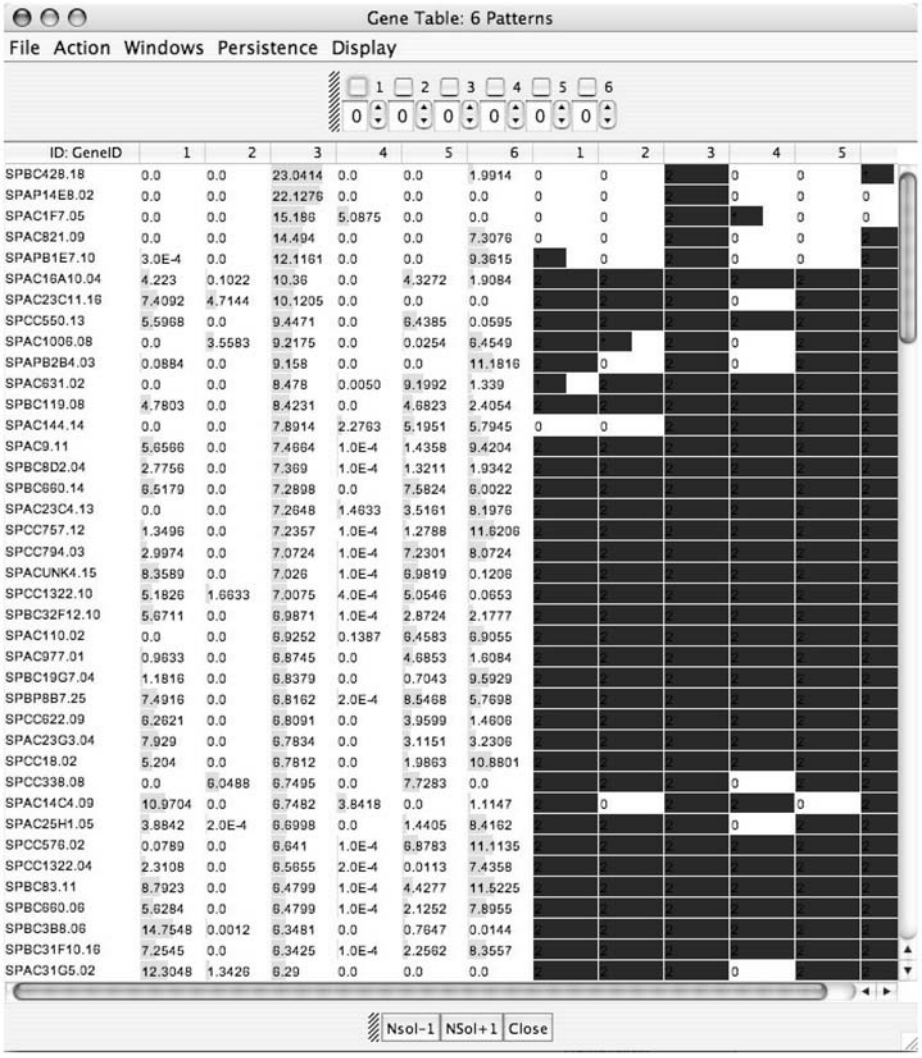
1. Load each simulation in turn, and record the χ^2 value at the top of viewer window. NMF algorithms will move to the local minimum in χ^2, so generally it will not find an optimal solution.
2. Reload the simulation with the best χ^2 value. This will be the simulation to use for further analysis. One can look at each pattern (behavior across condition, herein time-course) and decide if a pattern is of interest. To do this, mark the Pattern checkbox and press the Graph button. Use the **<** and **>** buttons to move between patterns. An example is shown in **Fig. 3**.
3. The best simulation can be output to a tab-delimited file by pressing the Export button. Provide a name for the output file in the Save dialog. Mark the Pattern and Distribution, check boxes, and then press the Save button. The output file can be edited with a spreadsheet program, the patterns can be plotted, and in **Subheading 3.4.3.** this information can be used for interpretation.

3.4.2. Interpreting the Simulation Results With ClutrFree

ClutrFree is a visualization tool for analyzing gene enrichment, GO annotation, and other aspects involved in gene expression and phylogenomic studies.

The annotation information ClutrFree needs can be prepared by using the ASAP system ***(16)*** or by generating tab-delimited GO files. For this example, sample GO files are provided in the download (cdc25-sep1_GO.txt).

1. Relocate to the directory (folder) wherein the cdc25-sep1.txt file is located (i.e., where LS-NMF was run). Review the recorded χ^2 values. Move all directories that are not from the lowest value to another directory outside the directory hierarchy. Alternatively, one might keep all directories that contain runs of similarly low χ^2 values, and ClutrFree will analyze all of these simultaneously (as described in Chapter 1). For this demonstration, there are two files.
2. Place the cdc25-sep1_GO.txt file in the directory and rename it annot.txt.
3. Start ClutrFree by double-clicking on the ClutrFree.jar icon on a Macintosh or Windows computer. On a Linux or Solaris computer, use the command java–jar ClutrFree.jar. The user interface will appear.
4. In ClutrFree, click on the File menu and choose Import Data. In the file chooser, move to the folder containing the annot.txt file, highlight that folder, and click on the Choose button. This will load the data and GO annotations. A new window for viewing the cluster shapes and a tree relating the clusters to each other for each analysis will appear. The >> button allows the users to view the individual cluster shapes (or pattern).
5. Next, press on the gene table button. A new window will open with the genes in the analysis listed together with their assignment to each pattern (yellow bars) and their persistence along the tree (blue bars).
6. For a pattern of interest (herein the third pattern is chosen, which in ones simulation is related to the G1 phase of the cell cycle), click on the number of the pattern above the yellow bars (*see* **Fig. 4**). This will reorder the genes by their strength within the pattern (*see* **Note 8**).
7. Use an appropriate website or annotation service to get specifics on each gene. For the *S. pombe* data, this can be done using GeneDB. For each gene that is highly tied to a pattern, one can retrieve details using GeneDB. Alternatively, one can use automated systems to do this.
8. For GeneDB, enter the gene ID in the search field, when the gene page appears one can add the gene to the basket. Do this for each gene that is strongly tied to the pattern. Unfortunately, this requires setting a cutoff and there is no reliable way to do this. In general, for this manual method, choosing the top 10 or 15 genes will typically give a list of genes with known and unknown functions.
9. Using the genes with known function, or the behavior of the pattern (herein a G1 linked cell cycle pattern), predict the gene function for unknown genes (*see* **Note 9**). This is then a prediction for the function of genes with unknown function. For this case, it is predicted that the gene SPAC1006.08 is involved in the G1 phase of the cell cycle, even though it is also involved in other patterns (2 and 6), which appear related to background processes (*see* **refs. *17*** and ***18*** for examples of analyses with such processes). In addition, one would predict that SPAP14E8.02, a predicted transcription factor, is uniquely involved in cell cycle, as its entire behavior is explained by pattern 3.



ID: GeneID	1	2	3	4	5	6	1	2	3	4	5	
SPBC428.18	0.0	0.0	23.0414	0.0	0.0	1.9914	0	0		0	0	
SPAP14E8.02	0.0	0.0	22.1276	0.0	0.0	0.0	0	0		0	0	0
SPAC1F7.05	0.0	0.0	15.186	5.0875	0.0	0.0	0	0			0	0
SPAC821.09	0.0	0.0	14.494	0.0	0.0	7.3076	0	0		0	0	
SPAPB1E7.10	3.0E-4	0.0	12.1161	0.0	0.0	9.3615		0		0	0	
SPAC16A10.04	4.223	0.1022	10.36	0.0	4.3272	1.9084						
SPAC23C11.16	7.4092	4.7144	10.1205	0.0	0.0	0.0				0		
SPCC550.13	5.5968	0.0	9.4471	0.0	6.4385	0.0595						
SPAC1006.08	0.0	3.5583	9.2175	0.0	0.0254	6.4549				0		
SPAPB2B4.03	0.0884	0.0	9.158	0.0	0.0	11.1816		0		0		
SPAC631.02	0.0	0.0	8.478	0.0050	9.1992	1.339						
SPBC119.08	4.7803	0.0	8.4231	0.0	4.6823	2.4054						
SPAC144.14	0.0	0.0	7.8914	2.2763	5.1951	5.7945	0	0				
SPAC9.11	5.6566	0.0	7.4664	1.0E-4	1.4358	9.4204						
SPBC8D2.04	2.7756	0.0	7.369	1.0E-4	1.3211	1.9342						
SPBC660.14	6.5179	0.0	7.2898	0.0	7.5824	6.0022						
SPAC23C4.13	0.0	0.0	7.2648	1.4633	3.5161	8.1976						
SPCC757.12	1.3496	0.0	7.2357	1.0E-4	1.2788	11.6206						
SPCC794.03	2.9974	0.0	7.0724	1.0E-4	7.2301	8.0724						
SPACUNK4.15	8.3589	0.0	7.026	1.0E-4	6.9819	0.1206						
SPCC1322.10	5.1826	1.6633	7.0075	4.0E-4	5.0546	0.0653						
SPBC32F12.10	5.6711	0.0	6.9871	1.0E-4	2.8724	2.1777						
SPAC110.02	0.0	0.0	6.9252	0.1387	6.4583	6.9055						
SPAC977.01	0.9633	0.0	6.8745	0.0	4.6853	1.6084						
SPBC19G7.04	1.1816	0.0	6.8379	0.0	0.7043	9.5929						
SPBP8B7.25	7.4916	0.0	6.8162	2.0E-4	8.5468	5.7698						
SPCC622.09	6.2621	0.0	6.8091	0.0	3.9599	1.4606						
SPAC23G3.04	7.929	0.0	6.7834	0.0	3.1151	3.2306						
SPCC18.02	5.204	0.0	6.7812	0.0	1.9863	10.8801						
SPCC338.08	0.0	6.0488	6.7495	0.0	7.7283	0.0				0		
SPAC14C4.09	10.9704	0.0	6.7482	3.8418	0.0	1.1147		0			0	
SPAC25H1.05	3.8842	2.0E-4	6.6998	0.0	1.4405	8.4162				0		
SPCC576.02	0.0789	0.0	6.641	1.0E-4	6.8783	11.1135						
SPCC1322.04	2.3108	0.0	6.5655	2.0E-4	0.0113	7.4358						
SPBC83.11	8.7923	0.0	6.4799	1.0E-4	4.4277	11.5225						
SPBC660.06	5.6284	0.0	6.4799	1.0E-4	2.1252	7.8955						
SPBC3B8.06	14.7548	0.0012	6.3481	0.0	0.7647	0.0144						
SPBC31F10.16	7.2545	0.0	6.3425	1.0E-4	2.2562	8.3557						
SPAC31G5.02	12.3048	1.3426	6.29	0.0	0.0	0.0				0		

Fig. 4. The gene table view in ClutrFree. The table has been sorted by Pattern 3, giving a list of genes that are tied to this pattern strongly. A more formal measure of the association can be calculated through the *Z*-score as described in the text.

3.4.3. Interpreting the Results Using Z-Scores

This approach will generate a list exactly matched to the ClutrFree view. The advantage of using *Z*-scores, which can be done with statistical software or with a spreadsheet, is that, one can compare the assignment of genes across patterns, as the *Z*-score will normalize the strength of assignment, based on the

variability within a pattern. Herein the process using a spreadsheet program is described, as these are widely available.

1. Output the **A** (distribution) matrix for the best simulation from the PattRun program by pressing the Export button once the results are viewed. Provide a name for the output file in the Save dialog. Check only the box for the Distribution and then press the Save button.
2. Open the output file with a spreadsheet program. The top lines of the file will look like

cdc25-sep1						
60147.00						
Wed	Sep	13	13:42:57	2006		
Distribution						
SPAC222.09	10.03	3.53	5.75	0	1.28	0
SPAC977.10	10.79	0	0	10.98	0	0.12
SPAC821.06	5.59	0.02	5.10	0.31	6.88	2.09
SPAC821.09	0	0	14.49	0	0	7.31

   with a header providing the name of the data set, the χ^2 value, and the date of the LS-NMF analysis. After the header, each row provides the gene name and the strength of assignment of that gene to each of the six patterns.
3. Choose the pattern of interest; herein again focus is on pattern 3 (column D in the spreadsheet). Calculate the mean and standard deviation of the column by replacing cell D3 with "=average(d6:dN)", where N is the last row with data and replacing cell D4 with "=stdev(d6:dN)." Using cut and paste these can be calculated for all patterns, if one wishes.
4. In an empty column, move to the sixth cell and enter "=(d6-d4)/d5" and return. Then copy this cell and fill down. These are the *Z*-scores for the genes.
5. Copy the first column (gene names) and the *Z*-scores, so they are side-by-side. If pasting into a new spreadsheet or page, choose to paste values. Sort the columns by the *Z*-score.
6. Again, one must choose a cutoff to produce a gene list; however in general, the larger the magnitude of the *Z*-score the more strongly a gene is associated with a pattern. This allows one to compare the strength of association of a gene across different patterns. Comparison with the list from ClutrFree will show the genes are in the same order, but the values have changed.

4. Notes

1. In addition, two command line C++ versions, one for single workstation (Desktop LS-NMF) and one for Beowulf cluster (LAM/MPI LS-NMF), are available for advanced users. Both versions are coded in C++, should be compiled using a standard C++ and mpiCC. Packages are downloadable in tar ball form, and a README file is included with all necessary steps for installation. Other than the LS-NMF code itself, two Perl scripts are included under API subdirectory for

results posttreatments, and a small example for microarray data set is also included in subdirectory of EXAMPLES.
2. The file formats and all further directions relate to the graphical version of LS-NMF. For the command line version, follow the directions included with these files.
3. The file format is slightly less flexible than that used by the MEV tool, as only a single column for IDs is allowed.
4. The same application allows the user to run NMF as well as LS-NMF. Herein focus is only on LS-NMF.
5. By design, NMF algorithms find a local minimum in the misfit between the data **D** and the reconstructed data **M**. This makes the algorithms prone to false minima, and the general approach is to try many simulations and use the one with the best fit to the data for further analysis.
6. After signing a material transfer agreement, advanced academic users might instead download the PattRun software, which includes three versions of the Bayesian decomposition algorithm as well as LS-NMF. For those users, the same directions apply but LS-NMF must be chosen from the drop down list of algorithms.
7. For the command line versions, files must be converted for use with ClutrFree. Two Perl scripts are provided within the LS-NMF package. One script named ForClutrFree.pl is used to prepare the simulation results for ClutrFree, and the other one, ForWebGestalt.pl, is for WebGestalt.
8. The power of both LS-NMF and Bayesian decomposition is that they can assign genes to multiple patterns, which matches biological reality as genes are usually multiregulated. In **Fig. 4**, this can be seen in the multiple assignment of genes such as SPAC821.09 and SPAPB1E7.10 among many others.
9. Analysis with pattern recognition or clustering methods across conditions will tend to link genes that are related in biological processes. This makes it a good complement to sequence-based analysis that links genes with similar molecular function, owing to sequence conservation of protein motifs.

References

1. Lee, D. D. and Seung, H. S. (1999) *Nature* **401,** 788–791.
2. Chagoyen, M., Carmona-Saez, P., Shatkay, H., Carazo, J. M., and Pascual-Montano, A. (2006) *BMC Bioinformatics* **7,** 41.
3. Sajda, P., Du, S., Brown, T. R., et al. (2004) *IEEE Trans Med. Imaging* **23,** 1453–1465.
4. Heger, A. and Holm, L. (2003) *Bioinformatics* **19(Suppl 1),** I130–I137.
5. Tresch, M. C., Cheung, V. C., and d'Avella, A. (2006) *J. Neurophysiol* **95,** 2199–2212.
6. Brunet, J. P., Tamayo, P., Golub, T. R., and Mesirov, J. P. (2004) *Proc. Natl. Acad. Sci. USA* **101,** 4164–4169.
7. Kim, P. M. and Tidor, B. (2003) *Genome Res.* **13,** 1706–1718.
8. Gao, Y. and Church, G. (2005) *Bioinformatics* **21,** 3970–3975.
9. Wang, G., Kossenkov, A. V., and Ochs, M. F. (2006) *BMC Bioinformatics* **7,** 175.

10. Draghici, S., Khatri, P., Bhavsar, P., Shah, A., Krawetz, S. A., and Tainsky, M. A. (2003) *Nucleic Acids Res.* **31,** 3775–3781.
11. Rustici, G., Mata, J., Kivinen, K., et al. (2004) *Nature Genetics* **36,** 809–817.
12. Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003) *Nucleic Acids Res.* **31,** E15.
13. Saeed, A. I., Sharov, V., White, J., et al. (2003) *Biotechniques* **34,** 374–378.
14. Bidaut, G. and Ochs, M. F. (2004) *Bioinformatics* **20,** 2869–2871.
15. Zhang, B., Kirov, S., and Snoddy, J. (2005) *Nucleic Acids Res.* **33,** W741–W748.
16. Kossenkov, A., Manion, F. J., Korotkov, E., Moloshok, T. D., and Ochs, M. F. (2003) *Bioinformatics* **19,** 675–676.
17. Moloshok, T. D., Datta, D., Kossenkov, A. V., and Ochs, M. F. (2003) "Bayesian decomposition classification of the project normal data set" in *Methods of Microarray Data Analysis III*, (Johnson, K. F. and LIn, S. M., eds.), Kluwer Academic, Boston, pp. 211–232.
18. Moloshok, T. D., Klevecz, R. R., Grant, J. D., Manion, F. J., Speier, Jr. W. F., and Ochs, M. F. (2002) *Bioinformatics* **18,** 566–575.

4

From Promoter Analysis to Transcriptional Regulatory Network Prediction Using PAINT

Gregory E. Gonye, Praveen Chakravarthula, James S. Schwaber, and Rajanikanth Vadigepalli

Summary

Highly parallel gene-expression analysis has led to analysis of gene regulation, in particular coregulation, at a system level. Promoter analysis and interaction network toolset (PAINT) was developed to provide the biologist a computational tool to integrate functional genomics data, for example, from microarray-based gene-expression analysis with genomic sequence data to carry out transcriptional regulatory network analysis (TRNA). TRNA combines bioinformatics, used to identify and analyze gene-regulatory regions, and statistical significance testing, used to rank the likelihood of the involvement of individual transcription factors (TF), with visualization tools to identify TF likely to play a role in the cellular process under investigation. In summary, given a list of gene identifiers PAINT can: (1) fetch potential promoter sequences for the genes in the list, (2) find TF-binding sites on the sequences, (3) analyze the TF-binding site occurrences for over/under-representation compared with a reference, with or without coexpression clustering information, and (4) generate multiple visualizations for these analyses. At present, PAINT supports TRNA of the human, mouse, and rat genomes. PAINT is currently available as an online, web-based service located at: http://www.dbi.tju.edu/dbi/tools/paint.

Key Words: Clustering; gene expression; gene regulation; network analysis; pattern recognition; transcription factors.

1. Introduction

Biomedical scientists have a long standing interest in acquiring gene lists because the character of differentiated cellular function and disease is often well described in this fashion. Organ system structure and function are the product of variations in differentiated gene expression (and gene-expression products) in interaction with the environment. Disruption of these distinct patterns of active genes can lead to organ disease and changes in behavior. Thus, associating patterns of gene activity (i.e., gene lists) with structure and function is a key ongoing

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

activity and problem within the biomedical research community. Gene lists can be and have been acquired in any number of ways (e.g., differential display ***[1]***, serial analysis of gene expression ***[2]***, and large-scale cDNA sequencing ***[3]***, or derived from the literature or by pathway analysis), and recently this has been greatly accelerated, as reagents have become available from the genome projects enabling "functional genomic" techniques. These "-omics" techniques continue to rapidly proliferate, develop, and improve, and it is clearly by no means at the end of this technology revolution. At present, for example, DNA microarray methods have evolved to a point wherein gene expression can be simultaneously measured for tens of thousands of genes under multiple conditions ***(4–6)***.

However, bridging the gap from raw gene-expression data to interpretation or understanding of its relevance to functional processes remains a large unmet need. For example, it is now understood that in microarray studies the wet-lab data acquisition phase of the study is minor in comparison with the analysis work that follows. Thus, whatever the past, present, or future source of gene lists there is a tremendous unmet need in the domain of their analysis within the research community and opportunity for informatics developments to meet the need.

The objective of the cluster of ongoing interactive developments under the umbrella of a transcriptional regulatory network analysis (TRNA) framework is to work in the area of this unmet need in at least two ways.

1. The tools and analysis approaches are useful to the biologist who wants to analyze the biological context of a gene list in order to more effectively identify transcription factors (TF) and associated genes for more detailed study.
2. The informatics will strengthen development of hypotheses and predictions of the functional regulation of systems of genes, and thereby greatly facilitate development of model structures, as an approach to systems biological problems.

1.1. Rationale for a Systems-Level Approach

The two points at the end of the previous paragraph highlight the intent that TRNA will be a continuously evolving analysis approach that will expand in quality and application over time. The ability to develop useful informatics in this area depends not just on the developments but also on the continued rapid expansion and improvement of the web-accessible public and private resources on which the approach rests. There is every reason to believe this will not only continue but greatly accelerate. Recently, there have been elegant demonstrations in simple model systems of how these kinds of data can be combined into models of system function (e.g., **refs.** ***7*** and ***8***). Thus, the potential for synthesizing these kinds of data toward functional understanding is an exciting and realistic prospect, for example, of predictions of network models of gene regulation, gene output phenotype, and of biochemical pathways/networks. However,

even in simple systems this is a significant challenge, and one that is so far unaddressed for the particular needs of biomedically relevant mammalian cell systems.

Zak et al. ***(9,10)*** performed *in silico* analyses of the potential for use of gene expression results in estimation of functions such as gene-regulatory networks. From these analyses it is clear that, whereas gene-expression data can greatly reduce the uncertainty in model estimation, meaningful predictions of a particular system of gene regulation (i.e., one that would be worth experimental test) cannot be reached using realistically obtainable gene-expression data alone. However, if gene-expression data can be combined with other data and/or knowledge, meaningful model predictions can be reliably achieved. For example, combination of gene-expression data with information on TF activity and localization can reliably predict gene-expression networks ***(10)***. These results support the hypothesis that the nascent large-scale data-acquisition methods in the present postgenomic period will be useful for a so-called systems biology approach, depending on development of appropriate informatics tools and analysis approaches motivating "promoter analysis and interaction network toolset" (PAINT) development for scalable TRNA ***(11)***.

1.2. TRNA Approach

As a starting point the TRNA approach resembles what an investigator would do when analyzing the regulation or expression of one or two genes by hand, finding what promoter sites are associated with the gene(s) and developing contextual information from the literature. However, in the case of TRNA this work is being done simultaneously for an indefinitely long gene list ***(11)***. The experimental and computational methods presented herein identify a set of genes and TF that are significant in understanding the function of the gene-regulatory network in question. The primary purpose of PAINT is to provide a scalable and extensible platform to automate the process of mining the existing databases for known regulatory information for a large number of genes of interest in a particular biological experiment or analysis. The analysis rests on use of databases of relevant information, includes evaluation of the results (e.g., of the probability of significance of a result), and automatically provides pattern data on gene groupings and relationships by various standards. Present technological developments have resulted in rapidly growing public resources containing systematic data sets of various types: gene expression changes from microarrays; protein–DNA interaction and TF-activity data from protein-binding assays, chromatin immunoprecipitation experiments, and DNA footprinting; protein–protein interactions from two-hybrid experiments and coimmunoprecipitation; and genomic sequence and ontology information in public databases.

Currently, PAINT can process a list of gene identifiers (GenBank accession numbers, Clone IDs, Ensembl Gene IDs, and Entrez Gene IDs) to retrieve corresponding promoter sequences and analyze the same for presence of TF-binding sites. The tool then uses a statistical analysis, with or without gene expression clustering results, to generate a set of candidate regulatory interactions and TF that are likely to play a key role in the mechanisms underlying the cellular response. PAINT has been used in studying co-ordinated gene regulation in a wide range of systems including neuronal differentiation, neuronal adaptation, blood cell development, retinal injury, brain stroke, and bladder inflammation ***(11–19)***.

2. Materials

1. *Gene level identifier resources:* the CloneUpdater tool for annotation updating and gene identifier conversion across different databases can be found at http://www.dbi.tju.edu/cloneupdater/html/template.php. The SOURCE tool for conversion between various gene identifiers can be found at http://source.stanford.edu. Ensembl gene identifiers can be obtained using the BioMart function at http://www.ensembl.org/Multi/martview.
2. *Gene list input data file:* a user-provided single column list of gene identifiers, one identifier per line, in a plain text file. The data set and associated identifier list files used in this article are a subset of the data described in **ref. *11*** and are available at http://www.dbi.tju.edu/dbi/publications/MiMBchapter/.
3. *Cluster membership data file:* a user-provided tab-delimited plain text file with two columns. The first column must contain one gene identifier per row and the second column must contain a corresponding single word alphanumeric cluster label. An example file is available in the online Supplemental Information at http://www.dbi.tju.edu/dbi/publications/MiMBchapter/.
4. *TF-binding site data:* TRNA requires definitions of TF-binding sites call positional weight matrices (PWM). PWMs for use with PAINT are provided in two forms from Biobase International Wolfenbüttel, Germany. A publicly available database of PWMs is accessed through http://www.gene-regulation.com. A professional and licensed version is available from http://www.biobase-international.com/. PAINT requires users to obtain an account with either of these resources if PAINT is to be used for binding-site analysis. The professional version of TRANSFAC™ Biobase International, Wolfenbüttel, Germany contains substantially higher number of TREs and TF than the public version, and hence, the use of former significantly improves the TRNA.
5. *PAINT:* the latest version of the PAINT is available at http://www.dbi.tju.edu/dbi/tools/paint/. The original version is described in **ref. *11***.

2.1. PAINT Architecture

The modular architecture of PAINT, depicted in **Fig. 1**, is not organism specific. The key requirements are the availability of annotated genome sequence and information on TF-binding site motifs. PAINT 3.5 can conduct analysis specific to the human, mouse, and rat genomes. The tool contains five components:

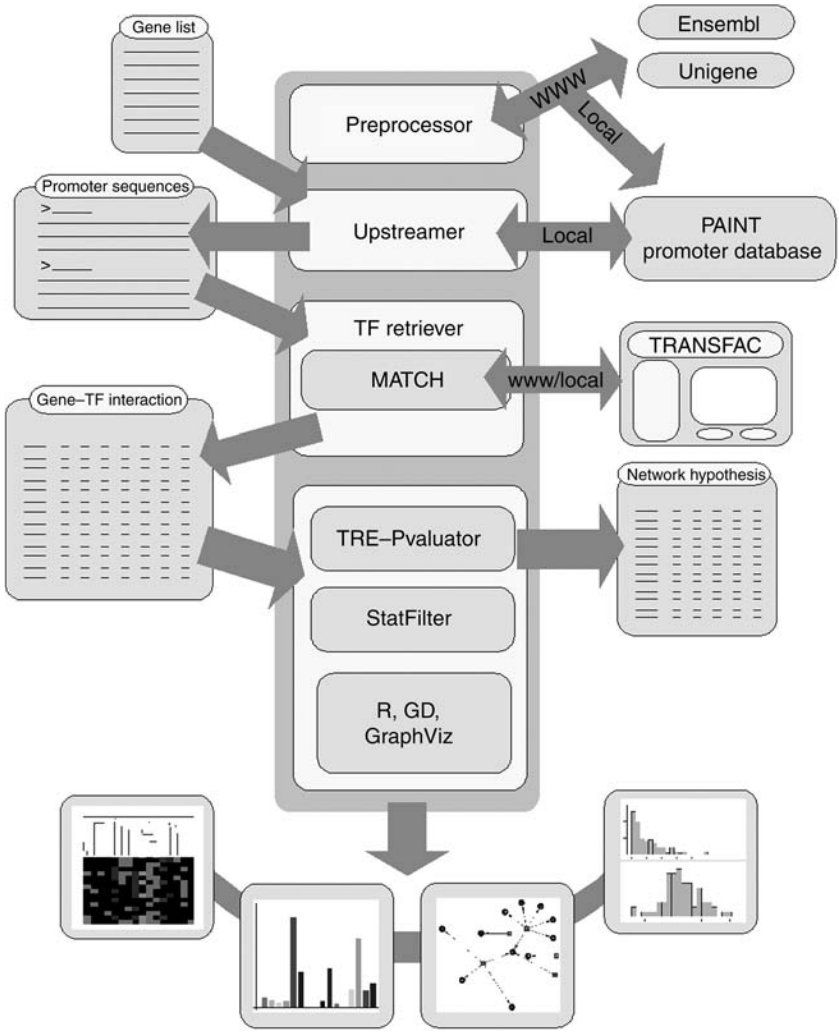


Fig. 1. A schematic of the PAINT architecture. The input and output data, the modules and their interactions, and different visualizations of the results are detailed in the text.

1. *PAINT promoter database:* a MySQL database containing a predicted promoter sequence for each gene that can be queried using the Ensembl GeneID, Entrez Gene ID, Clone ID, or GenBank accession number.
2. *Upstreamer:* a Perl module that provides the functionality of sequence retrieval from the PAINT promoter database given a list of unique identifiers for the genes of interest.
3. *TFRetriever:* a Perl module that processes the retrieved sequences through the TF-binding site inspection/discovery programs to identify potential TREs. The

dynamic nature of the databases containing TF information and user-specified parameter options require online retrieval rather than an offline processing for all the promoters in the PAINT promoter database.

4. *FeasnetBuilder:* a Perl module that processes the output of the TF inspection/discovery programs and produces a candidate interaction matrix, termed Feasnet, for the genes of interest.
5. *FeasnetAnalyzer and FeasnetViewer:* a Perl and R module that contains functions for analysis and visualization of PAINT results (TRE-Pvaluator, StatFilter, R, GD graphics library, Graphviz available at http://www.graphviz.org). A matrix image with optional clustering of data and a network layout diagram are available.

A detailed description of each of the modules and the input–output relationships is presented next.

2.2. PAINT Modules

2.2.1. PAINT Promoter Database and Preprocessor Module

For an organism of interest, the principal requirement for constructing the promoter database is annotated genome sequence assembly. Several genome assemblies are available for mammalian systems, for example, Ensembl ***(20)*** and Santa Cruz (http://genome.ucsc.edu), Celera (http://www.celera.com). For each of the human, mouse, and rat genomes, an UpstreamDB database was constructed for all the annotated genes (known and putative) in the corresponding Ensembl genome database. For each gene, 5000 bp upstream (5′ to the gene) were retrieved from the Ensembl database. The retrieved sequence was placed in the database only if at least 300-bp sequence immediately 5′ to the gene was available. The genome database contains sequences in 5′ to 3′ orientation on a single strand (conventionally denoted as +1) of DNA. For the genes that are located on the strand –1, the sequence from the genome database was reversed and complementary base pairs were computed to produce the upstream sequences.

One key aspect of any promoter analysis is using the correct sequence to represent the *cis*-regulatory control regions. Note that this requires information about the 5′-untranslated region of each gene in order to correctly identify the transcription start site, and hence, the corresponding adjacent *cis*-regulatory control region for each gene. In order to overcome the limitations of the incomplete annotation in Ensembl database, early versions of PAINT utilized 5′-untranslated region from RIKEN clone sequences to estimate the transcription start site in mouse genome ***(11,21)***. Subsequent versions of Ensembl annotation incorporated the experimentally determined 5′-untranslated sequence to the extent available, thus improving the Transcription start site (TSS) estimate significantly. Hence, starting from version 3.0, the preprocessor module in PAINT considers for each gene, the starting position of the

first exon in the Ensembl database to be the transcription start site. This approach was determined to be acceptable for *in silico* genome-wide location analysis ***(22)***.

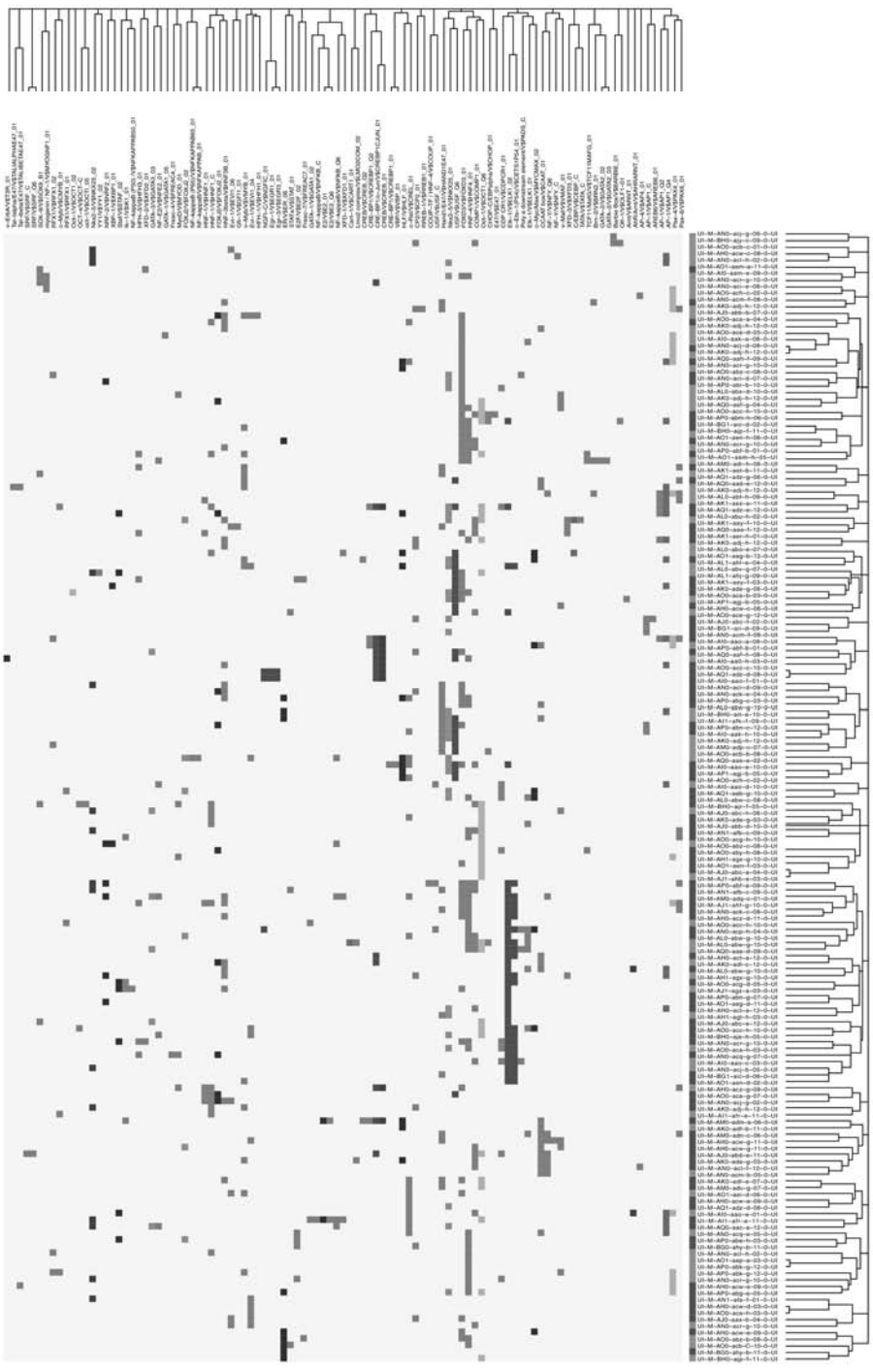
In addition to the promoter sequence for each gene, PAINT promoter database also contains the cross reference tables that enable retrieval of promoters using Entrez Gene, the cDNA clone ID, and Genbank accession number. This cross reference was constructed using information from the Unigene database. This allows for convenient retrieval of the promoter sequences directly from a list of genes marked as significantly varying in expression by any microarray analysis software or other gene-expression analysis methods. The PAINT promoter database is periodically updated when a new version of an annotated genome database is released.

2.3. The Upstreamer Module

The input from the user is a list of identifiers for the genes of interest and the number of base pairs of the upstream sequence needed for analysis. The length count is from the start of the gene toward the upstream (5′) end. The identifier list and parameters are used to query UpstreamDB. The output of the module is the actual genomic upstream sequences of specified length, for the genes that are on the user's list and referenced in the UpstreamDB database. The output is in FAST-ALL (source: http://www.ebi.ac.uk/fasta/) (FASTA) format for further processing by transcription binding motif inspection/discovery software in the TFRetriever module.

The TFRetriever module is envisaged to contain several submodules that can communicate with various local and web-based motif inspection and discovery software such as MATCH (TRANSFAC Public) ***(23)***, MatInspector ***(24)***, and MEME ***(25)***, and so on. A motif is a characteristic sequence of a binding site and functionally similar motifs are grouped together into families. PAINT 3.5 currently contains only the submodule for interacting with MATCH software. The set of vertebrate TF families is utilized for promoter inspection. The output of the TFRetriever module is the output from the motif discovery program for each input sequence list. TFRetriever runs MATCH with settings to minimize false-positives or to minimize the sum of false-positives and -negatives. However, users can filter the results further by specifying a threshold on the "core similarity" and choosing whether or not the Transcriptional Regulatory Element (TRE) occurrences on complementary sequence are to be considered in further analysis.

The FeasnetBuilder module processes and filters the output from MATCH to construct an interaction matrix (hereinafter termed "Feasnet") representing a candidate set of connections in the regulatory network based on the promoter sequence and TF/TRE information. The columns of the interaction matrix correspond to the TREs and each row corresponds to a gene from the input list. If the parameter for binary counting is set in PAINT, the regulation of a gene is



represented by one if the corresponding TRE is present on the promoter for that gene and by a zero otherwise. This matrix represents the constraints to a network identification scheme. The interaction parameters corresponding to zeros in the candidate matrix need not be computed, substantially reducing the dimensionality of the identification problem (**Figs. 2** and **3**).

The FeasnetAnalyzer module contains a submodule named StatFilter that calculates the significance of "enrichment" for each TRE resulting from comparing the selected genes submitted to PAINT with a random selection of genes. StatFilter computes *p*-values for the overrepresentation of each TRE in the set of promoters considered with respect to a background set of promoters. Specifically, the *p*-values give the probability that the observed counts for the TREs in the set of promoters could be explained by random occurrence in the background set of promoters. The *p*-values are calculated using the hypergeometric distribution ***(11,26–28)***. These raw *p*-values are adjusted for multiple testing using a false discovery rate (FDR) estimate ***(29)***. Typically, for a microarray experiment, the reference set is that of the genes on the microarray utilized in the experiments. For each TRE V$X, given (1) a reference Feasnet of n promoters of which n_1 promoters contain V$X, and (2) a Feasnet of interest with m promoters of which h contain V$X, the associated *p*-value for overrepresentation is given as in **Eq. 1**.

$$p = \sum_{i=h}^{\min(n_1,m)} \frac{\binom{n_1}{i}\binom{n-n_1}{m-i}}{\binom{n}{m}} \tag{1}$$

The *p*-value for underrepresentation of a TRE in the observed Feasnet is calculated similarly with the summation in the aforementioned equation going from 1 to h. These estimates of significance can be utilized in filtering for those TREs that meet a threshold (say, $p \leq 0.05$, or FDR-adjusted $p \leq 0.3$) to identify most likely regulators of the genes considered in the experimental context of interest. Given no information about the source of the genes from which the input list to PAINT is generated, PAINT can optionally utilize the Feasnet corresponding to all the genes in the PAINT promoter database as a reference Feasnet in the earlier enrichment analysis (also termed interchangeably as overrepresentation analysis).

Fig. 2. *(Opposite page)* A visualization of a Feasnet. The elements are color-coded to indicate the over- and underrepresentation of the transcriptional regulatory elements. Each row in the vertical color bar next to the gene identifiers indicates the cluster membership of the corresponding gene. The dendrograms are based on hierarchical clustering using average-linkage method and the binary distance as the dissimilarity metric. A high-resolution color version of the gray-scale image presented herein is available online at http://www.dbi.tju.edu/dbi/publications/MiMBchapter/.

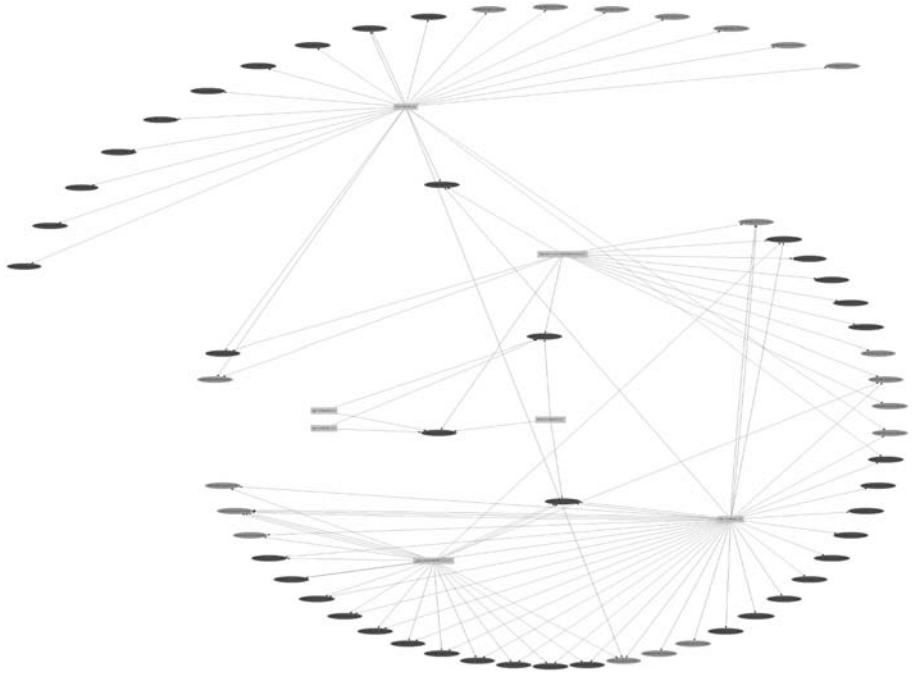


Fig. 3. A network visualization of a Feasnet that is filtered based on overrepresentation of transcriptional regulatory elements. Rectangular boxes represent the TREs and the elliptical boxes represent the promoters (colored based on the Gene Cluster Membership Data). A high-resolution color version of the gray-scale image presented herein is available online at http://www.dbi.tju.edu/dbi/publications/MiMBchapter/.

The FeasnetViewer module contains various functions for the visualization and analysis of a Feasnet. An image of the interaction matrix is produced in which the individual elements of the matrix are represented by a color based on the significance values for that particular TRE (*p*-values for overrepresentation in the observed Feasnet). This module also contains functionality for hierarchical clustering using "R" software for statistical analysis (http://www.r-project.org). For clustering, the pair-wise distance that is most appropriate for the Feasnet data is the binary distance. The binary distance between two genes (or TF) can be computed, as the ratio of number of elements for which the two rows (or columns) are dissimilar to the total number of elements for which either of the rows contains a one. For the genes, binary distance is the "dissimilarity" between the regulatory pattern of two genes as related to the total number of distinct binding sites present on either of them. For the TF, binary distance is the "dissimilarity" between the regulatory patterns of two TF as related to the total number of genes regulated by either of the TF.

In PAINT, the clustered data can be visualized as a matrix layout with the hierarchical tree structure aligned to the rows and the columns of the Feasnet. The zeros in the matrix are shown in black and the nonzero entries in the Feasnet are color based on the *p*-value of the corresponding TRE. The brightest shade of red represents low *p*-value (most significantly overrepresented in the Feasnet). Conversely, the brightest shades of cyan represent smaller *p*-values for underrepresentation in the observed Feasnet indicating more significantly underrepresented TREs. This image can optionally represent the cluster index of each gene, wherein such cluster indices are generated from other sources such as expression or annotation-based clustering. With such visualization, it is straightforward to explore the relationship between expression/annotation-based clusters and those based on *cis*-regulatory pattern (i.e., Feasnet). The FeasNetViewer module can also generate a network layout diagram using the GraphViz libraries (available at http://www.research.att.com/sw/tools/graphviz/). In the web-based PAINT, previous analyses can be retrieved and/or continued using a job key provided for each analysis. The PAINT results are presented in a hyperlinked report and can also be downloaded as a single compressed file for offline perusal.

Nomenclature for this article includes ***bold italic*** for onscreen text, **bold** for buttons, and `courier font` for files and folders.

3. Methods

The methods outlined next describe TRNA of biologically associated genes using PAINT. Genes are typically associated by highly parallel experimental approaches such as microarray-based gene-expression analysis or proteomic analyses. However, excellent results have been obtained by creating gene lists from extant literature by manual searches or computationally derived results from "knowledge database" searching.

3.1. Identification of Overrepresented TF-Binding Sites Using PAINT

A typical scenario of using PAINT is to study a group of genes identified or expected to be coregulated under specific experimental conditions. PAINT is used to investigate whether these genes share any TF-binding sites in their promoters and if such a shared coincidence of binding sites is significantly higher than random frequency as determined by Fisher's exact test.

3.1.1. Generation of PAINT-Compatible Input File

The starting point to PAINT is a file containing the list of genes under investigation. The file should be a single column plaintext file, each row listing a gene identifier. All the identifiers in the file need to be of the same type, for example, Genbank accession number. An example gene list file (named

`exGeneList.txt`) is available online at http://www.dbi.tju.edu/dbi/publications/MiMBchapter/.

3.1.2. TRNA Using PAINT

In this step, the gene list from **Subheading 3.1.1.** will be used to retrieve promoter sequences, analyze them using TRANSFAC Public, build a Feasnet, and analyze the resultant Feasnet as compared with a reference Feasnet to derive hypotheses on overrepresented TREs.

1. Use a web browser to open the web page http://www.dbi.tju/dbi/tools/paint/.
2. Follow the link "Start New Analysis" on the main page.
3. Select **Mus Musculus (mouse)** as the ***Organism Name***, **2000** for the ***Desired upstream length***, **Accession Number** for ***Gene Identifier type***, **Gene Identifiers List** for ***Upload text file of type***. Refer to the **Notes 4.1** and **4.2** for issues involved in the selection of the gene identifiers and the size of the gene list.
4. Click the **Browse** button to locate and select the file `exGeneList.txt` on the computer.
5. Ensure that the checkbox next to **TFRetriever** is selected.
6. Select **MATCH (TRANSFAC Public)** for ***TRE finding program***. Refer to **Note 4.3** on the issues involved in the choice of the TRE-finding programs.
7. Enter the user name and password for logging into the website http://www.gene-regulation.com.
8. Select **Minimize False-Positives** for the ***MATCH filter option***.
9. Select **1.00** for the ***Core similarity threshold***. Check the box for ***Find TREs on complementary strand?***
10. Click the button **Execute Feasnet Builder** at the end of the form. A new page will be loaded indicating the status of the analysis. Note down the job key at the top of the status page for later access, as the analysis might take considerable time depending on the size of the gene list.
11. Once the FeasnetBuilder analysis is complete, the highlighted status text at the top of the page will be replaced by a link to the ZIP file containing all the results including the status page.
12. After completion of FeasnetBuilder, the status page indicates the number of promoters that were retrieved (refer to **Note 4.4** on how redundancy in the gene list is handled), the promoter sequences in the FASTA format, and also a link to a list of genes for which the promoter sequences were not found in the database. Next, the page indicates whether the gene list was split into multiple parts for processing using MATCH. Links to the actual HTML output from MATCH are provided next to each of the split sequence files. Last, the overall Feasnet corresponding to the input gene list is given next to the text ***Feasnet file***.
13. After completion of the FeasnetBuilder step, the status page contains a link to the follow-up overrepresentation analysis and visualization. Click on the link indicated by the text ***Click here to continue with Feasnet Analysis and Visualization***.

14. On the analysis page, the parameters corresponding to the ***Feasnet, Organism, Upstream sequence length, Gene Identifier type, TRE finding program, Core similarity threshold, TREs on the complementary strand included?***, will be automatically set.
15. Under ***Clustering Options***, check both the boxes corresponding to ***TREs based on the promoters they are present on*** and ***genes based on the TREs present on their promoters***.
16. Under ***Select Reference Feasnet(s) for significance analysis of TREs***, check the box corresponding to ***All promoter sequences in PAINT database***. Refer to **Note 4.5** for additional information on how to choose appropriate reference feasnet.
17. Check the box next to ***Generate filtered Gene-TRE networks based on TRE over-representation***. Under this text, select **0.30** for the parameter ***Only those TREs of FDR-based adjusted p-value <=***, and **0.05** for the parameter ***Only those TREs of raw p-value <=***. Refer to **Note 4.6** for information on these two thresholds used in the analysis.
18. Click the **Execute Feasnet Analyzer/Viewer** button at the end of the form. A new page will be loaded indicating the status of the analysis. The job key will be same as earlier, as this is merely continuation of the analysis.
19. Once the analysis and visualization is complete, the highlighted status text at the top of the page will be replaced by a link to the ZIP file containing all the results including the status page.
20. The results from the overrepresentation analysis are under the heading ***Significance of TRE occurence (input list compared with a reference)***. Links to the specific reference used, *p*-values for overrepresentation, and the Feasnet images are provided. Under the subheading ***Hypothesis Gene-TRE network***, links are provided to the filtered Feasnet data and images based on the specified *p*-value thresholds (**0.30** and **0.05** in **step 17**). Network images and Graphviz source file are also given. Refer to **Note 4.7** for information on how to interpret the PAINT results.

3.2. Combining Coexpression Clustering Information With TRNA Using PAINT (Optional)

PAINT can also be used to simultaneously analyze multiple groups of genes (e.g., cluster analysis of multicondition microarray data). In this case, the over-representation analysis is performed for each individual cluster as compared with the specified reference as well as with the entire input list itself (i.e., all clusters combined).

3.2.1. Generation of PAINT-Compatible Cluster Information File

Cluster membership data file is a user-provided tab-delimited plain text file with two columns. The first column must contain one gene identifier per row and the second column must contain a corresponding single word alphanumeric cluster label. For example, consider a scenario in which Multiexperiment Viewer

(http://www.tm4.org) is used for cluster analysis. After clustering is performed, save each cluster table into a separate text file. Copy the gene lists from each file into a single column in a spread sheet, one file at a time. Each time, add a cluster label (e.g., A, B, C, and so on) in a second column for all the gene identifiers that are copied from a single cluster table. An example file named `exGeneClusterInfo.txt` containing cluster information in the specified format is available online at http://www.dbi.tju.edu/dbi/publications/MiMBchapter/.

3.2.2. Combining Cluster Membership Information With TRNA

1. Follow the steps in **Subheading 3.1.2.** until **step 17**.
2. After **step 17**, click the **Browse** button for the parameter ***Gene cluster information file*** to locate and select the `exGeneClusterInfo.txt` file.
3. Click the **Execute Feasnet Analyzer/Viewer** button at the end of the form. A new page will be loaded indicating the status of the analysis.
4. Once the analysis and visualization is complete, the highlighted status text at the top of the page will be replaced by a link to the compressed file containing all the results including the status page.
5. The results from the overrepresentation analysis are under the headings ***Significance of TRE occurrence (in clusters compared with a reference)*** and ***Significance of TRE occurrence (in individual clusters compared with the list)***. Links to the specific reference used, *p*-values for overrepresentation, and the Feasnet images are provided. Under the subheading ***Hypothesis Gene-TRE network***, links are provided to the filtered Feasnet data and images based on the specified *p*-value threshold (**0.10** in **step 17**). Network image and Graphviz source file are also given. Refer to **Notes 4.7** for information on how to interpret the PAINT results.

4. Notes

4.1. Selection of Gene Identifiers

A key issue that is often underappreciated is that of gene identifiers used in TRNA. Typically, if the gene list is derived from a microarray data set, then the most *convenient and proper* gene identifiers to use in PAINT are the corresponding Clone IDs or Genbank accession numbers. PAINT uses UniGene database to map the Clone IDs to the corresponding Entrez gene IDs (used to be named LocusLink) and then utilize the Ensembl cross-reference annotation information to obtain the corresponding unique set of Ensembl gene IDs. Because UniGene annotation is regularly updated and given that UniGene cluster IDs are not guaranteed to be stable across different UniGene releases, the use of UniGene IDs as gene identifiers is not permitted in TRNA using PAINT. In cases wherein the gene list is manually derived from previous knowledge of regulation, for example, all genes implicated in a particular cellular function, then the most *convenient and proper* gene identifiers to use in PAINT are the corresponding Entrez gene IDs.

4.2. Size of the Gene List

Another key issue in TRNA is the size of the gene list. Based on the results from multiple studies, it is recommended that the gene list correspond to at least 30 genes. Whereas a formal assessment of the robustness of PAINT to "noise" in the gene list (i.e., containing genes that do not "belong" in the coregulated set) has not been made, available experience on multiple data sets indicate that the results are not critically dependent on 100% accuracy of the gene list corresponding to truly coregulated genes. This has a significant impact in the cluster-based analysis, so that small inaccuracies (<10%) in the clustering algorithms do not significantly influence the results from TRNA.

4.3. Selection of TRANSFAC Version

To utilize much of PAINT functionality, users need to obtain appropriate licensed access to the public or professional versions of TRANSFAC database. The public version is hosted at http://www.gene-regulation.com (not affiliated with the PAINT team) and is available following a free registration process at http://www.gene-regulation.com (not affiliated with the PAINT team). Access to commercial version is available through http://www.biobase-international.com (not affiliated with the PAINT team). The login and password required in the analysis step are only used to interact with the appropriate web servers. This ensures proper handling of the license management issues whereas providing an option to PAINT users. The professional version of TRANSFAC contains substantially higher number of TREs and TF than the public version, and hence, the use of the former significantly improves the TRNA.

4.4. Multiple Promoters and Redundancy in the Gene List

It is likely that several gene identifiers in the input gene list (with the exception of Ensembl gene IDs) map to same Ensembl gene. The Upstreamer module builds the entire cross-referenced list of Ensembl genes that corresponds to the input gene list and then makes the resultant Ensembl gene ID list unique before proceeding with the TFRetriever step. In addition, owing to the nature of the cross reference in the Entrez gene and Ensembl databases, it is likely that a few of the gene identifiers in the input gene list (with the exception of the Ensembl gene IDs) individually map to more than one Ensembl gene. In these cases, PAINT includes all the mapped Ensembl genes in the analysis.

4.5. Selection of Reference Feasnet

The selection of appropriate reference set is the key to derive meaningful hypotheses in TRNA. Comparison of the experiment Feasnet to the entire genome gives erroneous results if the input gene list is obtained from a microarray that

does not span the entire genome or is specific to a particular tissue/disease. In most of the cases, the microarray gene list is first processed in the Feasnet Builder to obtain a microarray Feasnet. When analyzing the experimental gene lists (e.g., differentially expressed genes from a microaray experiment), this microarray Feasnet should be utilized as the "reference Feasnet" in **step 16** (*see* **Subheading 3.1.2.**). However, the choice of reference set does not end with using the reference Feasnet from the microarray gene list. For example, in comparison of an early upregulated gene set to the set of all upregulated genes the significantly enriched TREs point to those that are characteristic of the early upregulated genes relative to all the upregulated genes. If the input gene list is that of entire differentially expressed genes in an experiment, Feasnet from each gene cluster in the input list (typically, with specific-expression profile or function) can be compared with that of the input list itself. Such a "cluster-to-list" comparison can reveal TREs that are differentially specific to each gene cluster. TRNA using PAINT is based on multiple results arising from such comparisons, for TRE enrichment to derive specific regulatory network hypotheses. The Feasnet analysis and visualization step can be repeated multiple times by considering different gene cluster combinations such as those based on different clustering of expression pattern, biological function from gene ontology, or pathway data.

4.6. Multiple Testing Correction Using an FDR Estimate

In PAINT, the raw *p*-values in each overrepresentation analysis are corrected for multiple testing using a FDR estimate ***(29)***. As a first option, the results from the FDR-based, adjusted *p*-values should be used in identifying the significantly overrepresented TREs. However, in some cases, this particular correction is either inappropriate (e.g., if raw *p*-values do not follow a β-uniform distribution) or overconservative (owing to correlations among TREs). It is likely that filtering the FDR-based multiple testing corrected *p*-values yields little or no results. Hence, the Feasnet analysis and visualization step in PAINT includes filters for both the adjusted and raw *p*-values. In cases wherein the former yields no results, one can utilize the latter to follow a discovery approach to derive TRE hypotheses for further experimental validation. Whereas this alternative may result in a set of hypotheses that can potentially contain 100% false-positives in the extreme case (from the multiple testing perspective), in practice, this amounts to prioritizing the validation experiments based on individually enriched TREs. Considering that the primary role of any computational analysis is in generating candidates for further experimental validation, in cases whereby multiple testing correction yields little or no results the alternative raw *p*-value based approach is the next best option.

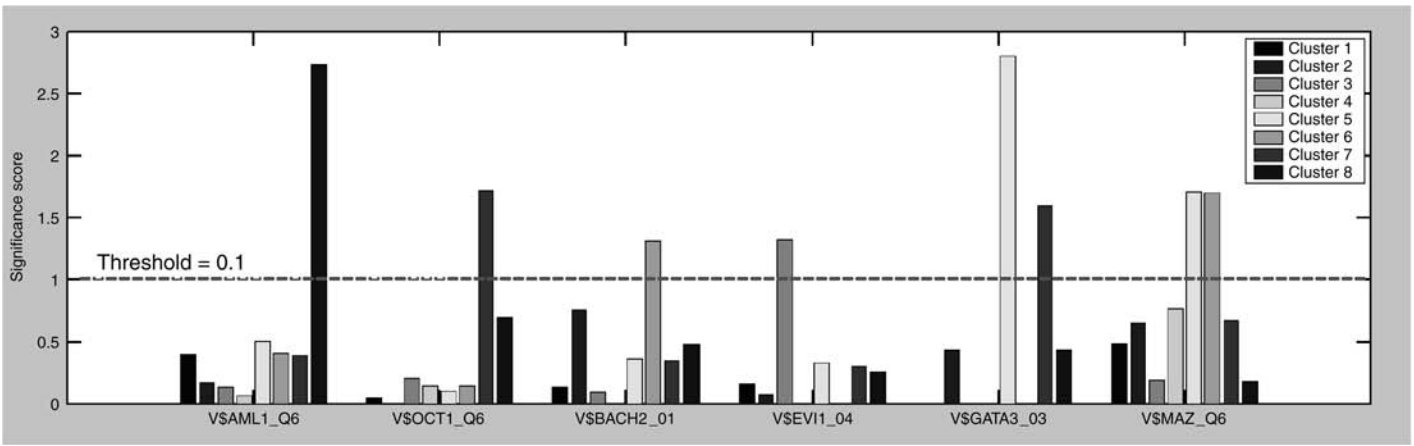


Fig. 4. Localization of enrichment of a subset of six TREs to specific clusters. The significance score is calculated as ($-\log_{10}[p]$). Score = 1 indicates the *p*-value threshold of 0.1 (dashed line).

4.7. Interpreting the PAINT Results

Hypothesized gene-TRE network from the enrichment analysis **(step 20)** indicates those TREs that are significantly overrepresented in the promoters of the entire input gene list as compared with the promoters in the reference (all promoters in PAINT promoter database). The significantly above-random nature of occurrence of certain TREs makes these ideal candidates for further experimental validation. Additional association of subsets of the master list of genes, after the first cutoff of "differentially expressed" is applied, is often used to identify subsets of genes sharing some type of more detailed behavior, typically either by coexpression grouping by any of a plurality of clustering algorithms or by functional grouping, for example, in conjunction with the gene ontology annotation. When using PAINT for TRNA, these subgroupings can be used to determine if any of the binding sites found on promoters of differentially expressed genes are diagnostic for any specific gene behavior (coexpression cluster, functional subgroup, and so on). Therefore, the desired result would be identification of a TRE determined to be statistically enriched in one or a few of the subgroups, but not all. When a cluster membership file is provided, PAINT will generate visualizations with the genes reordered into their respective groups based on the list order. Thus, group-enriched TREs will appear on the Feasnet image as a vertical collection of red boxes, which mirror the limits of the gene list for the group. The enrichment can be more easily visualized by graphing the –log(*p*-value) for each TRE of interest for each subgroup as shown in **Fig. 4**. The enrichment *p*-values of TREs in each subgroup can be obtained from the ***Significance of TRE occurrence (in clusters compared with a reference)*** section of the PAINT output, following the link ***Overrepresentation*** in either raw or FDR-adjusted *p*-values. In the example depicted in **Fig. 4**, the TRE V$AML1_Q6 is significantly enriched in only group 8, whereas the TRE V$GATA3_03 is enriched in both group 5 and group 7. The resultant biological inference is that these specific TREs, and their cognate TF, are specifically involved in the regulation of that subgroup of genes.

Acknowledgments

The authors acknowledge the financial support for this work from DARPA BioCOMP initiative under the project "Multi-Timescale Complex Adaptation," no. F30602-01-2-0578, PI: James Schwaber.

References

1. Liang, P. and Pardee, A. B. (1992) Differential display of eukaryotic messenger RNA by means of the polymerase chain reaction. *Science* **257,** 967–971.
2. Velculescu, V. E., Zhang, L., Vogelstein, B., and Kinzler, K. W. (1995) Serial analysis of gene expression. *Science* **270,** 484–487.

3. Okubo, K., Hori, N., Matoba, R., et al. (1992) Large scale cDNA sequencing for analysis of quantitative and qualitative aspects of gene expression. *Nat. Genet.* **2,** 173–179.
4. Lockhart, D. J., Dong, H., Byrne, M. C., et al. (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat. Biotechnol.* **14,** 1675–1680.
5. Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270,** 467–470.
6. Hughes, T. R., Mao, M., Jones, A. R., et al. (2001) Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. *Nat. Biotechnol.* **19,** 342–347.
7. Steffen, M., Petti, A., Aach, J., D'haeseleer, P., and Church, G. (2002) Automated modeling of signal transduction networks. *BMC Bioinformatics* **3,** 34.
8. Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F. (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* **18,** S233–S240.
9. Zak, D. E., Doyle, F. J., III., Gonye, G. E., and Schwaber, J. S. (2001) Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data. *Proc. Second Int. Conf. Syst. Biol.* 231–238.
10. Zak, D. E., Gonye, G. E., Schwaber, J. S., and Doyle, F. J., III. (2003) Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: Insights from the identifiability analysis of an *in silico* network. *Genome Res.* **13,** 2396–2405.
11. Vadigepalli, R., Chakravarthula, P., Zak, D. E., Schwaber, J. S., and Gonye, G. E. (2003) PAINT: a promoter analysis and interaction network generation tool for gene regulatory network identification. *Omics* **7,** 235–252.
12. Vadigepalli, R., Hao, H., Miller, G. M., Liu, H., and Schwaber, J. S. (2006) EGFR-induced circadian-time dependent gene regulation in suprachiasmatic nucleus. *Neuroreport* **17(13),** 1437–1441.
13. Addya, S., Keller, M. A., Delgrosso, K., et al. (2004) Erythroid-induced commitment of K562 cells results in clusters of differentially expressed genes enriched for specific transcription regulatory elements. *Physiol. Genomics* **19,** 117–130.
14. Pratt, C., Vadigepalli, R., Chakravarthula, P., Gonye, G. E., Philip, N. J., and Grunwald, G. B. (2007) Transcriptional regulatory network analysis during epithelial-mesenchymal transformation of retinal pigment epithelium. Molecular Vision (in press).
15. Stevens, S. L., Gopalan, B., Minami, M., et al. (2004) LPS preconditioning provides neuroprotection through reprogramming of cellular responses to stroke. *Soc. Neurosci.* Abstract 457.14.
16. Zak, D. E., Hao, H., Vadigepalli, R., Miller, G. M., Ogunnaike, B. O., and Schwaber, J. S. (2006) Systems analysis of circadian time dependent neuronal epidermal growth factor receptor signaling. *Genome Biol.* **7(6),** R48.
17. Saban, M. R., Hellmich, H. L., Turner, M., et al. (2006) The inflammatory and normal transcriptome of mouse bladder detrusor and mucosa. *BMC Physiol.* **6(1),** 1.

18. Dozmorov, M. G., Kyker, K. D., Saban, R., et al. (2006) Analysis of the interaction of extracellular matrix and phenotype of bladder cancer cells. *BMC Cancer* **6,** 12.
19. Keller, M. A., Addya, S., Vadigepalli, R., Banini, B., et al. (2006) Transcriptional regulatory network analysis of developing human erythroid progenitors reveals patterns of co-regulation and potential transcriptional regulators. *Physiol. Genomics* **28,** 114–128.
20. Hubbard, T., Barker, D., Birney, E., et al. (2002) The Ensembl genome database project. *Nucleic Acids Res.* **30,** 38–41.
21. Kawai, J., Shinagawa, A., Shibata, K., et al. (2001) Functional annotation of full-length mouse cDNA collection. *Nature* **409,** 685–690.
22. Aerts, S., Thijs, G., Coessens, B., Staes, M., Moreau, Y., and Moor, B. D. (2003) Toucan: deciphering the cis-regulatory logic of coregulated genes. *Nucleic Acids Res.* **6,** 1753–1764.
23. Matys, V., Frickle, E., Geffers, R., et al. (2003) TRANSFAC®: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res.* **31,** 374–378.
24. Quandt, K., Frech, K., Karas, H., Wingender, E., and Werner, T. (1995) MatInd and MatInspector—New fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Res.* **23,** 4878. A web-based version of the tool is available at http://www.genomatix.de. (last accessed May 5th, 2007).
25. Bailey, T. L. and Elkan, C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Second Int. Conf. Intelligent Syst. Mol. Biol.* **28–36,** AAAI Press, Menlo Park, California. http://meme.sdsc.edu (last accessed May 5th, 2007).
26. Bury, K. (1999) *Statistical Distributions in Engineering*. Cambridge University Press, Cambridge, UK.
27. Jakt, L. M., Cao, L., Cheah, K. S., and Smith, D. K. (2001) Assessing clusters and motifs from gene expression data. *Genome Res.* **11(1),** 112–123.
28. Elkon, R., Linhart, C., Sharan, R., Shamir, R., and Shiloh, Y. (2003) Genome-Wide in silico identification of transcriptional regulators controlling the cell cycle in human cells. *Genome Res.* **13,** 773–780.
29. Benjamini, Y. and Hochberg, Y. (1995) Controlling the False Discovery Rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B* **57,** 289–300.

5

Prediction of Intrinsic Disorder and Its Use in Functional Proteomics

Vladimir N. Uversky, Predrag Radivojac, Lilia M. Iakoucheva, Zoran Obradovic, and A. Keith Dunker

Summary

The number of experimentally verified, intrinsically disordered (ID) proteins is rapidly rising. Research is often focused on a structural characterization of a given protein, looking for several key features. However, ID proteins with their dynamic structures that interconvert on a number of time-scales are difficult targets for the majority of traditional biophysical and biochemical techniques. Structural and functional analyses of these proteins can be significantly aided by disorder predictions. The current advances in the prediction of ID proteins and the use of protein disorder prediction in the fields of molecular biology and bioinformatics are briefly overviewed herein. A method is provided to utilize intrinsic disorder knowledge to gain structural and functional information related to individual proteins, protein groups, families, classes, and even entire proteomes.

Key Words: Intrinsically disordered protein; natively unfolded protein; intrinsically unstructured protein; protein flexibility; disorder prediction; protein function.

1. Introduction

Although the protein sequence-structure-function paradigm (well known as the "lock-and-key" hypothesis ***[1]***), according to which a protein can achieve its biological function only on folding into a unique, structured state determined by its amino acid sequence, was a dominating view for more than 100 yr, it is recognized now that the phenomenon of functional intrinsic disorder is highly abundant in nature. For example, only less than one-third of the crystal structures in the protein data bank (PDB) are completely devoid of disorder ***(2)***. In fact, recent discoveries of intrinsically disordered (ID) or natively unstructured proteins have significantly broadened the understanding of protein functionality and revealed a new and unexpected role of dynamics, plasticity, and flexibility

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

in protein function. Importantly, bioinformatics played a key role in transforming a set of anecdotal examples of intrinsically disordered proteins (IDPs), which were originally considered to be intriguing exceptions within the protein realm, into a very promising branch of protein science.

1.1. Defining and Identifying IDPs

IDPs or ID protein regions are those that fail to form specific three-dimensional (3D) structure under physiological conditions in vitro. They are also known as partially folded ***(3)***, flexible ***(4)***, mobile ***(5)***, rheomorphic ***(6)***, natively denatured ***(7)***, natively unfolded ***(8)***, intrinsically unstructured ***(9)***, ID ***(10)***, and natively disordered ***(11)***. Furthermore, several other names representing different combinations of "natively, naturally, and intrinsically" with "unfolded, unstructured, flexible, mobile, and denatured," are present in literature ***(12)***. The interested reader will find the discussion of the etymology of the term "ID" in a recent review ***(13)***. In contrast to the ordered proteins, the atoms and dihedral angles of IDPs do not have equilibrium positions. Instead, IDPs exist as highly dynamic ensembles whose atoms and backbone Ramachandran angles fluctuate significantly over time. An ID region can be as short as a few amino acid residues, or it can propagate through the long disordered loops, ends, domains, or even through entire proteins ***(13)***. In the authors' view, an IDP is a protein that contains at least one disordered region.

Functional ID regions exist in at least two different structural forms: molten globule-like (collapsed) and random coil-like (extended) ***(14)***. Later, the existence of another functional disordered form, the premolten globule, which appears to be a distinct category between extended and molten-globular conformations, was suggested ***(15)***. Thus, protein function might be associated with three (or four) distinct conformations: ordered, molten globule, (premolten globule), and random coil, and with the transitions between them. These hypotheses are known as the protein-trinity ***(14)*** or protein-quartet models ***(15)***.

IDPs can be identified by the variety of physicochemical methods elaborated to characterize protein structure and self-organization. These methods include missing electron density in X-ray crystallography maps ***(16)***; nuclear magnetic resonance spectroscopy (for recent reviews *see* **refs. *11*** and ***17–20*** and references therein); circular dichroism spectroscopy in the near-ultraviolet ***(21)*** and far-ultraviolet regions ***(22–25)***; optical rotatory dispersion spectroscopy (ORD) ***(22,25)***; Fourier transform infased spectroscopy (FTIR) ***(25)***; Raman spectroscopy and Raman optical activity ***(26)***; fluorescence spectroscopy ***(27,28)***; gel-filtration, viscometry, small-angle X-ray scattering, small-angle neutron scattering, sedimentation, and dynamic and static light scattering ***(27–29)***; limited proteolysis ***(30–34)***; aberrant mobility in sodium dodecyl sulfate-gel electrophoresis ***(35,36)***; conformational stability ***(27,37–40)***; hydrogen/deuterium exchange (H/D exchange) ***(28)***; immunochemical methods ***(41,42)***; interaction with molecular chaperones

(27); and electron microscopy or atomic force microscopy. Interested readers can find more detailed description of these approaches in several recent reviews *(11,15,18,28)*.

1.2. Functional Repertoire of Intrinsic Disorder

Ordered proteins have evolved to carry out efficient catalysis and to bind specific ligands. This is complemented by the functional repertoire of IDPs, which are typically involved in regulation, signaling, and control pathways *(13,43,44)*. Using literature searches, Dunker et al. *(45)* cataloged 90 proteins in which disordered regions were functionally annotated. This group of disordered regions contained 28 specific functions, organized into four functional classes: (1) molecular recognition, (2) molecular assembly, (3) protein modification, and (4) entropic-chain activities *(45)*. Lately, this repertoire was significantly extended applying a novel bioinformatics tool to find functions associated with ID regions *(46–48)*. Using this approach it has been established that out of the 710 Swiss-Prot functional keywords that were associated with at least 20 proteins, 238 were found strongly positively correlated with long ID regions, whereas 302 were strongly negatively correlated *(46–48)*.

1.3. DisProt: A Database of IDPs

Although the first public resource containing disordered protein regions, the ProDDO database, was developed in 2001 *(49)*. This database did not provide information about type of disorder nor the function of disordered regions. Furthermore, it was not curated, being limited to the PDB entries only. These limitations were overcome by currently the most complete database of experimentally characterized disordered proteins, DisProt *(50)*. This database, which can be accessed through http://www.disprot.org, provides structural and functional (wherein available) information on experimentally characterized IDPs. As of June 2006, the database contained information on 458 proteins (1096 disordered regions).

1.4. Predicting ID Regions

As already emphasized, bioinformatics played a crucial role in the development of the IDP field. Already at the early stage of the field, simple statistical comparisons of amino acid compositions and sequence complexity indicated that disordered and ordered regions are different to a significant degree. In fact, based on the analysis of 150 ID segments and comparison of these segments with ordered proteins it has been suggested that the amino acids can be grouped into order promoting (C, F, I, L, F, N, V, W, and Y), disorder promoting (A, E, G, K, P, Q, R, and S), and neutral (D, H, M, and T) *(10)*. Several subsequent studies followed up this analysis using increasingly larger data sets *(51–54)*. In addition to the first-order statistics, recent studies also addressed higher-order patterns in amino acid

sequence space and analyzed the space of various physicochemical properties ***(55)***, confirming the existence of several biases in IDP sequences. The mentioned sequence biases were exploited to develop a multitude of highly accurate predictors of ID regions, which then were used to estimate the commonness of IDPs in the three kingdoms of life, as well as to elaborate first identifiers of IDP function.

The first predictor of ID regions was reported in 1997 ***(54)***. This two-layer feed-forward neural network, which achieved a surprising accuracy of about 70% clearly marked the beginning of a new epoch by showing that (1) there are significant compositional differences between ordered and ID protein regions, (2) the lack of fixed protein 3D structure is predictable from amino acid sequence alone, and (3) ID regions of different lengths (short, medium, and long) may be compositionally different from each other. The predictive model was later extended to the VLXT predictor ***(51)***, which is a combination of the VL1 and XT predictors ***(56)***. The letters describe the amino acids used for training, where VL stands for Variously-characterized Long disordered internal regions and XT stands for X-ray characterized Terminal regions. The VLXT designation is preceded by a descriptive prefix, Predictor of Natural Disordered Regions (PONDR) giving PONDR VLXT.

In 2000, it was noticed that natively unfolded proteins can be separated from ordered proteins by considering their average net charge and hydropathy ***(25)***. This observation led to the development of a simple binary classifier, the charge-hydropathy plot (CH-plot) ***(25)***, which was based on the analysis of the amino acid composition and instead of predicting ID on a per residue basis, classified entire protein as compact or natively unfolded. Another binary classifier is the cumulative distribution functions (CDF) analysis of disorder scores, which separates ordered and disordered sequences based on the per-residue disorder score retrieved by PONDR VLXT, and the optimal boundary ***(57,58)***. This method summarizes the per-residue predictions by plotting PONDR scores against their cumulative frequency, which allows ordered and disordered proteins to be distinguished based on the distribution of prediction scores.

Later, more sophisticated methods based on various statistical and machine-learning techniques (including bagging and boosting ***[59]*** and linear regression model for the prediction of long disordered regions ***[60]***) emerged, culminating in the inclusion of the disorder prediction as a separate category in the Critical Assessment of (protein) Structure Prediction (CASP) experiments ***(61,62)***. **Table 1** presents the information related to those ID predictors that are scientifically novel and/or published. These predictors are briefly outlined as follows:

1. DISOPRED ***(63)*** is a neural network classifier trained on the position-specific scoring matrices and combined disorder prediction with the predictor of secondary structure ***(64)***.
2. PONDR VL3 is an ensemble of feed-forward neural networks that uses evolutionary information and is trained on long disordered regions ***(65)***.

Table 1
Summary of the Web Servers Offering Prediction of Intrinsically Disordered Proteins

Server name	URL	Approach	References
VLXT (PONDR)	http://www.pondr.com	Feed-forward neural network with separate N-/C-terminus predictor. Based on amino acid compositions and physicochemical properties	***51,54,56***
FoldIndex©	http://bip.weizmann.ac.il/fldbin/findex	Charge/hydrophobicity score based on a sliding window	***25,74***
NORSp	http://rostlab.org/services/NORSp/	Rule-based using a set of several neural networks. Amino acid compositions and sequence profiles used as features	***68,69***
VL2/VL3	http://www.ist.temple.edu/disprot/predictor.php http://www.pondr.com	Ordinary least-squares linear regression (VL2) and bagged feed-forward neural network(VL3). All models use amino acid compositions and sequence compl exity. VL3 series uses sequence profiles	***2,60,65***
DISOPRED	http://bioinf.cs.ucl.ac.uk/disopred/	Feed-forward neural network (DISOPRED) and linear support vector machine (DISOPRED2) based on sequence profiles	***63,70,71***
GlobPlot	http://globplot.embl.de/	Autoregressive model based on amino acid propensities for disorder/globularity	***66***
DisEMBL™	http://dis.embl.de/	Ensemble of feed-forward neural networks	***67***
IUPred	http://iupred.enzim.hu/index.html	Linear model based on the estimated energy of pairwise interactions in a window around a residue	***72,73***

(Continued)

Table 1 ***(Continued)***

Server name	URL	Approach	References
PreLink	http://genomics.eu.org/spip/PreLink	Rule-based. Ratio of multinomial probabilities (for linker and structured regions) combined with the distance to the nearest hydrophobic cluster	***76***
RONN	http://www.strubi.ox.ac.uk/RONN	Feed-forward neural network in the space of distances to a set of prototype sequences of known fold state	***75***
DISpro	http://www.igb.uci.edu/servers/psss.html	Recursive neural network based on sequence profiles, predicted secondary structure and relative solvent accessibility	***77***
VSL	http://www.ist.temple.edu/disprot/predictorVSL2.php	Logistic regression (VSL1) and linear support vector machine (VSL2) based on sequence composition, physicochemical properties, and profiles. Combination of short and long disorder predictors	***78,79***
DRIP-PRED	http://www.sbc.su.se/~maccallr/disorder/	Kohonen's self-organizing maps based on sequence profiles	–
SPRITZ	http://protein.cribi.unipd.it/spritz/	Nonlinear support vector machine based on multiply aligned sequences. Separate predictors for short and long disorder regions	***80***

3. GlobPlot is based on derived amino acid propensities for disordered regions ***(66)***. DisEMBL server uses a support vector machine ***(67)***, trained on three proposed types of disorder: (1) loops/coil, i.e., structured regions missing regular secondary structure of helix and strand, (2) hot-loops, i.e., structured regions other than helix or strand, but having high C_α B-factors, and (3) remark465, i.e., regions with missing electron density from PDB.
4. NORS predictor identifies regions with nonregular secondary structure ***(68,69)***.
5. DISOPRED2 uses linear support vector machines ***(70,71)***.
6. IUPred is based on energy-derived coefficients ***(72,73)***.
7. FoldIndex ***(74)*** is based on the CH approach developed by Uversky et al. ***(25)*** and extended to calculations over a sliding window to achieve residue-based predictions.
8. RONN, a regional-order neural network, classifies residues in the space of distances between an input sequence and a set of carefully selected "prototype" sequences ***(75)***.
9. PreLink uses compositional bias and lack of hydrophobic clusters ***(76)***.
10. DISpro uses large 1D recursive neural networks trained with a variety of compositional, evolutionary, and derived attributes ***(77)***.
11. PONDR VSL incorporates the ideas of training separate models for short- ***(53)*** and long disordered regions ***(65)***, with subsequent combination of these models through a separately trained model ***(78,79)***.
12. SPRITZ uses nonlinear support vector machines for short- and long disorder regions based on multiply aligned sequences ***(80)***.

Recently, predictors of intrinsic disorder have been used to find functional regions in IDPs. In fact, short regions of predicted order bounded by extended regions of predicted to be disordered by PONDR VLXT, were shown in several cases to identify binding sites that involved disorder-to-order transitions on complex formation ***(81)***. These structures, which contained short regions of proteins bound to their partners, showed that the PONDR-indicated region often formed a helix, on binding to its partner. Many examples of these binding sites are found in the PDB ***(82)***. The pattern in the PONDR VLXT curve reveals short regions that undergo disorder-to-order transitions on binding. Additionally, these regions tend to have predictions of helix as well as hydrophobic moments. From such characteristics, a predictor of helix-forming molecular recognition features (α-MoRF) was developed ***(82)***.

Finally, it has been reported that amino acid compositions, sequence complexity, hydrophobicity, charge, and other sequence attributes of regions adjacent to phosphorylation sites are very similar to those of IDP regions ***(83)***. These observations were utilized in the development of a new web-based tool for the prediction of protein phosphorylation sites, disorder-enhanced phosphorylation predictor (DisPhos or DEPP), the accuracy of which reaches 76% for serine, 81% for threonine, and 83% for tyrosine ***(83)***.

1.5. When to Use the ID Predictions

In this section some indicators have been outlined regarding when to use the ID predictions, both for the individual protein analysis and for the large-scale studies.

1. ID predictions are priceless for the analysis of individual proteins. These predictions help to better understand and interpret experimental data (e.g., a monomeric protein predicted to be natively unfolded possesses large hydrodynamic volume. Such an unexpectedly large hydrodynamic dimension, being observed experimentally might be incorrectly interpreted in terms of oligomer formation if the protein was assumed to be globular). Such predictions also help to classify proteins and to understand their functionalities. This derives from the observation that the functional repertoires of ordered proteins and IDPs are extremely different. Therefore, knowing that the protein of interest is ID might help redirect its structural and functional analysis. The disorder predictions aided in structural characterization of the retinal tetraspanin ***(84)***, nicotinic acetylcholine receptor ***(85)*** Dribble, a member of the conserved Krr1P protein family ***(86)*** proapoptotic Bcl-2 homology domain-containing family of proteins ***(87)***, transcriptional corepressor CtBP ***(88)***, notch-signaling pathway proteins ***(89,90)***, and many others.
2. Utilizing bioinformatics tools based on ID phenomenon one might find potential protein–protein and protein–nucleic acid interaction sites (molecular recognition fragments) and identify potential sites of posttranslational modifications. This knowledge can be used to drive subsequent research with the major focus on finding binding partners, analysis of resulting complexes, and searching for small molecules modulating these interactions.
3. The majority of ID predictors are based on rather large training sets, which makes prediction of intrinsic disorder in a given protein fairly certain. An ID prediction also means that the analyzed protein is statistically similar to those used in the training of the ID predictors, thus indicating that a particular protein is not an exception, but a rule.
4. ID predictors are indispensable in estimating the commonness of protein disorder in large data sets. They allow scientifically sound extrapolation of knowledge gained on the basis of a few examples to collections including hundreds or even thousands of proteins. For example, proteins associated with cancer ***(43)*** and cardiovascular disease ***(91)*** were shown to be enriched in intrinsic disorder. ID was shown to be highly abundant in signaling proteins ***(43)***, transcription factors ***(92)***, proteins with PEST regions (e.g. regions rich in proline, glutamate, serine and threonine) ***(93)***, histones ***(94)***, serine/arginine-rich splicing factors ***(95)***, partners of 14-3-3 proteins ***(96)***, nucleoporins ***(97)***, and several other sets of proteins with different functions.

Finally, disorder prediction is crucial for protein crystallization and structural genomics projects. Disordered regions are generally not compatible with the crystallization process. Therefore, close examination of sequences that failed to crystallize may reveal ID regions interspersed with regions of order. Thus, accounting for protein disorder can improve target selection and prioritization for the structural genomics projects.

2. Materials

1. The Swiss-Prot database is described in **ref.** ***98*** and is available from http://www.expasy.org/sprot/.
2. The database of experimentally characterized disordered proteins, DisProt, is available from http://www.disprot.org. The original version of this database is described in **ref.** ***50***.
3. PONDRVLXT predictor is described in **ref.** ***51*** and is availal from http://www.pondr.com/.
4. PONDR VL3-BA is described in **ref.** ***65*** and is available from http://www.pondr.com/.
5. PONDR VSL is described in **refs.** ***78*** and ***79*** and is available from http://www.pondr.com/.
6. CH-plot predictor is available from http://www.pondr.com/. The basic algorithm of this binary classifier is described in **ref.** ***25***.
7. CDF analysis is available from http://www.pondr.com/. This predictor is described in **refs.** ***57*** and ***58***.
8. α-MoRF predictor is described in **ref.** ***82*** and is available from http://www.pondr.com/.
9. DisPhos predictor also known as DEPP is described in **ref.** ***83*** and is available from http://www.pondr.com/.

3. Methods

The methods outlined next describe the analysis of amino acid sequences using the intrinsic disorder knowledge to gain structural and functional information related to a protein, a protein family, or an entire proteome/database. Although numerous predictors of intrinsic disorder are currently available as web servers (*see* DisProt website, http://www.disprot.org, for a complete list of such servers), focus will be on utilization of PONDR tools, as they cover a wide range of potential applications of ID concept for structural and functional analysis of proteins. Obviously, this analysis could have been carried out with other ID predictors described earlier.

3.1. Analysis of Protein Amino Acid Composition

It has been already pointed out that a specific feature of a probable ID region is the amino acid compositional bias characterized by a low content of so-called order-promoting residues such as C, V, L, I, M, F, Y, and W and a high content of so-called disorder-promoting residues, including Q, S, P, E, K, G, and A ***(10,51,60)***. Therefore, the analysis of the amino acid composition biases can provide useful information related to the nature of a given protein. The fractional difference in amino acid composition between a given protein (or a given protein data set) and the set of reference globular proteins is based on the recently elaborated approach ***(10)*** and provides a perfect visualization tool for

elucidating compositional biases. Here, the fractional difference is calculated as $[f(r) - f_{\text{globular}}(r)]/f_{\text{globular}}(r)$, where $r \in$ {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}, $f(r)$ is the frequency of residue r in a given protein set and $f_{\text{globular}}(r)$ is the frequency of residue r in the reference set of globular proteins, and plotted for each amino acid. Negative bars in such a plot correspond to amino acids that are depleted in a given protein in comparison with the set of globular proteins, whereas positive bars reflect the relative increase in the particular amino acid content. Step-by-step design of the fractional difference plot is described next.

3.1.1. Retrieving Sequence Information From the Swiss-Prot Database

Start the Swiss-Prot database by typing http://www.expasy.org/sprot/ in the Internet browser. Use the following steps to download sequence information in FASTA format.

1. In the window *Search* (located at the top of the front page), choose *Swiss-Prot/TrEMBL* from the pull-down menu. Type the protein name in the neighboring window and click *Go*. Alternatively, click *Full text search* in the UniProt Knowledgebase link located in the *Access to the UniProt Knowledgebase* section of the front page. Type the protein name in the *Enter search terms* window and click *Submit*.
2. On a *search in UniProt Knowledgebase (Swiss-Prot and TrEMBL)* page choose a protein of interest from the list of hits and click the corresponding link.
3. Go to the bottom of the *UniProtKB/Swiss-Prot* entry page and click *FASTA format* link located at the bottom- right corner of the *Sequence Information* section of the page.
4. Copy content of the page, which includes a descriptive header related to the protein and a protein sequence. Keep this information as it will be used in the subsequent analysis. This can be done in Notepad or Microsoft Word. A separate document for each protein is recommended in which all the results of different analyses will be stored.

3.1.2. Applying Proteomic Tools to Obtain Amino Acid Composition

1. Direct approach (if you started with Swiss-Prot database).
   a. Go to the bottom of the *UniProtKB/Swiss-Prot* entry page and click the *ProtParam* link in *Sequence analysis tools* section.
   b. *On the* ProtParam*:* selection of endpoints on the sequence page, click *Submit* if you are going to analyze entire sequence from the previous page. Otherwise, enter the desired endpoints of the sequence in windows provided for *N*- and *C-terminal* points, then hit *Submit*.
   c. Copy a section of the *ProtParam* page describing *amino acid composition*. Keep this information as it will be used in the subsequent analysis. These are $f(r)$ values for the protein.
2. Alternative approach (if the sequence was retrieved from another source):
   a. On the Swiss-Prot home page, hit the *Proteomics tools* link located in the top-right corner.

b. Choose *primary structure analysis* among the several links at the top of the *ExPASy Proteomics tools* page.
c. Click *ProtParam* link.
d. Enter a Swiss-Prot/TrEMBL accession number in the space provided or one's own sequence in the box and click *Compute parameters*.
e. *Copy a section of the* ProtParam: user-provided sequence page describing *amino acid composition*. Keep this information, as it will be used in the subsequent analysis. These are $f(r)$ values; i.e., the frequencies of residue r in the protein.

3.1.3. Compositional Profiling

Table 2 lists averaged frequencies of different residues in a reference set of globular proteins, $f_{\text{globular}}(r)$, and those in a set of experimentally validated IDPs (458 proteins, 1096 disordered regions) from the DisProt database ***(50)***, $f_{\text{IDP}}(r)$.

1. Rearrange the data for the protein by taking into account that the order of residues you retrieved from the Swiss-Prot is alphabetical (for the three-letter code): Ala(A), Arg(R), Asn(N), Asp(D), Cys(C), Gln(Q), Glu(E), Gly(G), His(H), Ile(I), Leu(L), Lys(K), Met(M), Phe(F), Pro(P), Ser(S), Thr(T), Trp(W), Tyr(Y), and Val(V), whereas it is suggested to list residues according to their disorder propensity, from the least to the most disorder-promoting C, W, Y, I, F, V, L, H, T, N, A, G, D, M, K, R, S, Q, P, and E.
2. Use $f_{\text{globular}}(r)$ values from **Table 2** and $f(r)$ values from the **Subheading 3.1.2.** to calculate the relative frequencies of amino acid residues in the protein as $\left[f(r) - f_{\text{globular}}(r)\right]/f_{\text{globular}}(r)$, where $r \in \{\text{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}\}$. This can be done using Excel (Microsoft Corporation, Redmond, WA), SigmaPlot (SYSTAT Software, Inc., San Jose, CA), Origin (OriginLab Corporation, Northampton, MA), or any other graphical software.
3. Use $f_{\text{globular}}(r)$ and $f_{\text{IDP}}(r)$ values from **Table 2** to calculate the relative frequencies of amino acid residues in a set of IDPs as $\left[f_{\text{IDP}}(r) - f_{\text{globular}}(r)\right]/f_{\text{globular}}(r)$. This also can be done using Excel, SigmaPlot for Windows, Origin, or any other graphical software.
4. Create a vertical bar chart by plotting the calculated $\left[f(r) - f_{\text{globular}}(r)\right]/f_{\text{globular}}(r)$ and $\left[f_{\text{IDP}}(r) - f_{\text{globular}}(r)\right]/f_{\text{globular}}(r)$ values for each amino acid residue. For better visual representation, residues should be ranged as follows: C, W, Y, I, F, V, L, H, T, N, A, G, D, M, K, R, S, Q, P, and E; i.e., from the most order-promoting at the left to the most disorder-promoting at the right (*see* **Fig. 1**).
5. Compare the compositional profiling plot for the protein with that of "averaged" IDP.

Figure 1 illustrates this approach by representing the relative amino acid compositions of the N-terminal (transactivation) domain of the human progesterone receptor (residues 1-566, Swiss-Prot accession no. P06401), protein arginine N-methyltransferase 1 (Swiss-Prot accession no. Q99873), and a set of ID regions available in the DisProt database ***(50)***. By these computations,

Table 2
Averaged Frequencies of Different Residues (*r*) in a Reference Set of Globular Proteins, $f_{globular}(r)$, and Those in a Set of Experimentally Validated IDPs (458 Proteins, 1096 Disordered Regions) From the DisProt Database *(50)*, $f_{IDP}(r)$

r	$f_{globular}(r)$	Rmsd	$f_{IDP}(r)$	Rmsd	R	$f_{globular}(r)$	rmsd	$f_{IDP}(r)$	rmsd
W	1.50	0.01	0.78	0.07	T	6.02	0.03	5.33	0.17
C	2.27	0.03	1.04	0.12	R	4.63	0.03	5.64	0.26
F	3.95	0.02	2.45	0.12	G	7.69	0.03	8.43	0.32
I	5.35	0.03	3.28	0.15	Q	3.77	0.02	5.27	0.24
Y	3.70	0.02	2.13	0.14	S	6.29	0.03	8.70	0.28
V	6.88	0.03	4.75	0.17	N	4.53	0.02	4.05	0.19
L	8.34	0.04	6.18	0.20	P	4.80	0.04	6.84	0.28
H	2.33	0.02	2.04	0.11	D	5.43	0.02	6.11	0.23
M	1.92	0.01	2.28	0.12	E	6.09	0.03	8.87	0.32
A	7.98	0.04	8.30	0.30	K	6.23	0.03	7.48	0.30

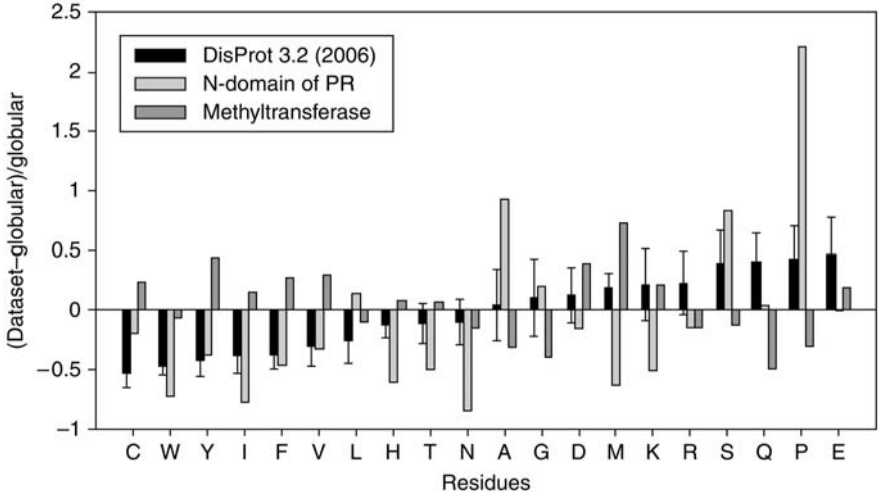


Fig. 1. Amino acid composition, relative to the set of globular proteins globular-3D, of an illustrative IDP, N-terminal (transactivation) domain of the human progesterone receptor (residues 1-566, Swiss-Prot accession no. P06401) (light gray bars); an illustrative ordered protein, protein arginine N-methyltransferase 1 (Swiss-Prot accession no. Q99873) (dark gray bars), and a set of ID regions available in the DisProt 3.2 database (454 proteins, black bars). The arrangement of the amino acids is by peak height for the DisProt 3.2 release. Confidence intervals were estimated using per-protein bootstrapping with 10,000 iterations.

arginine N-methyltransferase 1 is clearly ordered, whereas the transactivation domain is clearly disordered.

3.2. Analyzing Disorder Propensity by PONDR Tools

3.2.1. Entering Information to the PONDR Site and Retrieving Results of ID Prediction

1. Go to the official PONDR site by typing http://www.pondr.com/ in the Internet browser. You have to be registered to use the bioinformatics tools available at this site. If you are not registered as yet, click *Create a new User Account* link and follow simple instructions there. You will be provided with a username and password. If you are a registered user of PONDR, then click *Log in* to a *User Account link*, type the username and password in the corresponding windows, and hit *OK*. This will bring you to the PONDR working page.
2. While on the PONDR working page, select boxes corresponding to the desired *Predictors* (VLXT, VL3-BA, VSL1, CDF, and CH). When CH box is marked, two new boxes (*From:* and *To:*) will appear. Leave both empty. Put *Protein name* in the space provided (optional). Enter *NCBI Accession Code* or *Protein Sequence* (FASTA format or sequence only) in the corresponding boxes. Scroll down the page and check

the box *Raw Output* at the *Output Options* section. Clicking *Submit Query* will bring you to the PONDR results page.

3. It is recommended that you keep the content of the entire PONDR results page. Figures can be used as illustrations. *STATISTICS* section provides useful information on the number of residues predicted to be disordered, overall percent of disordered residues, number of disordered regions, the length of the longest disordered region, and the average prediction score. You will find herein a list of regions predicted to be disordered. Raw output values can be used to plot the results for several proteins on one graph.

3.2.2. Understanding the Results of the PONDR Analyses

1. *PONDR scores*. The PONDR results page starts with the plot providing the distribution of PONDR scores over the amino acid sequence. There will be two color lines, blue and red, corresponding to the results for the VLXT and VL3-BA predictions, respectively. Note, when using PONDR VSL1, the results will be emailed. Scores above the threshold of 0.5 correspond to the regions predicted to be disordered. Long disordered regions (with more than 30 consecutive residues predicted to be disordered) are indicated as thick black lines. **Figure 2A1**,**A2** represent illustrative PONDR score plot for the ID transactivation domain of human progesterone receptor (residues 1-566, Swiss-Prot accession no. P06401) **(Fig. 2A1)** and an ordered protein, protein arginine N-methyltransferase 1 (Swiss-Prot accession no. Q99873) **(Fig. 2A2)**. VSL1 curves are added for clarity. The vast majorities of all three curves in **Fig. 2A2** are above the threshold, reflecting the fact that the transactivation domain is highly disordered. Contrarily, the majority of curves for methyltransferase are below the threshold, confirming that this protein is highly ordered. Raw data of these analyses are at the end of the page in the *PREDICTOR VALUES* section.
2. *CDF analysis*. Second plot at the PONDR data page represents the results of CDF analysis. An illustrative CDF curve is shown in **Fig. 2B**. Remember that CDF analysis summarizes the per-residue disorder predictions by plotting PONDR scores against their cumulative frequency, which allows ordered and disordered proteins to be distinguished based on the distribution of prediction scores ***(57,58)***. In this case, order–disorder classification is based on whether a CDF curve is above or below a majority of boundary points: if curve is located below the majority of the boundary points (as shown in **Fig. 2B**), then entire protein is predicted to be mostly disordered. However, if the CDF curve is above the boundary, then the analyzed protein is mostly ordered (*see* **Fig. 2B**). Raw data to reproduce this plot (results for the protein and boundary) are in the *CDF OUTPUT* section.
3. *CH-plot analysis*. The last figure at the PONDR results page shows the CH-plot ***(25)***. As aforementioned, compact and natively unfolded proteins plotted in CH space can be separated to a significant degree by a linear boundary, with proteins located above the indicated boundary line being unfolded (red circles) and with proteins below the boundary line being compact (blue squares) **(Fig. 2C)**. The protein being tested is marked as a large green square. If this square is above the boundary, then the protein is natively unfolded. If it is below the boundary (as shown

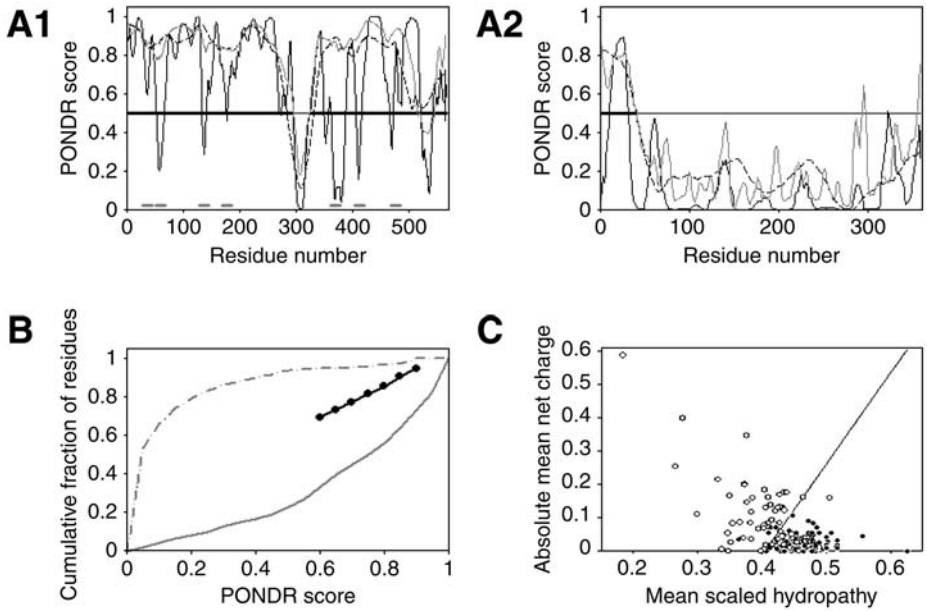


Fig. 2. Illustrative outputs of PONDR algorithms for an illustrative IDP, N-terminal (transactivation) domain of the human progesterone receptor (residues 1-566, Swiss-Prot accession no. P06401) **(Fig. 2A1**,**B**,**C)** and an illustrative ordered protein, protein arginine N-methyltransferase 1 (Swiss-Prot accession no. Q99873) **(Fig. 2A2**,**B**,**C)**. Results of the protein analysis by PONDR VLXT (black solid curves), VL-3B (black dashed), and VSL1 (gray curves) are shown in **Fig. 2A1**,**A2**. CDF curves for the transactivation domain and methyltransferase are presented in **Fig. 2B** as solid and dashed lines, respectively. **Figure 2C** illustrates corresponding CH-plots, wherein the data for the transactivation domain and methyltransferase are shown as open-crossed square and triangle, respectively. Results of α-MoRF prediction for the transactivation domain of the human progesterone receptor are shown as gray horizontal bars in **Fig. 2A1**. Seven potential α-MoRFs (fragments 27–44, 51–68, 128–145, 168–185, 360–377, 403–420, and 468–485) were identified. *Note:* on your computer screen, results of PONDR and α-MoRF predictions will be present in color: PONDR VLXT will be shown in red, VL-3B in blue, and VSL1 in magenta curves, whereas the results of a-MoRF analysis will be shown as magenta horizontal bars. In CH-plot, data for ordered and natively unfolded proteins are shown as blue squares and red circles, respectively.

in **Fig. 2C**), then the protein is compact. Raw data to build this plot (results for the protein, boundary as well as coordinates of sets of natively unfolded and ordered proteins) are in the *CHARGE-HYDROPATHY OUTPUT* section.

4. *Interpretation of PONDR data* is rather straightforward. As pointed previously, high PONDR scores (more than 0.5) for all three predictors (VLXT, VL3-BA, and VSL1) are characteristic of regions with high propensity to be disordered. Some

peculiarities of the VLXT curve might correlate with protein functionality (*see* **Subheading 3.3.1**). VL3-BA usually provides very smooth output, as it was trained on long regions of disorder and its raw predictions are averaged over an output window of length 31 to obtain the final prediction for a given position ***(65)***. VL3-BA is useful for the accurate prediction of long disordered regions. VSL1 is the most accurate predictor of intrinsic disorder at least in the PONDR series. Its training set is 1335 nonredundant protein sequences, containing 230 long disordered regions with 25,958 residues, 983 short disordered regions with 9632 residues, and 354,169 ordered residues ***(78,79)***.

5. *Interpretation of CDF and CH-plot analyses* is straightforward too. It has been pointed out that sometimes these two analyses provide seemingly contradictory data, with CDF analysis predicting a much higher frequency of disorder in sequence databases than CH-plot discrimination ***(58)***. The reasons for this discrepancy are outlined in **Subheading 4.** (*see* **Note 1**). Differences in predictions by these two classifiers were suggested to be physically interpretable in terms of the protein trinity ***(14)*** or protein-quartet models ***(15)***. Proteins predicted to be disordered by both CH-plot and CDF (i.e., polypeptide chains with high net charge and low hydrophobicity) are likely to be in the extended disorder class. Proteins predicted to be disordered by CDF, but predicted to be ordered by CH-plot, should have properties consistent with a dynamic, collapsed chain and are likely to be in the collapsed disorder class (i.e., molten globules). This supposition needs to be further tested by additional experiments. Rarely, proteins are predicted to be disordered by CH-plot, but ordered by the CDF analysis. This may represent structured proteins with an unusually high net charge; such proteins are likely to exhibit slat-sensitive structures. Finally, proteins predicted to be ordered by both algorithms are of course likely to be in the well-structured class ***(58)***. In the application to the illustrative examples of **Fig. 2**, this means that the transactivation domain of human progesterone receptor is most likely a native molten globule, whereas protein arginine N-methyltransferase 1 is likely to be ordered.

3.3. Intrinsic Disorder-Based Functional Analyses

3.3.1. Predicting the Molecular Recognition Fragments, α-MoRFs

The use of disorder predictor to find potential protein-binding sites is based on the observation that the sharp-order dips in otherwise predicted to be disordered regions, could indicate short loosely structured binding regions that undergo disorder-to-order transitions on interaction with the specific binding partner ***(81)***. Based on this presumption and the fact that such regions tend to have high α-helical propensities and high hydrophobic moments, a predictor of helix-forming α-MoRF was developed ***(82)***. Disorder-to-order transition brings a large decrease in conformational entropy, which is thought to uncouple specificity from binding strength, making highly specific interactions easily reversible. This process is illustrated in **Fig. 3**. The α-MoRF predictor can be accessed at

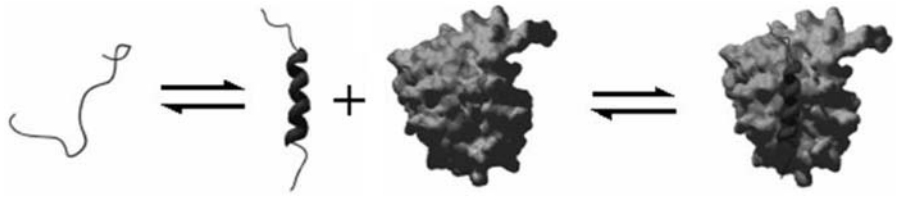


Fig. 3. Illustration of disorder-to-order transition on binding. This example shows the binding of a disordered region of Bad (ribbon) binding to Bcl-XL (globular). Modified from Oldfield et al. ***(82)***.

the official PONDR site (http://www.pondr.com/) by special request. A typical output of this predictor is shown in **Fig. 2A** as magenta horizontal bars. Notice that the predicted α-MoRFs are located within the distinctive downward spike in the PONDR VLXT curve.

3.3.2. Predicting Potential Phosphorylation Sites

It has been shown that intrinsic disorder prediction might help increase the prediction accuracy of several protein posttranslational modification sites, including protein phosphorylation ***(83)*** and methylation ***(99)***. For example, DEPP (or DisPhos) uses disorder information to improve the discrimination between phosphorylation and nonphosphorylation sites. The retrieved prediction score approximates the probability that the residue is phosphorylated. Only residues with a prediction score more than 0.5 (which) are considered to be phosphorylated. The step-by-step protocol of DEPP analysis is presented next.

1. Go to the PONDR working page and click the *DEPP Prediction* button. This will bring you to the DEPP working page. While on this page, type *Protein name* in the space provided (optional) and enter *NCBI Accession Code* or *Protein Sequence* (FASTA format or sequence only) in the corresponding boxes. Scroll down the page and check the box *Raw Output* at the *Output Options* section. By clicking *Submit Query* button you will be forwarded to the DEPP results page.
2. The top of DEPP results page represents the plot providing the distribution of DEPP scores over the amino acid sequence. You will have three types of symbols corresponding to the Thr (green triangles), Ser (blue squares), and Tyr residues (red circles) predicted to be phosphorylated. Only residues possessing DEPP scores more than 0.5 are shown. **Fig. 4** represents an illustrative DEPP plot for the transactivation domain of human progesterone receptor (residues 1-566, Swiss-Prot accession no. P06401).
3. Raw data related to this analysis are at the end of the page in the *PREDICTOR VALUES* section. The *DEPP NNP STATISTICS* section provides useful information on the number of phosphorylated serines, threonines, and tyrosines, together with

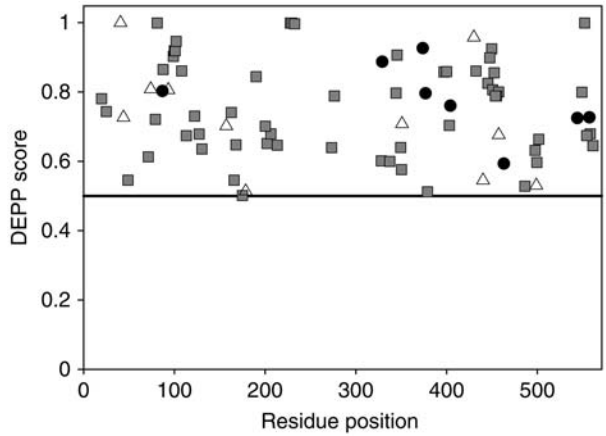


Fig. 4. Prediction of phosphorylation sites in the transactivation domain of the human progesterone receptor (residues 1-566, Swiss-Prot accession no. P06401) by DEPP. The DEPP plot provides the distribution of phosphorylation probability over the amino acid sequence. Symbols corresponding to the Thr (open triangles), Ser (gray squares), and Tyr residues (black circles) predicted to be phosphorylated. Only residues possessing DEPP scores more than 0.5 are shown. *Note:* on your computer screen, results of prediction will be present in color: Thr, Ser, and Tyr residues predicted to be phosphorylated will be shown by green triangles, blue squares, and red circles, respectively.

the total number of these residues in a given protein and the relative phosphorylation efficiency. Once again, it is recommended that one keeps the content of the entire DEPP results page for future use.

4. Notes

1. The difference in the ID prediction by CDF analysis and CH-plot likely results from the fact that the CH-plot is a linear classifier that takes into account only two parameters of the particular sequence—charge and hydrophobicity ***(25)***, whereas the CDF analysis is dependent on the output of the PONDR VL-XT predictor, a nonlinear neural network classifier, which was trained to distinguish order and disorder based on a significantly larger feature space that explicitly includes net charge and hydropathy ***(57,58)***. Therefore, CH feature space can be considered as a subset of PONDR VL-XT feature space. By definition, CH-plot analysis is predisposed to discriminate proteins with substantial amounts of extended disorder (random coils and premolten globules) from proteins with globular conformations (molten globule-like and rigid well-structured proteins). On the other hand, PONDR-based CDF analysis may discriminate all types of disordered conformations, including molten globules, premolten globules, and coils from ordered proteins ***(58)***.

Acknowledgments

The Indiana Genomics Initiative, funded in part by the Lilly Endowment, and National Institute of Health Grant no. 1 R01 LM007688-0A1 provided support for P.R., V.N.U, Z.O, and A.K.D. This work received additional support from the Programs of the Russian Academy of Sciences for the "Molecular and cellular biology" and "Fundamental science for medicine" especially for V.N.U. L.M.I. was supported by National Science Foundation (NSF) grant no. MCB0444818.

References

1. Fischer, E. (1894) Einfluss der configuration auf die wirkung der enzyme. *Ber. Dtsch. Chem. Ges.* **27,** 2985–2993.
2. Obradovic, Z., Peng, K., Vucetic, S., Radivojac, P., Brown, C. J., and Dunker, A. K. (2003) Predicting intrinsic disorder from amino acid sequence. *Proteins* **53,** 566–572.
3. Linderstrom-Lang, K. U. and Schellman, J. A. (1959) Protein structure and enzyme activity, in *The Enzymes*, (Boyer, P. D., Lardy, H., and Myrback, K., eds.), Academic Press, New York, pp. 443–510.
4. Pullen, R. A., Jenkins, J. A., Tickle, I. J., Wood, S. P., and Blundell, T. L. (1975) The relation of polypeptide hormone structure and flexibility to receptor binding: the relevance of X-ray studies on insulins, glucagon and human placental lactogen. *Mol. Cell Biochem.* **8,** 5–20.
5. Cary, P. D., Moss, T., and Bradbury, E. M. (1978) High-resolution proton-magnetic-resonance studies of chromatin core particles. *Eur. J. Biochem.* **89,** 475–482.
6. Holt, C. and Sawyer, L. (1993) Caseins as rheomorphic proteins: interpretation of primary and secondary structures of the αs1-, β-, and κ-caseins. *J. Chem. Soc. Faraday Trans.* **89,** 2683–2692.
7. Schweers, O., Schoenbrunn-Hanebeck, E., Marx, A., and Mandelkow, E. (1994) Structural studies of tau protein and alzheimer paired helical filaments show no evidence for β-structure. *J. Biol. Chem.* **269,** 24,290–24,297.
8. Weinreb, P. H., Zhen, W., Poon, A. W., Conway, K. A., and Lansbury, P. T., Jr. (1996) NACP, a protein implicated in Alzheimer's disease and learning, is natively unfolded. *Biochemistry* **35,** 13,709–13,715.
9. Wright, P. E. and Dyson, H. J. (1999) Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm. *J. Mol. Biol.* **293,** 321–331.
10. Dunker, A. K., Lawson, J. D., Brown, C. J., et al. (2001) Intrinsically disordered protein. *J. Mol. Graph. Model* **19,** 26–59.
11. Daughdrill, G. W., Pielak, G. J., Uversky, V. N., Cortese, M. S., and Dunker, A. K. (2005) Natively disordered protein, in *Protein Folding Handbook*, (Buchner, J. and Kiefhaber, T. eds.), Wiley-VCH: Verlag GmbH & Co., KGaA, Weinheim, pp. 271–353.
12. Uversky, V. N. (2003) A protein-chameleon: conformational plasticity of alpha-synuclein, a disordered protein involved in neurodegenerative disorders. *J. Biomol. Struct. Dyn.* **21,** 211–234.

13. Uversky, V. N., Oldfield, C. J., and Dunker, A. K. (2005) Showing your ID: intrinsic disorder as an ID for regcognition, regulation, and cell signaling. *J. Mol. Recognit.* **18,** 343–384.
14. Dunker, A. K. and Obradovic, Z. (2001) The protein trinity-linking function and disorder. *Nat. Biotechnol.* **19,** 805, 806.
15. Uversky, V. N. (2002) Natively unfolded proteins: a point where biology waits for physics. *Protein Sci.* **11,** 739–756.
16. Ringe, D. and Petsko, G. A. (1986) Study of protein dynamics by X-ray diffraction. *Methods Enzymol.* **131,** 389–433.
17. Dyson, H. J. and Wright, P. E. (2002) Insights into the structure and dynamics of unfolded proteins from nuclear magnetic resonance. *Adv. Protein Chem.* **62,** 311–340.
18. Bracken, C., Iakoucheva, L. M., Romero, P. R., and Dunker, A. K. (2004) Combining prediction, computation and experiment for the characterization of protein disorder. *Curr. Opin. Struct. Biol.* **14,** 570–576.
19. Dyson, H. J. and Wright, P. E. (2004) Unfolded proteins and protein folding studied by NMR. *Chem. Rev.* **104,** 3607–3622.
20. Dyson, H. J. and Wright, P. E. (2005) Elucidation of the protein folding landscape by NMR. *Methods Enzymol.* **394,** 299–321.
21. Fasman, G. D. (1996) Circular dichroism and the conformational analysis of biomolecules. Plenum Press, New York.
22. Adler, A. J., Greenfield, N. J., and Fasman, G. D. (1973) Circular dichroism and optical rotatory dispersion of proteins and polypeptides. *Methods Enzymol.* **27,** 675–735.
23. Provencher, S. W. and Glockner, J. (1981) Estimation of globular protein secondary structure from circular dichroism. *Biochemistry* **20,** 33–37.
24. Woody, R. W. (1995) Circular dichroism. *Methods Enzymol.* **246,** 34–71.
25. Uversky, V. N., Gillespie, J. R., and Fink, A. L. (2000) Why are "natively unfolded" proteins unstructured under physiologic conditions? *Proteins* **41,** 415–427.
26. Smyth, E., Syme, C. D., Blanch, E. W., Hecht, L., Vasak, M., and Barron, L. D. (2001) Solution structure of native proteins with irregular folds from Raman optical activity. *Biopolymers* **58,** 138–151.
27. Uversky, V. N. (1999) A multiparametric approach to studies of self-organization of globular proteins. *Biochemistry (Mosc)* **64,** 250–266.
28. Receveur-Brechot, V., Bourhis, J. M., Uversky, V. N., Canard, B., and Longhi, S. (2006) Assessing protein disorder and induced folding. *Proteins* **62,** 24–45.
29. Glatter, O. and Kratky, O. (1982) Small angle X-ray scattering. Academic Press, London.
30. Markus, G. (1965) Protein substrate conformation and proteolysis. *Proc. Natl. Acad. Sci. USA* **54,** 253–258.
31. Mikhalyi, E. (1978) Application of proteolytic enzymes to protein structure studies. CRC Press, Boca Raton.
32. Hubbard, S. J., Eisenmenger, F., and Thornton, J. M. (1994) Modeling studies of the change in conformation required for cleavage of limited proteolytic sites. *Protein Sci.* **3,** 757–768.

33. Fontana, A., de Laureto, P. P., de Filippis, V., Scaramella, E., and Zambonin, M. (1997) Probing the partly folded states of proteins by limited proteolysis. *Fold. Des.* **2,** R17–R26.
34. Fontana, A., de Laureto, P. P., Spolaore, B., Frare, E., Picotti, P., and Zambonin, M. (2004) Probing protein structure by limited proteolysis. *Acta Biochim. Pol.* **51,** 299–321.
35. Iakoucheva, L. M., Kimzey, A. L., Masselon, C. D., Smith, R. D., Dunker, A. K., and Ackerman, E. J. (2001) Aberrant mobility phenomena of the DNA repair protein XPA. *Protein Sci.* **10,** 1353–1362.
36. Tompa, P. (2002) Intrinsically unstructured proteins. *Trends Biochem. Sci.* **27,** 527–533.
37. Privalov, P. L. (1979) Stability of proteins: small globular proteins. *Adv. Protein Chem.* **33,** 167–241.
38. Ptitsyn, O. (1995) Molten globule and protein folding. *Adv. Protein Chem.* **47,** 83–229.
39. Ptitsyn, O. B. and Uversky, V. N. (1994) The molten globule is a third thermodynamical state of protein molecules. *FEBS Lett.* **341,** 15–18.
40. Uversky, V. N. and Ptitsyn, O. B. (1996) All-or-none solvent-induced transitions between native, molten globule and unfolded states in globular proteins. *Fold. Des.* **1,** 117–122.
41. Westhof, E., Altschuh, D., Moras, D., et al. (1984) Correlation between segmental mobility and the location of antigenic determinants in proteins. *Nature* **311,** 123–126.
42. Berzofsky, J. A. (1985) Intrinsic and extrinsic factors in protein antigenic structure. *Science* **229,** 932–940.
43. Iakoucheva, L. M., Brown, C. J., Lawson, J. D., Obradovic, Z., and Dunker, A. K. (2002) Intrinsic disorder in cell-signaling and cancer-associated proteins. *J. Mol. Biol.* **323,** 573–584.
44. Dunker, A. K., Cortese, M. S., Romero, P., Iakoucheva, L. M., and Uversky, V. N. (2005) Flexible nets. The roles of intrinsic disorder in protein interaction networks. *FEBS J.* **272,** 5129–5148.
45. Dunker, A. K., Brown, C. J., Lawson, J. D., Iakoucheva, L. M., and Obradovic, Z. (2002) Intrinsic disorder and protein function. *Biochemistry* **41,** 6573–6582.
46. Xie, H., Vucetic, S., Iakoucheva, L. M., et al. (2007) Functional anthology of intrinsic disorder. I. Biological processes and functions of proteins with long disordered regions. *J. Proteome Res*. **6,** 1882–1898.
47. Vucetic, S., Xie, H., Iakoucheva, L. M., et al. (2007) Functional anthology of intrinsic disorder. 2. Cellular components, domains, technical terms, developmental processes, and coding sequence diversities correlated with long disordered regions. *J. Proteome Res.* **6,** 1899–1916.
48. Xie, H., Vucetic, S., Iakoucheva, L. M., et al. (2007) Functional anthology of intrinsic disorder. 3. Ligands, post-translational modifications and diseases associated with intrinsically disordered proteins. *J. Proteome Res*. **6,** 1917–1932.
49. Sim, K. L., Uchida, T., and Miyano, S. (2001) ProDDO: a database of disordered proteins from the Protein Data Bank (PDB). *Bioinformatics* **17,** 379–380.

50. Vucetic, S., Obradovic, Z., Vacic, V., et al. (2005) DisProt: a database of protein disorder. *Bioinformatics* **21,** 137–140.
51. Romero, P., Obradovic, Z., Li, X., Garner, E. C., Brown, C. J., and Dunker, A. K. (2001) Sequence complexity of disordered protein. *Proteins* **42,** 38–48.
52. Wootton, J. C. (1993) Statistic of local complexity in amino acid sequences and sequence databases. *Comput. Chem.* **17,** 149–163.
53. Radivojac, P., Obradovic, Z., Smith, D. K., et al. (2004) Protein flexibility and intrinsic disorder. *Protein Sci.* **13,** 71–80.
54. Romero, P., Obradovic, Z., Kissinger, C. R., Villafranca, J. E., and Dunker, A. K. (1997) Identifying disordered regions in proteins from amino acid sequences. *IEEE Int. Conf. Neural Netw.* **1,** 90–95.
55. Lise, S. and Jones, D. T. (2005) Sequence patterns associated with disordered regions in proteins. *Proteins* **58,** 144–150.
56. Li, X., Romero, P., Rani, M., Dunker, A. K., and Obradovic, Z. (1999) Predicting protein disorder for N-, C-, and internal regions. *Genome Inform. Ser. Workshop Genome Inform.* **10,** 30–40.
57. Dunker, A. K., Obradovic, Z., Romero, P., Garner, E. C., and Brown, C. J. (2000) Intrinsic protein disorder in complete genomes. *Genome Inform. Ser. Workshop Genome Inform.* **11,** 161–171.
58. Oldfield, C. J., Cheng, Y., Cortese, M. S., Brown, C. J., Uversky, V. N., and Dunker, A. K. (2005) Comparing and combining predictors of mostly disordered proteins. *Biochemistry* **44,** 1989–2000.
59. Vucetic, S., Radivojac, P., Obradovic, Z., Brown, C. J., and Dunker, A. K. (2001) Methods for improving protein disorder prediction, in *International Joint INNS-IEEE Conference on Neural Networks*, Washington, DC, pp. 2718–2723.
60. Vucetic, S., Brown, C. J., Dunker, A. K., and Obradovic, Z. (2003) Flavors of protein disorder. *Proteins* **52,** 573–584.
61. Melamud, E. and Moult, J. (2003) Evaluation of disorder predictions in CASP5. *Proteins* **53(Suppl 6),** 561–565.
62. Jin, Y. and Dunbrack, R. L., Jr. (2005) Assessment of disorder predictions in CASP6. *Proteins* **61(Suppl 7),** 167–175.
63. Jones, D. T. and Ward, J. J. (2003) Prediction of disordered regions in proteins from position specific score matrices. *Proteins* **53,** 573–578.
64. Jones, D. T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* **292,** 195–202.
65. Peng, K., Vucetic, S., Radivojac, P., Brown, C. J., Dunker, A. K., and Obradovic, Z. (2005) Optimizing long intrinsic disorder predictors with protein evolutionary information. *J. Bioinformatics Comput. Biol.* **3,** 35–60.
66. Linding, R., Russell, R. B., Neduva, V., and Gibson, T. J. (2003) GlobPlot: exploring protein sequences for globularity and disorder. *Nucleic Acids Res.* **31,** 3701–3708.
67. Linding, R., Jensen, L. J., Diella, F., Bork, P., Gibson, T. J., and Russell, R. B. (2003) Protein disorder prediction: implications for structural proteomics. *Structure* **11,** 1453–1459.

68. Liu, J., Tan, H., and Rost, B. (2002) Loopy proteins appear conserved in evolution. *J. Mol. Biol.* **322,** 53–64.
69. Liu, J. and Rost, B. (2003) NORSp: Predictions of long regions without regular secondary structure. *Nucleic Acids Res.* **31,** 3833–3835.
70. Ward, J. J., Sodhi, J. S., McGuffin, L. J., Buxton, B. F., and Jones, D. T. (2004) Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J. Mol. Biol.* **337,** 635–645.
71. Ward, J. J., McGuffin, L. J., Bryson, K., Buxton, B. F., and Jones, D. T. (2004) The DISOPRED server for the prediction of protein disorder. *Bioinformatics* **20,** 2138–2139.
72. Dosztanyi, Z., Csizmok, V., Tompa, P., and Simon, I. (2005) The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins. *J. Mol. Biol.* **347,** 827–839.
73. Dosztanyi, Z., Csizmok, V., Tompa, P., and Simon, I. (2005) IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics* **21,** 3433–3434.
74. Prilusky, J., Felder, C. E., Zeev-Ben-Mordehai, T., et al. (2005) FoldIndex: a simple tool to predict whether a given protein sequence is intrinsically unfolded. *Bioinformatics* **21,** 3435–3438.
75. Yang, Z. R., Thomson, R., McNeil, P., and Esnouf, R. M. (2005) RONN: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics* **21,** 3369–3376.
76. Coeytaux, K. and Poupon, A. (2005) Prediction of unfolded segments in a protein sequence based on amino acid composition. *Bioinformatics* **21,** 1891–1900.
77. Cheng, J., Sweredoski, M. J., and Baldi, P. (2005) Accurate prediction of protein disordered regions by mining protein structure data. *Data Mining Knowledge Disc.* **11,** 213–222.
78. Obradovic, Z., Peng, K., Vucetic, S., Radivojac, P., and Dunker, A. K. (2005) Exploiting heterogeneous sequence properties improves prediction of protein disorder. *Proteins* **61(Suppl 7),** 176–182.
79. Peng, K., Radivojac, P., Vucetic, S., Dunker, A. K., and Obradovic, Z. (2006) Length-dependent prediction of protein intrinsic disorder. *BMC Bioinformatics* **7,** 208.
80. Vullo, A., Bortolami, O., Pollastri, G., and Tosatto, S. C. (2006) Spritz: a server for the prediction of intrinsically disordered regions in protein sequences using kernel machines. *Nucleic Acids Res.* **34,** W164–W168.
81. Garner, E., Romero, P., Dunker, A. K., Brown, C., and Obradovic, Z. (1999) Predicting binding regions within disordered proteins. *Genome Inform. Ser. Workshop Genome Inform.* **10,** 41–50.
82. Oldfield, C. J., Cheng, Y., Cortese, M. S., Romero, P., Uversky, V. N., and Dunker, A. K. (2005) Coupled folding and binding with alpha-helix-forming molecular recognition elements. *Biochemistry* **44,** 12,454–12,470.
83. Iakoucheva, L. M., Radivojac, P., Brown, C. J., et al. (2004) The importance of intrinsic disorder for protein phosphorylation. *Nucleic Acids Res.* **32,** 1037–1049.

84. Ritter, L. M., Arakawa, T., and Goldberg, A. F. (2005) Predicted and measured disorder in peripherin/rds, a retinal tetraspanin. *Protein Pept. Lett.* **12,** 677–686.
85. Kukhtina, V., Kottwitz, D., Strauss, H., et al. (2005) Intracellular domain of nicotinic acetylcholine receptor: the importance of being unfolded. *J. Neurochem*.
86. Yiu, C. P., Beavil, R. L., and Chan, H. Y. (2006) Biophysical characterisation reveals structural disorder in the nucleolar protein, Dribble. *Biochem. Biophys. Res. Commun.* **343,** 311–318.
87. Hinds, M. G., Smits, C., Fredericks-Short, R., et al. (2007) Bim, Bad and Bmf: intrinsically unstructured BH3-only proteins that undergo a localized conformational change on binding to prosurvival Bcl-2 targets. *Cell Death Differ*. **14,** 128–136.
88. Nardini, M., Svergun, D., Konarev, P. V., et al. (2006) The C-terminal domain of the transcriptional corepressor CtBP is intrinsically unstructured. *Protein Sci.* **15,** 1042–1050.
89. Roy, S., Schnell, S., and Radivojac, P. (2007) Unraveling the nature of the segmentation clock: intrinsic disorder of clock proteins and their interaction map. *Comput. Biol. Chem*. **30,** 241–248.
90. Popovic, M., Coglievina, M., Guarnaccia, C., et al. (2006) Gene synthesis, expression, purification, and characterization of human Jagged-1 intracellular region. *Protein Expr. Purif.* **47,** 398–404.
91. Cheng, Y., Le Gall, T., Oldfield, C. J., Dunker, A. K., and Uversky, V. N. (2006) Abundance of intrinsic disorder in proteins associated with cardiovascular disease. *Biochemistry* **45,** 10,448–10,460.
92. Liu, J., Perumal, N. B., Oldfield, C. J., Su, E. W., Uversky, V. N., and Dunker, A. K. (2006) Intrinsic disorder in transcription factors. *Biochemistry* **45,** 6873–6888.
93. Singh, G. P., Ganapathi, M., Sandhu, K. S., and Dash, D. (2006) Intrinsic unstructuredness and abundance of PEST motifs in eukaryotic proteomes. *Proteins* **62,** 309–315.
94. Hansen, J. C., Lu, X., Ross, E. D., and Woody, R. W. (2006) Intrinsic protein disorder, amino acid composition, and histone terminal domains. *J. Biol. Chem.* **281,** 1853–1856.
95. Haynes, C. and Iakoucheva, L. M. (2006) Serine/arginine-rich splicing factors belong to a class of intrinsically disordered proteins. *Nucleic Acids Res.* **34,** 305–312.
96. Bustos, D. M. and Iglesias, A. A. (2006) Intrinsic disorder is a key characteristic in partners that bind 14-3-3 proteins. *Proteins* **63,** 35–42.
97. Denning, D. P., Patel, S. S., Uversky, V., Fink, A. L., and Rexach, M. (2003) Disorder in the nuclear pore complex: the FG repeat regions of nucleoporins are natively unfolded. *Proc. Natl. Acad. Sci. USA* **100,** 2450–2455.
98. Boeckmann, B., Bairoch, A., Apweiler, R., et al. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.* **31,** 365–370.
99. Daily, K. M., Radivojac, P., and Dunker, A. K. (2005) Intrinsic disorder and protein modifications: building an SVM predictor for methylation, in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, *CIBCB 2005*, San Diego, California, CA, pp. 475–481.

II

Computational Methods II

6

Sybil: Methods and Software for Multiple Genome Comparison and Visualization

Jonathan Crabtree, Samuel V. Angiuoli, Jennifer R. Wortman, and Owen R. White

Summary

With the successful completion of genome sequencing projects for a variety of model organisms, the selection of candidate organisms for future sequencing efforts has been guided increasingly by a desire to enable comparative genomics. This trend has both depended on and encouraged the development of software tools that can elucidate and capitalize on the similarities and differences between genomes. "Sybil," one such tool, is a primarily web-based software package whose primary goal is to facilitate the analysis and visualization of comparative genome data, with a particular emphasis on protein and gene cluster data. Herein, a two-phase protein clustering algorithm, used to generate protein clusters suitable for analysis through Sybil and a method for creating graphical displays of protein or gene clusters that span multiple genomes are described. When combined, these two relatively simple techniques provide the user of the Sybil software (The Institute for Genomic Research [TIGR] Bioinformatics Department) with a browsable graphical display of his or her "input" genomes, showing which genes are conserved based on the parameters supplied to the protein clustering algorithm. For any given protein cluster the graphical display consists of a local alignment of the genomes in which the clustered genes are located. The genomes are arranged in a vertical stack, as in a multiple alignment, and shaded areas are used to connect genes in the same cluster, thus displaying conservation at the protein level in the context of the underlying genomic sequences. The authors have found this display—and slight variants thereof—useful for a variety of annotation and comparison tasks, ranging from identifying "missed" gene models or single-exon discrepancies between orthologous genes, to finding large or small regions of conserved gene synteny, and investigating the properties of the breakpoints between such regions.

Key Words: Bioinformatics; Bioperl; comparative genomics; ortholog; paralog; protein clustering; visualization.

1. Introduction

There are many ways to compare genomes and Sybil focuses on one of the simplest methods for evaluating potential functional differences between

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

genomes, which is to examine their relative protein-coding gene complements. Doing this requires that one make judgments about which of the genes are orthologs, under the assumption that these genes are most likely to have conserved functional roles. Numerous published algorithms deal with the problem of computing clusters of orthologous and paralogous genes ***(1–6)*** and such clusters may also be refined or defined manually, with the aid of trained curators ***(7–9)***. Although the cluster analysis and display tools in Sybil are largely agnostic with respect to the question of how the proteins are clustered, they have been used primarily with the combination of simple protein clustering techniques described in **Subheading 3.1.** This is a two-phase heuristic protein clustering method that combines an initial step in which a Jaccard similarity coefficient ***(10)*** is calculated for every pair of proteins (*see* **Subheading 3.1.2.**), with a second step that performs a bidirectional best hit analysis (*see* **Subheading 3.1.3.**) on the *clusters* generated by the first phase of the algorithm, rather than on individual proteins.

Once protein clusters representing paralogs and/or orthologs have been defined, Sybil provides a web-based interface that allows the cluster data to be explored. At the level of entire genomes the protein clusters are used to support queries about relative gene complements (e.g., clusters which contain at least one representative from genome A and at least one representative from genome B but none from genomes C or D), and to support the generation of multiple-genome comparative figures (*see* **Fig. 1**). At the level of individual genes the clusters are used for finer-grained analyses (e.g., enumerate all differences in gene structure that appear to be unique to genome B). Central to this latter, high-resolution view of the protein clusters, is a graphical display that shows each gene in a cluster in its relevant genomic context, with nearby gene clusters highlighted (*see* **Fig. 2**). Variants of this basic graphical view are utilized in a number of places in Sybil and the method used to generate this view, which leverages the Bio::Graphics package of Bioperl (http://en.wikipedia.org/wiki/Open_source, http://www.bioperl.org/wiki/History_of_BioPerl) ***(11)***, is described in **Subheading 3.2.**

Other tools display matches between sequences and/or genomes in a similar way ***(12–14)***, but the figures produced by Sybil tend to be somewhat simpler and easier to interpret owing to the use of the protein cluster as a "minimum unit" of conservation. Sybil can also make use of the protein clusters to infer the presence of regions of conserved synteny or "syntenic blocks." A number of tools have been developed in Sybil to identify and visualize such large-scale conserved regions and how they are rearranged between genomes (***21***; **Fig. 2** and ***21***; color plate no. 1). However, these are beyond the scope of this chapter.

It should be noted that although the current system relies on certain software packages, programming libraries, data exchange formats, languages, and databases, these choices are largely incidental to the protein clustering method

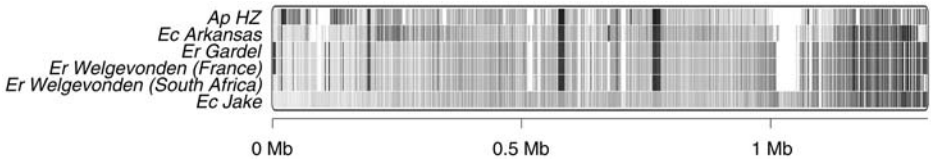


Fig. 1. A six-way genome comparison generated using one of the Sybil tools available at http://www.tigr.org/sybil/rcd. The genes in the reference sequence **(bottom row)** are color-coded according to their position in the genome. Those in all the other sequences are assigned the color of their orthologs in the reference sequence (or are left uncolored if they have none). Therefore, the figure provides a very high-level view of deletions, insertions, and rearrangements of any number of sequences compared with a fixed reference. For more examples of this type of figure *see* ***(29)*** **(Fig. 3)** and ***(30)*** **(Fig. 2)**.

described in **Subheading 3.1.** Therefore, in that section, an attempt is made to provide a largely implementation-neutral description of the technique, relegating any comments on specific implementation choices and strategies to **Subheading 4.** On the other hand, **Subheading 3.2.** describes a technique that would take significantly longer to implement without using Bioperl, and which might be of general interest in its own right. Therefore, in that section, one pays closer attention to the specific technical details that must be observed in order to interoperate with the Bio::Graphics package.

2. Materials

2.1. Protein Clustering

1. *Genome sequences:* two or more sequenced genomes, preferably in a finished or nearly finished state (*see* **Note 1**).
2. *Gene models/predictions:* a complete set of gene models for each of the sequenced genomes (*see* **Note 2**). At minimum each gene model should consist of a set of protein-coding exon locations, plus the translation start and stop positions if either the 5′- or 3′-exon contains untranslated sequence, i.e., the same information that is typically encoded in a GenBank (Protein-coding sequence) CDS feature.
3. *Polypeptides:* a polypeptide sequence for each of the protein-coding genes in **step 2**. If the polypeptide sequences are not specified explicitly then they can be computed from the gene model information supplied in this section.

2.2. Protein Cluster Visualization

1. A set of protein clusters in which no protein is a member of more than one cluster (*see* **Note 3**).
2. A database that contains (at least) the protein clusters in addition to the genome sequence data, gene models, and polypeptides for each input genome from **Subheading 2.1.1.** (*see* **Note 4**).

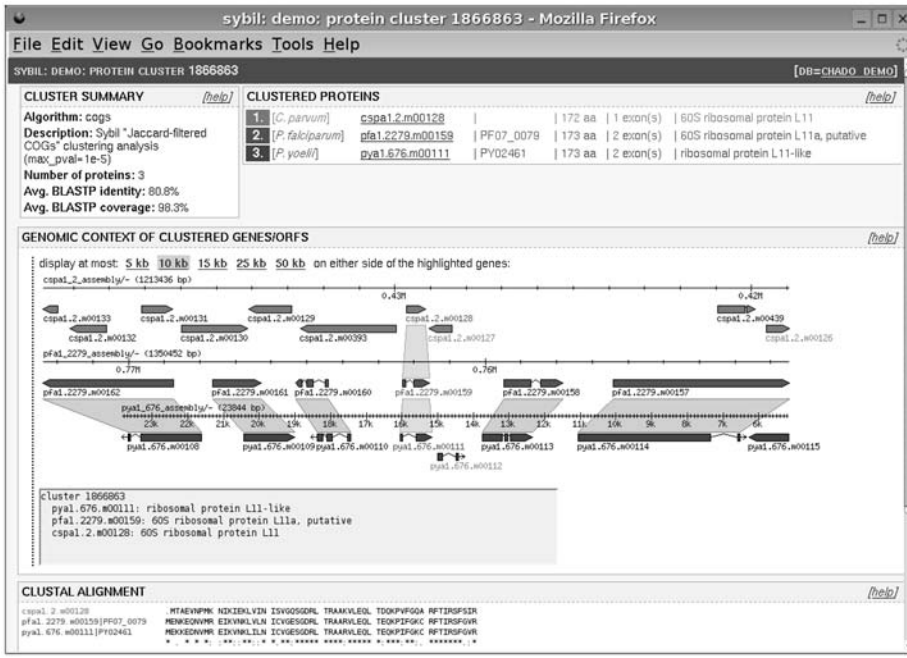


Fig. 2. Sybil protein cluster report page. Genes in the selected cluster are listed at the top of the page **(top right)** along with the cluster's average percent identity and percent coverage scores **(top left)**. The selected cluster appears in the middle of the graphical display, which shows sequences from three different genomes. Each of the three sequences is oriented and positioned so that the three clustered genes appear to be in the same orientation and are centered horizontally. As a result this is strictly a local alignment of the sequences, based solely on the three genes in the selected cluster. The main cluster display is clickable and when the mouse is placed over a gene or a gene cluster a description of the relevant gene(s) appears in the text area below the display. The lower two genomes match very closely here, although note that the bottom sequence has a gene model (pya1.676.m00112) that has no orthologs in this particular region of the middle sequence. In fact, the name of this model is drawn in a lighter shade of gray, a cue used to indicate that it does not have a predicted ortholog anywhere in either of the other two genomes.

2.3. Example Data and Display Software (Optional)

1. A complete example data set (including gene models, predicted polypeptides, all-vs-all protein-protein BLAST (Basic Local Alignment Search Tool) BLASTP results, protein clusters, and protein cluster alignments) may be downloaded from http://sybil.sourceforge.net. Also available for download is a set of Perl modules that implement the graphical protein cluster display algorithm described in **Subheading 3.2.**

3. Methods

3.1. Protein Clustering

3.1.1. "All-vs-All" BLASTP Analysis

1. xdformat is used to create a BLASTP-searchable database of the predicted polypeptide sequences from all of the input genomes: xdformat –p –I –o all-peptides all-peptides.fsa ***(15)***. It is assumed that each polypeptide has been assigned a unique identifier and can be related back to the gene of which it is a product.
2. Each of the predicted polypeptide sequences is searched against the database from **step 1** with WU-BLASTP ***(15,16)*** (*see* **Note 5**) and the results are stored for use in subsequent steps (*see* **Note 6**): blastp all-peptides pep-1.fsa –E 1e-5 –matrix BLOSUM62 –wordmask none –B 150 –V 150 –gspmax 5 –shortqueryok –novalidctxok –cpus 1 > pep-1-vs-all-blastp.raw.

3.1.2. Clustering Phase 1: Jaccard Coefficient-Based Protein Clustering

The first phase of the protein clustering algorithm is run on each input genome separately. In this phase, a subset of the all-vs-all BLASTP matches is used to compute a Jaccard similarity coefficient ***(10)*** for every pair of polypeptides from the same genome. All pairs of polypeptides whose Jaccard coefficient is more than a specified threshold are then subjected to a straightforward graph analysis to determine the resulting clusters. For each input genome:

1. Identify the subset of the BLASTP matches to be used. By default only BLASTP matches with at least 80% sequence identity and an *E*-value of at most 1×10^{-5} are used in the subsequent steps (*see* **Note 7**).
2. Use the BLASTP matches from **step 1** to determine which pairs of polypeptides are "related" to one another; by definition one considers two polypeptides related if *either one* has a BLASTP match to the other that meets the conditions described in **step 1**. Every polypeptide is also considered to be related to itself, regardless of whether a BLASTP self-match was found in **step 1**.
3. Compute and record a Jaccard similarity coefficient for each pair of predicted polypeptides. **Fig. 3** illustrates how this is done for a representative pair of polypeptides. For any two polypeptides P1 and P2 the Jaccard similarity coefficient is the ratio of the number of polypeptides (including P1 and P2 themselves) that are related to *both* P1 *and* P2 to the number of polypeptides that are related to *either* P1 *or* P2. Therefore, the Jaccard similarity coefficient for any pair of polypeptides P1 and P2 is a number between zero and one that reflects how similarly connected P1 and P2 are to the other polypeptides in the same data set (in this case, a single genome).
4. Create a graph (*see* **Fig. 4**) in which each node corresponds to one of the polypeptides from the selected input genome, and an edge is drawn between two polypeptides P1 and P2 only if the Jaccard similarity coefficient of P1 and P2 is equal to or more than a predetermined threshold (set to 0.6 by default) (*see* **Note 8**).
5. The connected components of the graph generated in **step 4**, when treated as sets of polypeptides, are referred to as "Jaccard clusters," or "JACs" for short. These

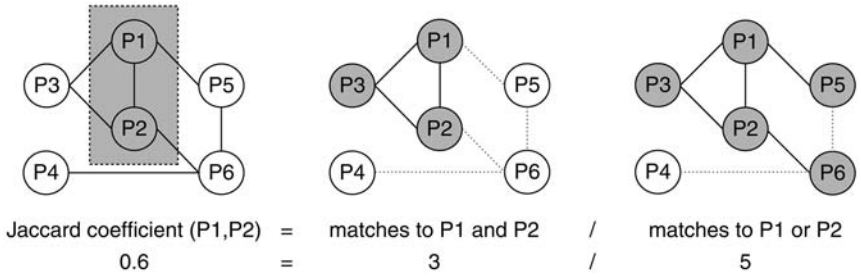


Fig. 3. Computing the Jaccard similarity coefficient for P1 and P2. In each of the three graphs the labeled circles represent proteins and the edges between the circles indicate which proteins have a BLASTP match above the preset thresholds; the edges are nondirectional and an edge is drawn if either protein matches the other at the requisite percent identity and *E*-value. In addition every protein is assumed to match itself (these edges are not shown). In this example there are six proteins with BLASTP matches as shown **(left panel)**. There are three proteins that match both P1 and P2 (highlighted in the **middle panel**) and five proteins that match either P1 or P2 (highlighted in the **right panel**). The Jaccard coefficient for P1 and P2 is therefore 3/5 or 0.6.

clusters are the output of the first phase of the clustering process (*see* **Fig. 4**, right panel).

3.1.3. Clustering Phase 2: Bidirectional Best (BLASTP) Hit Clustering

The second phase of the clustering algorithm consists of a bidirectional best (BLASTP) hit analysis (*see* **Note 9**):

1. Identify pairs of JACs (JAC1 and JAC2) that satisfy the following conditions:
   a. Each of the two clusters (JAC1 and JAC2) is from a different input genome.
   b. The highest-scoring BLASTP match (*see* **Note 10**) of at least one polypeptide in JAC1 is to a polypeptide in JAC2, and vice versa.

   An optional filtering step limits the BLASTP matches considered in condition **b** to those with an *E*-value that falls below a given threshold. In practice, this threshold is typically set to the same one that is applied in both the all-vs-all BLASTP and Jaccard clustering steps.
2. Transform the pairs of JACs found in **step 1** into a graph whose nodes are the individual JACs. An edge should be drawn between two nodes JAC1 and JAC2 only if JAC1 and JAC2 are among the pairs of JACs with bidirectional best hits from **step 1**.
3. The connected components (*see* **Note 11**) of the graph constructed in **step 2** are referred to as "Jaccard orthologous clusters," or "JOCs." Although these clusters are actually clusters of JACs, they can be easily converted to polypeptide clusters, by taking the union of the polypeptides in the JACs. These polypeptide clusters are the output of the second and final phase of the clustering process (*see* **Fig. 6**).

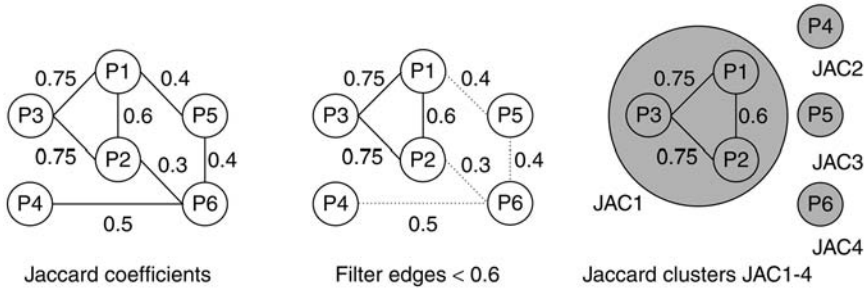


Fig. 4. Computing JACs. A graph is created in which each pair of proteins with a nonzero Jaccard coefficient is connected by an edge **(left panel)**. Edges labeled with Jaccard coefficients below the default threshold of 0.6 are removed **(middle panel)**. The connected components of the resulting graph are the JACs (JAC1–JAC4 in the **right panel**). Note that in the current implementation of the algorithm only clusters of size two or greater are reported and stored in the database (i.e., JAC1) and any polypeptide not in one of these clusters is assumed, by convention, to be a cluster of size one.

3.1.4. Generate ClustalW Alignments

1. ClustalW is run on each of the protein clusters (*see* **Note 12**) generated by the previous step to produce a set of multiple sequence alignments ***(17)***. These alignments are stored alongside the clusters and presented in the Sybil interface as a means to assess the quality of each cluster.

3.1.5. Compute Cluster Summary Scores

The all-vs-all BLASTP results are used to compute two scores for each of the JACs and JOCs. The first score is an average percent identity score and the second is an average coverage score; together these two numbers allow one to make a rapid quantitative assessment of a cluster without having to examine its full ClustalW alignment. The average percent identity score reflects how well-conserved the matching regions of the clustered polypeptides are, whereas the average percent coverage score reflects how much of each of the clustered polypeptides matches the others (i.e., how completely the BLASTP GSPs "cover" the clustered proteins). Using only these two scores one can quickly identify the most highly conserved high-confidence clusters—they are those with both a high average percent identity and a high percent coverage score. If, on the other hand, the average percent identity score is very high but the coverage score is relatively low, it may indicate a cluster of polypeptides that share a common motif (or one or more exons, in the case of alternatively spliced transcripts or misannotated genes). Finally, a cluster with a high percent coverage score but a relatively low percent identity score may be a genuine cluster of orthologous genes whose members are only distantly related.

3.1.5.1. Filter BLASTP GSPs

1. For each pair of proteins in the cluster P1 and P2 find the highest-scoring BLASTP, GSPs, or high-scoring segment pair (HSPs) that align P1 and P2.

3.1.5.2. Average Percent Identity Score

1. Calculate the (unweighted) average percent identity of all the high-scoring GSPs from **Subheading 3.1.5.1.**; this is the cluster's average percent identity score.

3.1.5.3. Average Percent Coverage Score

1. Retrieve all high-scoring BLASTP HSPs/GSPs from **Subheading 3.1.5.1.** for a *single* pair of polypeptides (P1 and P2) in the cluster.
2. Create a list of all the intervals on P1 that are aligned to P2 by an HSP/GSP.
3. Merge (take the union of) any intervals that overlap until no overlaps remain.
4. Sum the lengths of the merged intervals and divide this quantity by the length of P1. The result should be a number between zero and one.
5. Repeat **steps 2–4** for P2.
6. Repeat **steps 1–5** for all pairs of polypeptides in the cluster.
7. Compute the average of all the values computed in **step 4**, multiplying by 100 to obtain a percentage value. This is defined to be the cluster's average percent coverage score (*see* **Note 13**).

3.2. Protein Cluster Visualization

This section describes how to generate a multiple-genome graphical display like the one that appears in the Sybil protein cluster report page shown in **Fig. 2**. Each individual genomic sequence or genomic sequence fragment that appears in the display is rendered using the Perl package Bio::Graphics::Panel, which is part of the Bioperl ***(11)*** toolkit. The technique allows several Bio::Graphics::Panels to appear in the same image, with additional shaded areas used to indicate which genes belong to the same cluster. Given a cluster identifier the algorithm proceeds as follows:

1. Retrieve all proteins in the cluster (*see* **Note 4**).
2. Retrieve the gene models that correspond to the proteins in **step 1** and determine their respective genomic locations.
3. Retrieve all gene models and any other features of interest within a specified vicinity (*see* **Note 14**) of the clustered genes (*see* **Note 15**).
4. Convert all genomic sequence fragments, gene models, and other sequence features into bioPerl objects (*see* **Note 16**).
5. Create a Bio::Graphics::Panel for each of the genomic sequence fragments to appear in the figure (*see* **Note 17**). This is done in a top-to-bottom fashion so that the vertical offset of each successive panel can be set so that it does not overlap with the panels above it (*see* **Note 18**). Calling the *height()* method of a panel forces it to compute the layout of all the features contained within it, but without actually drawing any of those features.

6. Once the individual panels have been initialized it is possible to determine the overall dimensions of the combined image. This information is used to allocate a drawing area (in the form of a Perl GD::Image object) that is large enough to hold all of the panels, arranged vertically as described.
7. Generate a complete set of "matching gene pairs." Two genes are considered matching if both have a protein product in the same cluster (JAC or JOC).
8. Transform the genes in the figure into a graph, creating a node for each distinct (panel, gene) pair. An edge is drawn between (pA, geneA) and (pB, geneB) for each matching gene pair (geneA, geneB) identified in **step 7**. and each panel pA, pB in which geneA and geneB, respectively, appear. Each edge is assigned a weight using the following formula, which depends only on the *panels* in which geneA and geneB appear (*see* **Note 19**):

   edge_weight [(pA, geneA), (pB, geneB)] = [distance (pA, pB)]^2 – 1
   where distance (pA, pB) = (number of panels between pA and pB) + 1.

9. Use any minimum spanning tree (MST) algorithm ***(18)*** to select a minimal set of edges from those calculated in **step 8** (*see* **Note 20**). These edges represent the gene–gene matches that will be drawn in the figure (*see* **Note 21** and **Fig. 7**).
10. Draw the filtered set of matches into the background of the image (*see* **Note 22**), using the *boxes()* method of Bio::Graphics::Panel to determine the on-screen locations of the matching gene pairs.
11. Draw the individual Bio::Graphics::Panels on top of the previously drawn matches from **step 10** (*see* **Note 23**).
12. Generate a Portable Network Graphics (PNG) or Joint Photographic Experts Group (JPEG) (*see* **Note 24**) image suitable for display on a web page (*see* **Fig. 2**) using the standard GD::Image methods.

4. Notes

1. As the protein clustering algorithm uses a bidirectional best hit analysis to compute orthologs, it is important that the respective polypeptide sets be as complete as possible, lest one of the polypeptides not find its true "mate" owing to an incompletely sequenced or annotated genome. The algorithms can and have been used on partial polypeptide sets, but the limitation of such data sets is that they cannot reliably be used to ask questions about the *absence* of an ortholog for a particular gene or protein.
2. An automated gene prediction algorithm may be used for this purpose. It is not critical that all the gene models are completely accurate; indeed, if a sufficiently similar and well-annotated genome is included in the analysis then a subsequent comparative analysis of the gene calls can be used to identify many of the omissions and discrepancies. To this end, a comparative "structural annotation tool" that allows curators to examine several genomes at once and tag common annotation discrepancies for later correction has been developed. The annotation tool also allows one to manually add or remove proteins to or from any protein cluster, and to create or delete entire clusters.

3. For the sake of simplicity it is assumed that no protein is a member of more than one cluster *in the same protein clustering analysis*. However, the system is routinely used to compare protein clusters generated using either different algorithms or the same algorithm with different parameter settings.
4. Sybil reads annotation and comparative data from a Sybase or PostgreSQL (*see* www.postgresql.org/about/history for details) relational database using the chado ***(19)*** schema, the official database schema of the General Model Organism Database (GMOD) project ***(20)***. However, the system is based on a three-tier architecture that largely isolates the various display and query tools from the specific implementation details of the database server and schema.
5. WashU-BLASTP 2.0 (produced/licensed by the Washington University in St. Louis School of Medicine. *See* http://blast.wustle.edu/ for details) is used for the all-vs-all BLASTP search. The parameters are configurable but by default the following options are used: "–E 1e-5 –matrix BLOSUM62 –wordmask none –B 150 –V 150 –gspmax 5 –shortqueryok –novalidctxok –cpus 1."
6. The current system uses bioinformatic sequence markup language (BSML) ***(21)*** to store the intermediate BLASTP results (which are also eventually loaded into the chado comparative database). BSML is an XML-based data exchange format for sequence-related data. In subsequent steps of the analysis the BLASTP matches are read from BSML flat files using a custom Perl API.
7. It should be emphasized that no additional conditions are placed on the BLASTP matches used to create the JACs, other than the *E*-value score and percent identity thresholds. In particular, there is no requirement that the BLASTP matches must cover a minimum percentage of either sequence, which means that a relatively short match—if of sufficiently high identity and statistical significance—is often enough to group polypeptides into the same Jaccard cluster. In early comparative databases this lack of stringency was found to be more of a help than a hindrance, particularly when one or more of the input genomes has relatively low-quality automated annotation. Gene models that incorrectly lack one or more exons (and thus have artificially abridged polypeptide sequences) are nonetheless incorporated into the same cluster as the (correct) full-length versions of those genes. When an expert curator examines these clusters, possible annotation errors can be rapidly identified and tagged for correction in a future data release. However, in more recent comparative databases that contain more genomes and larger protein families, this lack of stringency, in conjunction with the subsequent single-linkage connected component analysis, has led to some pathological cases, in which a single well-conserved domain results in artificially large clusters of otherwise unrelated polypeptides. It is hoped that using a more stringent linkage criterion to compute the connected components will address this issue.
8. The default Jaccard clustering thresholds—80% identity for the BLASTP matches and 0.6 for the Jaccard coefficient threshold—were chosen by running the algorithm on a single representative comparative database using a range of different parameter values. The resulting matrix of Jaccard cluster sets was evaluated by an expert curator and default parameter values were chosen that satisfied the following conditions:

a. The results did not appear to be overly sensitive to the values chosen (i.e., small changes in the parameter values in the neighborhood of 80% and 0.6 did not produce disproportionately large changes in the composition of the resulting protein clusters).
b. The protein clusters produced were—in the judgment of the curator—a good approximation of the "true" paralogous families in each of the genomes in question.

With respect to condition **b** it is worth noting that the Jaccard clustering phase of the clustering analysis can serve multiple purposes. Its primary goal is to cluster paralogs within each genome and prevent them from confusing the subsequent bidirectional best hit analysis. However, the Jaccard clustering phase can be viewed more generally as a kind of compression algorithm that eliminates duplicate or near-duplicate polypeptides and their corresponding genes from the data set. In realistic data sets such duplicates can be produced by processes other than recent gene duplication. For example, in one recent project ***(22)*** sequencing was performed on genomic DNA sampled from two distinct haplotypes and in this case the Jaccard clustering was used to collapse the two extremely similar sets of polypeptides into one, which greatly simplified the downstream analyses. Incomplete or erroneously assembled sequence contigs in early versions of draft genomes may also contain small-scale duplications that are artifacts of the assembly process and lead to duplicate gene calls.

9. An earlier version of the clustering algorithm relied solely on the second phase of the clustering process (*see* **Fig. 5**), which is acceptable for analyzing compact genomes with relatively little recent gene duplication. But as a bidirectional best hit analysis is easily confounded by the presence of close paralogs, the initial Jaccard clustering phase was introduced and the best hit analysis was modified to run on (Jaccard) clusters instead of individual polypeptides (*see* **Fig. 6**).
10. The "highest-scoring" BLASTP match is determined by comparing BLAST *E*-values. In the case of a tie one of the matches is picked arbitrarily as the "highest-scoring." The exact method for doing this is not important, but it should be deterministic so that the algorithm generates reproducible results. In practice, it should not matter how such ties are broken, because any two polypeptides that match a third equally well are likely to be clustered together by the first phase of the algorithm.
11. A consequence of using connected components is that the clustering of genes from genomes A and B may depend on the other genomes included in the analysis. For example, if genomes A, B, and C are clustered and gene A1 is a reciprocal best hit of B1 but not C1, and B1 is a reciprocal best hit of C1 but not A1, then A1, B1, and C1 will be placed in the same cluster. If, however, genome B were not included in the analysis then A1 and C1 would not be clustered. At first glance this may seem to be an undesirable property of the algorithm. However, it is justifiable from a logical standpoint, because if it is believed that A1 and B1 are orthologs and B1 and C1 are orthologs then it follows from the definition of the term that it should also be believed that A1 and C1 are orthologs.
12. As particularly large clusters (in terms of the number of proteins) can take much longer to run through ClustalW, and may even cause the program to (eventually) fail, a parameter for this phase of the analysis allows the ClustalW computation to

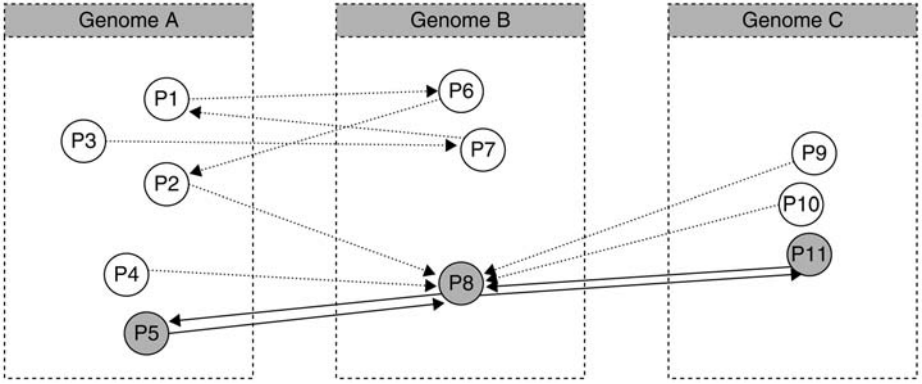


Fig. 5. Best bidirectional hit analysis on individual polypeptides. The circles represent polypeptides from three distinct genomes, and an edge from PX to PY indicates that PY is the best BLASTP match for PX (among all polypeptides in the same genome as PY). In order to simplify the example matches between Genome C and Genome A, which would usually be taken into account, are not considered. If the bidirectional best hit clustering were applied to this example—considering only matches between individual polypeptides—then the result would be a single orthologous cluster (JOC), containing P5, P8, and P11. No other polypeptides would be clustered.

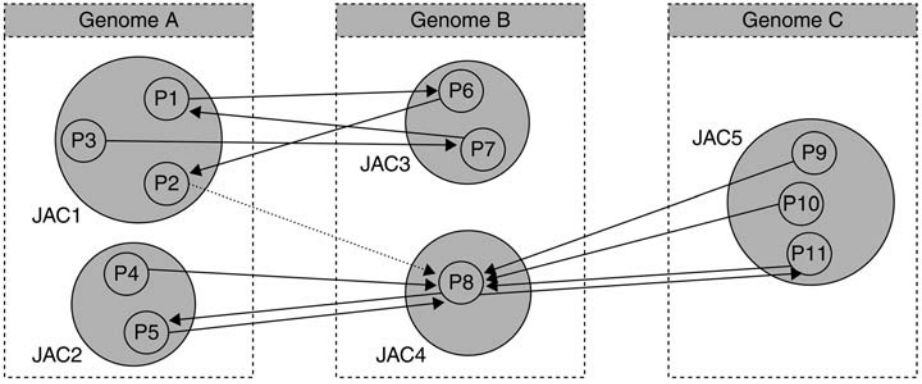


Fig. 6. Best bidirectional hit analysis on JACs. By computing bidirectional best BLASTP matches between JACs instead of individual polypeptides one is able to generate larger clusters. In this example, based on the same polypeptides and BLASTP matches as in **Fig. 4**, there would be two JOCs: one that contains all the polypeptides in JAC1 and JAC3 and one that contains all those in JAC2, JAC4, and JAC5.

be skipped for any cluster that contains more than a given number of polypeptides. This parameter is set by default to 30 proteins.

13. As a consequence of the way that the clusters are calculated and the average percent coverage and identity scores are defined, it is possible to have a cluster of two

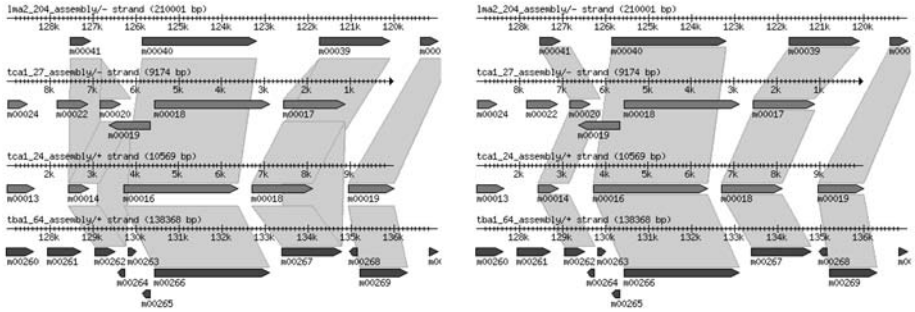


Fig. 7. Using a MST algorithm to remove redundant matches. Protein cluster image before **(left)** and after **(right)** applying the MST filter.

polypeptides P1 and P2 such that P1 has 100 amino acids and P2 has only 50, but the two match each other perfectly—over the region of the match—and therefore the cluster is assigned a percent identity score of 100% and a percent coverage score of 100%. This is not a completely satisfactory result, so in order to distinguish this case from one in which the polypeptides in a cluster match perfectly *and* are of the same length, one frequently calculates and stores a third quantity, namely the ratio of the length of the shortest polypeptide in the cluster to the length of the longest polypeptide in the cluster.

14. The default sequence "neighborhood" size is configurable and may also be changed by clicking on the links ("5 kb," "10 kb," and so on) that appear above the graphical display on the protein cluster report page (*see* **Fig. 2**). The amount of additional sequence to display is calculated by taking the extent of the longest gene in the cluster and then adding the specified neighborhood size (e.g., 5 kb) on either side of it. For the other (shorter) genes in the cluster slightly more sequence must be displayed on either side in order to make all of the sequences line up at the left and right edges of the display (assuming that the ends of the contigs are not reached before the edge of the display).
15. The features are retrieved from the chado comparative database using a standard Structured Query Language (SQL) *see* http://en.wikipedia.org/wiki/SQL range query on the chado featureloc.fmin and featureloc.fmax columns. This has produced acceptable performance, but in future one may adopt a binning scheme for faster retrieval of sequence features within a given range, as is done in GBrowse *see* http://www.bioperl.org/wiki/Lincoln_Stein and http://www.bioperl.org/wiki/Gbrowse ***(23)*** and the University of California, Santa Cruz (UCSC) Genome Browser (UCSC Genome Bioinformatics Group) ***(24)***.
16. The Bioperl features (instances of Bio::SeqFeatureI) that are created are "skeleton" features that contain the coordinates and unique identifiers of the features read from the database. A mapping is stored that allows each Bioperl feature to be mapped back to the data that were read from the database.

17. Some clusters may contain adjacent or nearby genes from the same genome, because of tandem gene duplication. If the repeated genes are within the specified "neighborhood" distance then the system will automatically include only one copy of the relevant genomic subsequence in the comparative sequence display; the alternative, which is to include multiple copies of the same sequence that are offset from one another, can be quite confusing visually.
18. The order in which the genomes and/or sequences appear in the protein cluster display may be set to a fixed default in Sybil, or one may click on a gene in the display to force that genome and sequence to appear at the top of the image. This feature was used in the TriTryp comparative annotation project ***(25)*** to designate one of the genomes as a reference against which the others were (manually) compared. Curators were able to traverse three genomes simultaneously by navigating along the fixed reference sequence using a modified version of the protein cluster display shown in **Fig. 2**.
19. The purpose of this weight function is simple; by setting the edge weights in this fashion one ensures that whenever geneA, geneB, and geneC are arranged from top to bottom in the cluster display, the algorithm will always prefer shorter matches (edges) to longer ones; geneA will be connected to geneB and geneB will be connected to geneC, instead of drawing one long match between geneA and geneC, followed by another between geneA and geneB, or geneB and geneC.
20. The authors use Kruskal's algorithm ***(26)***, as implemented by the Perl module Graph::Kruskal.
21. This approach does not always produce an ideal figure layout, but in the authors' experience it does well in simple cases, and in complex cases it will at least remove the redundant matches.
22. The gene–gene matches are drawn differently depending on whether the genes in question appear in the same orientation. This provides an easy-to-see visual cue for genes that are inverted in one genome relative to the others.
23. In order for this to work a small patch must be made to Bio::Graphics::Panel, in which calls to GD::Image::colorAllocate() are replaced with identical calls to GD::Image::colorResolve(). This change allows all the panels in the image to share the same GD color palette.
24. Sybil also supports Scalable Vector Graphics (SVG) http://www.w3.org/graphics/SVG ***(27)*** output. SVG format images can be converted to PDF with the Apache Batik package *see* http://www.apache.org/ and http://xmlgraphics.apache.org/batik/contributors.html ***(28)***, which provides an easy way to generate high-resolution images suitable for presentation or publication.

Acknowledgments

The authors would like to thank all the Institute for Genomic Research (TIGR) faculty and staff who made suggestions and contributed feedback at every stage of the algorithm and software development process. This work is funded by the National Institute of Allergy and Infectious Diseases (NIH-NIAID-DMID-04-34).

References

1. Remm, M., Storm, C. E., and Sonnhammer, E. L. (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.* **314(5),** 1041–1052.
2. Enright, A. J., Van Dongen, S., and Ouzounis, C. A. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30(7),** 1575–1584.
3. O'Brien, K. P., Remm, M., and Sonnhammer, E. L. (2005) Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Res.* **33,** D476–D480.
4. Fujibuchi, W., Ogata, H., Matsuda, H., and Kanehisa, M. (2000) Automatic detection of conserved gene clusters in multiple genomes by graph comparison and P-quasi grouping. *Nucleic Acids Res.* **28(20),** 4029–4036.
5. Ogata, H., Fujibuchi, W., Goto, S., and Kanehisa, M. (2000) A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Res.* **28(20),** 4021–4028.
6. Li, L., Stoeckert, C. J., Jr., and Roos, D. S. (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.* **13(9),** 2178–2179.
7. Tatusov, R. L., Koonin, E. V., and Lipman, D. J. (1997) A genomic perspective on protein families. *Science* **278,** 631–637.
8. Tatusov, R. L., Galperin, M. Y., Natale, D. A., and Koonin, E. V. (2000) The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.* **28,** 33–36.
9. Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., et al. (2001) The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Res.* **29,** 22–28.
10. Jaccard, P. (1908) Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.* **44,** 223–270.
11. Stajich, J. E., Block, D., Boulez, K., et al. (2002) The Bioperl toolkit: perl modules for the life sciences. *Genome Res.* **12(10),** 1611–1618.
12. Pan, X., Stein, L., and Brendel, V. (2005) SynBrowse: a synteny browser for comparative sequence analysis. *Bioinformatics* **21(17),** 3461–3468.
13. Carver, T. J., Rutherford, K. M., Berriman, M., Rajandream, M. A., Barrell, B. G., and Parkhill, J. (2005) ACT: the Artemis Comparison Tool. *Bioinformatics* **21(16),** 3422–3423.
14. Lewis, S. E., Searle, S. M. J., Harris, N., et al. (2002) Apollo: a sequence annotation editor. *Genome Biol.* **3(12),** RESEARCH0082.
15. Gish, W. (1996–2005) http://blast.wustl.edu.
16. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool. *J. Mol. Biol.* **215(3),** 403–410.
17. Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22,** 4673–4680.
18. Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001) Minimum Spanning Trees, in *Introduction to Algorithms*, *2nd ed.*, MIT Press and McGraw-Hill, pp. 561–579.

19. Chado—The GMOD Database Schema. http://www.gmod.org/schema.
20. GMOD—Generic Software Components for Model Organism Databases. http://www.gmod.org.
21. BSML: Bioinformatic Sequence Markup Language. http://www.bsml.org.
22. El-Sayed, N. M., Myler, P. J., Bartholomeu, D. C., et al. (2005) The genome sequence of *Trypanosoma cruzi*, etiologic agent of Chagas disease. *Science* **309(5733),** 409–415.
23. Stein, L. D., Mungall, C., Shu, S., et al. (2002) The generic genome browser: a building block for a model organism system database. *Genome Res.* **12(10),** 1599–1610.
24. Kent, W. J., Sugnet, C. W., Furey, T. S., et al. (2002) The Human Genome Browser at UCSC. *Genome Res.* **12(6),** 996–1006.
25. El-Sayed, N. M., Myler, P. J., Blandin, G., et al. (2005) Comparative Genomics of Trypanosomatid Parasitic Protozoa. *Science* **309(5733),** 404–409.
26. Kruskal, J. B. (1956) On the shortest spanning subtree and the traveling salesman problem. *Proc. AMS* **7,** 48–50.
27. Scalable Vector Graphics (SVG). http://www.w3.org/Graphics/SVG/.
28. Batik SVG Toolkit. http://xmlgraphics.apache.org/batik/.
29. Dunning Hotopp, J. C., Lin, M., Madupu, R., et al. (2006) Comparative Genomics of Emerging Human Ehrlichiosis Agents. *PLoS Genet.* **2(2),** E21.
30. Tettelin, H., Masignani, V., Cieslewicz, M. J., et al. (2005) Genome analysis of multiple pathogenic isolates of *Streptococcus agalactiae*: implications for the microbial "pan-genome". *Proc. Natl. Acad. Sci.* **102(39),** 13,950–13,955.

7

Estimating Protein Function Using Protein–Protein Relationships

Shailesh V. Date

Summary

Many newly identified gene products from completely sequenced genomes are difficult to characterize in the absence of sequence homology to known proteins. In such a scenario, the context of the proteins' functional associations can be used for annotation; overrepresented functional linkages with a certain class of proteins or members of a pathway allow putative function assignments based on the "guilt-by-association" principle. Two computational functional genomics methods, phylogenetic profiling and identification of Rosetta stone linkages, are described in this chapter, which allow assessment of functional linkages between proteins, consequently facilitating annotation. Phylogenetic profiling involves measuring similarity between profiles that describe the presence or absence of a protein in a set of reference genomes, whereas Rosetta stone fusion sequences help link two or more independently transcribed and translated proteins. Both methods can be applied to investigate functional associations between individual proteins, and can also be extended to reconstruct the genome-wide network of functional linkages by querying the entire protein complement of an organism.

Key Words: Interactome; protein-protein interactions; functional linkages; phylogenetic profiles; matual information; Rosetta stone fusion sequences.

1. Introduction

The number of organisms with fully sequenced genomes is growing at a rapid pace. However, many sequenced genomes are not fully annoted; analysis of this sequence data reveals that many genes and their products lack confident functional assignments, primarily because of absence of any similarity with sequences of known genes. Empirical observations suggest that the number of such uncharacterized genes is close to 30% for almost any sequenced genome, and can be as high as 60% for some, an example being the genome of the human malarial parasite *Plasmodium falciparum* ***(1)***. Absence of information about such a significant number of genes or their

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

products prevents the understanding of the biology of the organisms in detail, a fact that becomes even more important when dealing with genomes of pathogenic organisms. In this regard, use of several recently introduced computational functional genomics methods is proving beneficial, especially in assigning function to genes that are difficult to characterize using homology-based methods alone. Herein, the implementation of two such *in silico* methods—phylogenetic profiling ***(2,3)*** and identification of Rosetta stone sequences ***(4)*** is discussed, which can be used to assign function to gene products based on their linkages with proteins of known function.

Phylogenetic profiling involves generating presence/absence profiles of proteins with reference to a set of fully sequenced genomes. Matching profiles are indicative of functional protein–protein interactions between the corresponding entities, with functional interactions being defined as associations that can range from direct physical contact to shared membership in the same pathway or cellular system ***(2)***. Identification of Rosetta stone links is another means of establishing functional associations between protein entities. The method was developed based on the observation that independently transcribed and translated proteins sometimes appear together as a fused protein, either in the same organism, or in the genome of some other organism ***(4)***. The presence of fusion proteins is likely to indicate a strong functional linkage between independent candidates, suggesting that the pathways they are a part of are proximate enough for the occurrence of a dual function protein.

Functional associations suggested by the methods describe protein–protein relationships, which can be used to assign putative function to uncharacterized proteins. If a function, or more commonly, members, of a particular pathway appear to be overrepresented in a set of linked proteins, it is highly likely that the query protein either performs a similar function, or is directly or indirectly linked to the particular pathway. Functional linkages obtained using phylogenetic profile data have helped identify new pathways ***(5)*** and understand patterns of evolution and conservation ***(6,7)***. Similarly, besides elucidating functional relationships, Rosetta stone linkage data has been used to reconstruct metabolic pathways in *Escherichia coli* ***(4)***, and has been combined with other experimental and computational functional genomics data sets to generate genome-wide interaction maps of high confidence in other organisms ***(8,9)***.

Protocols for constructing phylogenetic profiles and identifying Rosetta stone links are described below (*see* Methods). It is important to note that their implementation requires the ability to write and execute computer programs, including some that involve the creation of complex logical structures. Proficiency in computer programming is therefore assumed, as is the knowledge of basic local alignment search

tool (BLAST) ***(10)*** sequence comparison approach. Users who find computer programming daunting should collaborate with an experienced computer programmer.

2. Materials

2.1. Hardware

A personal computer or a computer cluster with a modern processor is required. The use of a computer cluster is always advised over a stand-alone machine for reducing analysis and compute time.

2.2. Software

1. *Operating system (OS):* the use of UNIX or a UNIX-based OS such as Linux is advocated over other common OS. If other systems are preferred, users should ensure that the OS supports the ability to write, compile, and execute custom-generated code (*see* **Note 1**).
2. *The BLAST package:* BLAST package ***(10)*** is required for the purposes of sequence comparison. The BLAST package also includes precompiled binaries ("ready-to-run" programs) of various tools and utilities besides BLAST, and is available for download from the National Center for Biotechnology Information (NCBI) website for most commonly used OS (http://www.ncbi.nlm.nih.gov/BLAST/download.shtml). The NCBI version of BLAST differs from the WU Washington University-BLAST package ***(11)*** available from the Washington University in St. Louis (http://blast.wustl.edu), both in implementation and results. Therefore, programs that parse BLAST output should be modified based on the version used. This protocol assumes the use of the NCBI BLAST package.
3. *A programmatic wrapper:* a wrapper program that is able to run BLAST searches sequentially for each protein in the query set is required (*see* **Note 2**). The wrapper program can be extended to include other steps in the protocol as well, such as parsing of BLAST results, and execute house-keeping tasks such as moving and storing various data in proper directories and compressing output files. This program has to be written by the user.
4. *BLAST results parser program:* a program that is able to extract relevant information from raw BLAST output (such as expectation values and start and stop coordinates of the matching sequence span) is also required. Information relevant to each method is described in detail in Methods. Raw BLAST output could be subsequently discarded, or saved if disk space is inexpensive. The user is free to write a parser program or use parsing programs available over the Internet (*see* **Note 2**).
5. *Other required programs:* computer programs are also needed for the following tasks:
   a. Generating phylogenetic profiles using BLAST results.
   b. Comparing phylogenetic profiles and measuring profile similarity.
   c. Finding fusion proteins using BLAST results.

Details of these tasks are described in Methods. These programs have to be coded by the user.

3. Methods

Both the computational methods described in this chapter have common initial requirements (described in **Subheading 3.1.**), such as the need for a database of reference genomes, and parsed results of BLAST searches against the database. Steps specific to the individual methods are described in **Subheading 3.2.**

3.1. Common Initial Steps for the Phylogenetic Profiling Method and the Rosetta Stone Method

3.1.1. Creating a Database of Reference Genomes

Both methods require a database of genomes against which the query sequences are compared for similarity. It is important to note that this database should contain genomes that are fully sequenced, as opposed to, say, creating a database similar to the BLAST nonredundant database, which is essentially a repository of all known protein sequences. Use of a database that contains proteins without regarding genome sequence status will generate incorrect profiles and create complications when applying statistical tests of confidence to Rosetta stone linkages.

Complete protein complements of fully sequenced genomes can be downloaded from NCBI (ftp://ftp.ncbi.nih.gov) or from websites of individual genome sequencing centers (*see* **Note 3**). Users should ensure that all amino acid sequences included in the database possess unique identifiers (*see* **Note 4**). The more genomes included in the sequence file, the better the methods will perform. Amino acid sequences of all proteins from the downloaded genomes are concatenated into a single file for ease of use and more accurate calculation of BLAST expectation (E) values. This file can also include proteins encoded by bacterial plasmids, if so desired.

For illustration purposes, this database will be referred to as "*myDatabaseFile*," populated with the following sequences:

```
>ssolfataricus|gi|15896972
MIVPVKNEERVLPRLLDRLVNLEYDKSKYEIIVVEDGSTDRTFQICKEY
EIKYN NLIRCYSLPR
>ecoli_K12|gi|16127996
MRVLKFGGTSVANAERFLRVADILESNARQGQVATVLSAPAKITNHLVA
MIEKTISGQDALPNI
>hsapiens|gi|20093441
```

```
MSLIEIDGSYGEGGGQILRTAVGMSALTGEPVRIYNIRANRPRPGLSHQH
LHAVKAVAEICDAE>hsapiens|gi|20093442
MGVIEDMMKVGMRSAKAGLEATEELIKLFREDGRLVGSILKEMEPEEI
TELLEGASSQLIRMIR>hsapiens|gi|20093443
MSGNFRKMPEVPDPEELIDVAFRRAERAAEGTRKSFYGTRTPPEVRAR
SIEIARVNTACQLVQ>>celegans|gi|1453778
MEYIYAALLLHAAGQEINEDNLRKVLEAAGVDVDDARLKATVAALEEV
DIDEAIEEAAVPAAAP>celegans|gi|1453779
MVPWVEKYRPRSLKELVNQDEAKKELAAWANEWARGSIPEPRAVLLHG
PPGTGKTSAAYALAHD
>scerevisiae|gi|6799765
MAEHELRVLEIPWVEKYRPKRLDDIVDQEHVVERLKAYVNRGDMPNLL
FAGPPGTGKTTAALCL
```

The contents of this file are in the "FASTA" format, wherein lines starting with the ">" sign are treated as comments (for more details, *see* http://www.ebi.ac.uk/ help/formats_frame.html) and lines that do not start with the ">" sign are treated as sequence. For the proteins described in the mock database above, the comment line contains an abbreviation of the organism name (the first letter of the genus name concatenated with the entire species name), followed by the words "gi," which alert the user to the fact that the following identifier is a Genbank identifier. The identifier associated with each sequence is unique, and can be used to retrieve records from the NCBI website. This comment structure is used here just for illustration purposes, and other more suitable formats can be envisioned depending on need.

Advanced users familiar with this step and the subsequent database formatting step for BLAST can choose to construct more advanced databases, such as those with indices. If an advanced database is to be created, users are advised to carefully follow instructions with respect to identifiers associated with the individual sequences.

3.1.2. Formatting the Database for Use by BLAST

The database of reference genomes requires formatting before it can be used for sequence comparison by BLAST. A special program called "*formatdb*," included with the BLAST tools package, is needed for this task. A number of options can be set for *formatdb*, depending on the type of input and output desired. However, for this protocol, as the input is a file containing amino acid sequences and no additional information is to be generated, no options need to be specified for *formatdb* (i.e., *formatdb* is run with default options):

```
% /path/to/blast/package/formatdb –i
        myDatabaseFile
```

This step creates additional files with the same name, but with different extensions such as "*.pin*" and "*.psq*." All possible *formatdb* parameters can be viewed with the '-- *help*' flag. After this step, the original file *myDatabaseFile* is no longer needed by BLAST for sequence comparison; however, this file should be maintained if disk space availability is not an issue. Once the database file is formatted, it is ready for use by BLAST.

3.1.3. Running BLAST for All Proteins in the Query Set

Individual amino acid sequences in the query set are compared with the database of reference genomes in a sequential manner, using the wrapper program described in **Subheading 2.2.** The following is a simple example of the BLAST command executed from within the wrapper script (with the wrapper script written in the PERL language; for PERL, see http://www. perlfoundation.org/):

```
% perl /path/to/wrapper/myBlastWrapper.pl –i
myInputFile –d myDatabase File –b /path/to/BLAST/exe-
cutable –p myBlastParser.pl
```

Details of the options used for the wrapper program are as follows:

–i	Name of the input file containing query protein sequences
–d	Name of the reference genome database to use
–b	The directory where the BLAST programs are located
–p	The BLAST results parser program to use

Users are free to select and set more options based on the tasks ascribed to the wrapper program. Information about various BLAST options can be obtained using the '-- *help*' flag.

The primary job of the wrapper program is to execute BLAST for each input sequence, and store the output such that it can be retrieved with a unique identifier. The output can either be stored separately as results of each comparison, or jointly, based on the configuration of the wrapper program. In addition, as mentioned previously, other steps of the protocol such as parsing BLAST results can also be executed from within the wrapper, greatly reducing the complexity and size of the output. In the author's experience, computational efficiency is high if the wrapper program couples the BLAST step with the parser step, and stores the parsed output for all input sequences in a single text file. The following pseudocode describes this approach.

```
open myInputFile;
foreach querySequence j {
  system "blastall –p blastp –i querySequence –d
myDatabaseFile –o myBLASTOutputForProtein_j";
```

```
system "myBlastParser.pl –i
myBLASTOuputForProtein_ j >>
myParsedBLASTOuputForProtein_ j";
compress myBLASTOutputForProtein_ j;
move myBLASTOutputForProtein_j.compressed to dir
storeRawBLASTData/;
}
close myInputFile;
```

3.1.4. Parsing BLAST Results

Parsing of BLAST results is required so that only the information necessary for generating phylogenetic profiles and identifying Rosetta stone sequences is retained from sequence matches against the database. This greatly reduces the size of the input required for subsequent steps. For every match of the query sequence against the database, at least five important details need to be captured and retained from the raw output:

1. The unique identifier of the subject sequence.
2. The genome to which the subject sequence belongs.
3. The BLAST expectation value of the high-scoring pair (HSP).
4. The start and stop position of the HSP on the query sequence.
5. The start and stop position of the HSP on the subject sequence.

Besides these attributes, other bits of information such as raw scores, or the percentage of sequence identity, can also be captured (*see also* **Note 2**). As the user becomes more familiar with the methods, other pieces of information can be utilized as filters, or even as substitutes for the primary attributes, when deciding the quality of a match or a hit against the reference database.

One possible form of output from a parser program is described next:

```
>query >subject raw_score: value | E-value: value |
query_start: value | query_end: value |
subject_start: value | subject_end: value |
match_length: value | identity_percentage: value |
similarity_percentage: value | query_length: value |
subject_length: value
>hsapiens|gi|20093443 >hsapiens|gi|20093443
raw_score: 300 | E-value:
1e-155 | query_start: 1 | query_end: 140 | sub-
ject_start: 1 | subject_end: 140 | match_length: 140
| identity_percentage: 100 | similarity_percentage:
100 | query_length: 140 | subject_length: 140
```

```
>hsapiens|gi|20093443 >hsapiens|gi|14556780
raw_score: 220 | E-value: 1e-138 | query_start: 1 |
query_end: 105 | subject_start: 15 | subject_end:
155 | match_length: 105 | identity_percentage: 78 |
similarity_percentage: 91 | query_length: 140 | sub-
ject_length: 244
>hsapiens|gi|20093443 >celegans|gi|85444128
raw_score: 132 | E-value: 1e-66 | query_start: 22 |
query_end: 80 | subject_start: 107 | subject_end:
165 | match_length: 58 | identity_percentage: 70 |
similarity_percentage: 88 | query_length: 140 | sub-
ject_length: 111
```

In this illustration, each line represents a BLAST hit to the query in the database of reference genomes. The output is divided into three columns: the first column is the identifier for the query, the second is the identifier for the hit (the subject) in the database, and the third describes details of the match, such as E-values and start–stop coordinates. Herein, besides the required attributes, raw scores, sequence identities and similarities, and subject sequence length are also captured. Users are free to experiment with the various stand-alone parser programs available for free through the Internet, or write their own (*see* **Note 2**). One advantage of writing a custom parser program is that it proves helpful in getting acquainted with the raw BLAST results.

3.2. Steps Specific to Individual Methods

3.2.1. The Phylogenetic Profiling Method

Phylogenetic profiles for each of the query input sequences can be created using the parsed BLAST results. In this protocol, transformed BLAST E-values will be used to construct the profile vector, rather than representing presence or absence of the query in a genome using simple binary values of 0 and 1. This use of BLAST E-values in generating profiles results in profile vectors with a higher resolution, wherein the similarity or the distance between the vectors can be measured more accurately.

3.2.1.1. Generating Phylogenetic Profiles From BLAST Data

Generating profiles involves checking BLAST results for information about best matches to the query sequence, from each genome included in the database. The E-value of this best match is retained, transformed, and used in profile construction. One method of transforming E-values, as described by Pellegrini and coworkers ***(2)***, uses the following formulation:

For each protein i and its highest scoring match in a genome j, E_{ij} represents the BLAST expectation value of the match and p_{ij} represents the transformation of E_{ij}, such that

$$p_{ij} = -\frac{1}{\log E_{ij}}$$

Introduction of logarithm-induced artifacts during this transformation is avoided by truncating values of $p_{ij} > 1$ to 1. When encoded in a computer program, the result of this procedure is vector of N p_{ij} values, where N is the number of completely sequenced genomes included in the reference database. Users are free to experiment with other ways of transforming BLAST E-values. The following is a real-life example of the phylogenetic profile vector for the *P. falciparum* protein PFA0110w, created by comparing the query sequence against a database of 163 completely sequenced genomes.

```
>pfalciparum|Pfa3D7|pfal_chr1|PFA0110w|Annotation|Sanger
1.000 1.000 1.000 1.000 1.000 0.072 0.076 0.068 1.000
1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 0.067
1.000 1.000 0.079 0.070 0.082 1.000 1.000 0.072 0.070
0.070 0.070 0.079 0.086 1.000 1.000 1.000 0.072 0.072
1.000 1.000 1.000 0.084 0.080 0.076 0.069 1.000 1.000
1.000 1.000 0.072 0.061 0.082 0.072 0.072 0.072 0.072
0.059 1.000 0.061 0.084 0.071 0.080 0.080 0.080 0.080
0.072 0.080 0.072 1.000 1.000 0.080 0.070 1.000 0.084
0.084 0.087 0.062 1.000 0.084 0.086 1.000 1.000 0.058
0.062 1.000 1.000 1.000 0.072 0.068 0.069 1.000 1.000
0.054 1.000 1.000 0.076 0.062 1.000 1.000 1.000 1.000
0.086 0.082 1.000 0.079 0.065 1.000 1.000 0.087 1.000
0.076 0.058 1.000 1.000 0.068 0.068 0.068 1.000 1.000
0.068 0.080 0.080 1.000 1.000 0.072 1.000 1.000 1.000
1.000 1.000 1.000 0.079 0.079 0.079 1.000 1.000 1.000
1.000 1.000 0.067 0.079 1.000 1.000 0.076 0.070 1.000
1.000 1.000 0.079 0.061 0.067 1.000 1.000 1.000 1.000
1.000  1.000  0.024  0.072  0.061  1.000  0.069  0.062
0.000*0.072 0.076
                      Scale
  1.000          →    0.000
  (complete absence)        (confident presence)
```

In this example, underlined scores represent archaeal genomes, whereas scores in italics represent eukaryotic genomes, and "*" indicates the transformed

BLAST score from the match against the *P. falciparum* genome. Sequence matches with BLAST E-values greater than 10^{-5} are typically discarded (*see* **Note 5**). Besides E-values, other attributes of the HSP can also be used to decide the quality of the match. For instance, a user might reject a match wherein the length of the HSP (or all HSPs combined) is not greater than 50% of the query length, or a match might be rejected based on a cutoff derived from the number of shared identical amino acids, thereby assuming absence of the query protein in the particular genome. These choices are reflected in the profile vector, and will ultimately affect the quality of the final results.

Using this method, phylogenetic profiles are constructed for each amino acid sequence included in the input file. The query set can be extended to include all known proteins from the given genome, whereby profiles can be generated on a genome-wide scale.

3.2.1.2. Measuring Profile Similarity for Function Inference

Similarity between phylogenetic profiles is indicative of functional linkage between the corresponding proteins and can be measured in a number of different ways. Besides commonly used metrics such as Euclidean distance or Pearson correlation, advanced measures such as mutual information, Hamming distance, Jaccard coefficient, or the chance co-occurrence probability distribution can also be used ***(12)***. Mutual information ***(13–15)*** is the metric of choice for this protocol, as it has the ability to capture inverse and nonlinear relationships in the data, in addition to detecting direct and linear relationships. However, users are free to use other metrics, if they seem to perform better.

Mutual information is an information theoretic measure, which is the greatest when there is complete covariation between two sets of observations, and tends to zero as the sets diverge. For two vectors of proteins X and Y, mutual information (MI) can be calculated as follows:

$$\mathrm{MI}(X,Y) = H(X) + H(Y) - H(X,Y)$$

In this equation, $H(X) = -\sum p(x) \ln p(x)$ represents the marginal entropy of the probability distribution $p(x)$ of gene X in each genome included in the database, summed over intervals in the probability distribution, whereas $H(X,Y) = -\sum\sum p(x,y) \ln p(x,y)$ represents the intrinsic entropy of the joint probability distribution of genes X and Y. Date and Marcotte ***(5)*** have described in detail the application of mutual information for measuring profile similarity. Users are directed to this paper for more information about the implementation of the method.

Mutual information can be measured in a pairwise manner for proteins in the query set. Naturally, all mutual information values are not biologically meaningful,

and a threshold needs to be adopted for discarding false-positives and functional linkages that conflict with known biological facts. Unfortunately, it is difficult to devise a cutoff that is useful for confidently describing both profile similarity as well as biological validity of the linkages. One way of obtaining a primary cutoff value relies on the use of shuffled profiles; mutual information scores between normal profiles that match or fall below the highest score observed when comparing shuffled profiles can be discarded. Certainly, other techniques can also be imagined, such as using a set of known linkages to derive a true-positive to false-positive ratio, which can then be used as a threshold.

Once a reasonable set of matching profiles is obtained, annotations of the included proteins can be searched for overrepresentation of a particular function. Overrepresented annotations reveal functional links to particular pathways, suggesting a putative role for the query protein, especially if the query protein in uncharacterized. As a test case, the profile of the *P. falciparum* protein PFB0445c was generated and compared with profiles of all known *P. falciparum* proteins. The results capture functional links between PFB0445c and other helicases in the parasite genome:

Query	PFB0445c (helicase, putative)
0.70	PF10_0309 (hypothetical protein)
0.69	MAL6P1.119 (DEAD/DEAH box ATP-dependent RNA helicase, putative)
0.61	MAL7P1.113 (DEAD box helicase, putative)
0.58	PF14_0436 (helicase, truncated, putative)
0.57	PFE0215w (ATP-dependent helicase, putative)

In this example, mutual information scores in the left column indicate confidence in the functional links; the greater the mutual information values, the greater the confidence in the predicted linkages. Comparison against the Protein families (Pfam) database (http://www.sanger.ac.uk/Software/Pfam/) reveals that the hypothetical protein PF10_0309 included in the results also contains helicase domains, demonstrating that the method captures biologically valid functional links. Profile data used for this example is available for download from the plasmoMAP website (http://cbil.upenn.edu/plasmoMAP/) ***(8)***. A score of 0.559, based on scores derived from a comparison of permuted profiles was used as the cutoff in this example.

As described previously, the input query set can be expanded to include the entire protein complement of any given genome. After profiles are constructed for all proteins, an all vs all comparison of profile similarity reveals functional linkages on a local and genome-wide scale. This is highly useful in understanding relationships between genes, and in some cases, has the ability to reveal new systems and pathways, especially if a majority of the components involved are of unknown function ***(5)***.

Table 1
A Sample of Results From Proteome-Wide Analysis of Functional Linkages in the Malarial Parasite *P. falciparum* Generated Using the Rosetta Stone Protocol

Protein A			Protein B			Rosetta sequence	
ID	Start	End	ID	Start	End	ID	Genome
PF10_0224	653	927	PFL2190c	314	481	37362641	*S. cerevisiae*
PF14_0314	261	431	PF13_0018	3	78	19913371	*H. sapiens*
PF08_0031	44	333	PF13_0208	1276	1427	NCU04792.1	*N. crassa*
PF08_0031	201	471	PFA0345w	19	471	33598954	*H. sapiens*
PFD1155w	581	1024	PFE0040c	1098	1231	11036632	*H. sapiens*
PF14_0126	424	642	PF14_0724	975	1064	24497618	*H. sapiens*
PF10_0320	216	743	PF11_0507	831	1022	30260710	*B. anthracis*
PF10_0320	90	322	PFE0280c	621	851	30018740	*B. cereus*

Table 2
Results of Rosetta Stone Analysis Arranged Differently, Than in Table 1

Protein A	Protein B	Number of fusions	Rosetta sequences
PF10_0224	PFL2190c	1	37362641|*S. cerevisiae*
PF14_0314	PF13_0018	1	19913371|*H. sapiens*
PF08_0031	PF13_0208	1	NCU04792.1|*N. crassa*
PF08_0031	PFA0345w	5	33598954|*H. sapiens*, 15241360|*A. thaliana*, NCU01564.1| *N. crassa*, 25152391|*C. elegans*, 39930485|*H. sapiens*
PFD1155w	PFE0040C	2	PFA0665w|*P. falciparum*, 11036632|*H. sapiens*
PF14_0126	PF14_0724	1	24497618|*H. sapiens*
PF10_0320	PF11_0507	2	30260710|*B. anthracis_Ames*, 30018740| *B. cereus_ATCC14579*
PF10_0320	PFE0280c	1	30018740|*B. cereus_ATCC 14579*

Proteins are displayed with information pertaining to all identified fusions between the pairs. The first two columns contain identifiers of the linked proteins, whereas the third column displays the total number of fusion sequences identified. The last column contains identifiers of the fusion proteins and organisms in which they are found.

3.2.2. The Rosetta Stone Method

The Rosetta stone fusion sequences can also be used to identify functional links between proteins. The method entails looking for two or more proteins that appear as a fused protein either in the same genome, or the genome of some other organism. The presence of a fusion protein indicates that the independent proteins are very likely to share a pathway, or are parts of pathways that are interlinked in some way, or even likely to physically interact with each other.

3.2.2.1. Application of the Method

Initial steps of the protocol for identifying functional links based on fusion proteins are similar to the protocol for generating phylogenetic profiles. The method requires a database of completely sequenced reference genomes, and computer programs to take an input file containing multiple amino acid sequences and compare them against the database. The BLAST results need to be parsed, or generated in a form where the attributes such as E-values and start–stop coordinates are retained. Once parsed BLAST output is available, it can be searched for sequences with nonoverlapping regions of similarity for two or more independent proteins. This can be algorithmically described as follows:

For any two proteins X and Y in a genome, identify all proteins (R) from a set of completely sequenced genomes (N), sharing similarities with both X and Y in distinctly different regions, where:

$$X_p \neq Y_p \neq R_{ij}; \text{ and}$$

$$S(R_{ij}, X_p)^{BEGIN} > S(R_{ij}, Y_p)^{END} \text{ or, } S(R_{ij}, Y_p)^{BEGIN} > S(R_{ij}, X_p)^{END}; \text{ and}$$

$$p \in N$$

In this formulation, S represents the region of similarity spanning all identified HSPs between the fusion protein R_i from genome j (contained in N), and proteins X and Y, from genome p, whereas *BEGIN* and *END* denote amino acid positions of the similarity span on protein R_{ij}. The E-value assigned to the span is the minimum E-value observed among all HSPs that consists of the match, provided all E-values are lower than 10^{-5} (*see* **Note 5**). This algorithm needs to be coded as a computer program by the user, and set to use parsed BLAST results as the input. The output should ideally contain identifiers of the individual proteins and that of the Rosetta stone sequence, name of the genome in which the fusion sequence was identified, and the *BEGIN* and *END* positions associated with the Rosetta stone sequence.

The following is an example of results obtained when the protocol was applied to the genome of *P. falciparum*. Amino acid sequences of all 5334

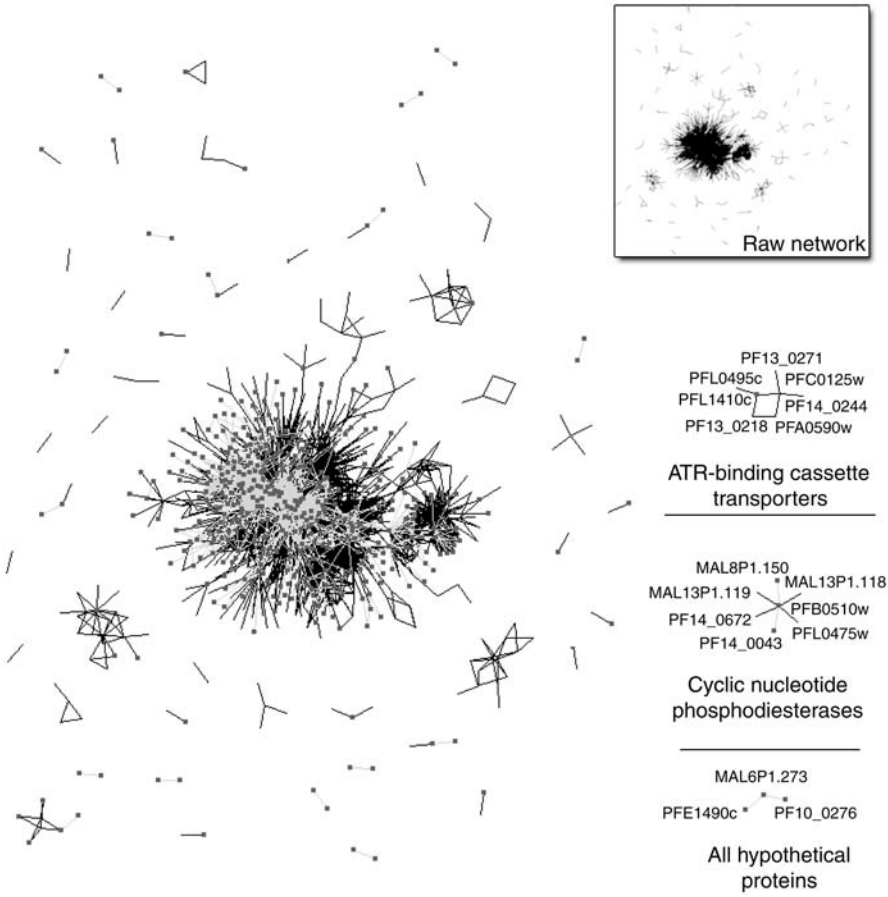


Fig. 1. A network of functional linkages in the human malarial parasite *P. falciparum* generated using the Rosetta stone method. Functional linkages were identified for the entire protein complement of the parasite using the Rosetta stone protocol. From this set, Rosetta links based on fusion sequences that were present only in the *P. falciparum* genome were discarded. Overall, 5176 links between 993 proteins were retained and used to reconstruct a part of the *P. falciparum* functional interaction network (*see* **inset**, proteins are represented as nodes and functional links as unweighted edges). Nodes and edges in the network are categorized based on their annotation; nodes representing hypothetical proteins are larger in size, and connections between hypothetical proteins are indicated in gray. Three examples of protein clusters that contain increasing numbers of hypothetical protein components are described as examples. Characteristics of the cluster help reveal characteristics of the included proteins and allow function assignments, as in the case of the hypothetical protein PFL1410c, which can be implicated in transport based on connections with a number of ATP-binding cassette transporters. Clusters made up entirely of novel proteins, such as the cluster containing hypothetical proteins MAL6P1.273, PFE1490c, and PF10_0276,

P. falciparum proteins were included in the input file, and the results indicated functional linkages between 993 unique proteins.

The output contains information about the protein pair linked by the fusion protein, along with information about the start and stop coordinates of the similarity span between the queries and the database hit. "A" and "B" represent the linked proteins, whereas "RS" indicates the Rosetta stone sequence that links the proteins together. The program can be coded to produce this or any other form of output. In the author's experience, multiple forms of the output that capture different aspects of the fusion are helpful in organizing information. For instance, the output arranged in the following manner allows the user to comprehend at a glance the extent of detected fusions:

Together, the different forms of output allow efficient analysis of the results. Quality of the results can be further enhanced and the occurrence of false-positives can be reduced by incorporating as filters, more features associated with the HSPs. *See* **Note 6** for additional information about enhancing result quality and choosing possible filters.

3.2.2.2. Testing Confidence of the Rosetta Linkages

To ensure the absence of errors, it is important to check the output of the program using known examples of fusion proteins. One well-known example of a fusion that can be used for testing data quality is the dihydrofolate reductase thymidylate synthase (DHFR-TS) protein in *P. falciparum*, which represents a fusion of the independently encoded dihydofolate reductase and thymidylate synthase in humans, or the yeast topoisomerase II protein, which links the *E. coli* proteins gyrA and gyrB. Any implementation of the method should be able to correctly identify functional links and fusion proteins, when the earlier examples are used for testing.

Although most functional links identified by this method are accurate, it is likely that some false-positives will be included in the result set, especially when dealing with genomes of higher-order eukaryotes. It is therefore best to statistically determine the validity of the results, such as by using a test

---

Fig. 1. *(Continued)* are likely to reveal previously unknown pathways or cellular systems. *P. falciparum* functional linkages derived using the Rosetta stone method are available for download from the plasmoMAP website (http://cbil.upenn.edu/plasmoMAP/) ***(8)***. The network was generated using the LGL package ***(19)***. Some independent clusters are repositioned for clarity.

described by Verjovsky Marcotte and Marcotte ***(16)***, which provides a confidence value for each predicted functional link. The test is introduced here in brief, and users are encouraged to refer to the original publication for details and tips on computational implementation. The author takes into account two types of possible ambiguities when determining probability of finding functionally linked proteins by random chance. First, the probability of finding k number of fusions by random chance is calculated based on the hypergeometric distribution, given the number of BLAST hits for proteins X and Y in a database of size N.

$$p(\text{number of fusions} \geq k \mid x,\ y,\ N) = 1 - \sum_{i=0}^{k-1} p(i \mid x, y, N)$$

Here, x and y represent hits to proteins X and Y in the database, respectively, and i represents a counter for summation.

Next, the author introduces a correction term, which addresses potential problems arising because of the presence of paralogs of the proteins X and Y.

$$p(X,\ Y \text{ are functionally linked in the presence of paralogs}) = 1 / \max\left(X_{\text{paralogs}}, Y_{\text{paralogs}}\right)$$

This term directly addresses problems encountered when deciding the accuracy of identifying proteins represented in the fusions; if X and Y are represented by single copies, then the probability of finding linked proteins will be one. The probability decreases as more paralogs of X and Y occur in the genome. The final probability of finding proteins X and Y linked by random chance given these conditions, is then simply the product of the two probabilities. Based on the information provided by the author, this score performs adequately when benchmarked against information derived from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database ***(17)***.

3.2.2.3. Expanding the Rosetta Stone Method

The Rosetta stone method can be extended to include the entire genome, whereby functional links are identified on a genome-wide scale, using the entire protein complement as input **(Fig. 1)**. The fullest potential of this method can be achieved when the query genome is a part of the database, and all sequences in the query genome are compared with each other. Besides identifying functional links, this also identifies fusion proteins in the query genome that serve to link independent proteins in other genomes. Thus, the entire landscape of functional linkages is revealed, along with information about fusions that might indicate coupling between pathways and systems.

3.3. Verifying the Value of Functional Linkages Obtained by Phylogenetic Profiling and the Rosetta Stone Method

Both protocols described in this chapter are representative of computational functional genomics methods that can be used to identify linkages between proteins on a large scale. Final verification of the usefulness and validity of the biological knowledge gained by the application of such methods is only possible in the experimental realm. However, evidence from other small- or large-scale experiments can prove quite helpful in corroborating results. It is also important to note that predictions from these and other methods, such as yeast two-hybrid and tandem-affinity purification experiments, may not necessarily overlap with each other, as each method independently captures only those parts of the interactome that are most accessible to it, and works best with proteins and linkages that fit the method's definitions. Therefore, increasingly, studies aimed at finding functional linkages and subsequently protein function now aim to combine data from diverse sources, resulting in more robust and confident reconstruction of biological scenarios ***(8,9,18)***. Ultimately, it is up to the users to decide how best to use the results, such that they prove maximally beneficial in understanding protein function and protein–protein relationships.

4. Notes

1. Several utilities that allow users to write and execute custom code are available for commonly used OS such as Microsoft Windows and Macintosh. In addition, programs like Cygwin (http://www.cygwin.com/) provide Windows users a UNIX-like environment to compile and run programs in common programming languages. Recent versions of the Macintosh OS (Mac OS X or higher) provide a "terminal" interface through the X11 suite.
2. Users familiar with the BLAST package will notice that a separate program is not necessary to run individual sequences included in a single input file, indeed, the "*blastall*" program can compare all proteins sequentially. This is a valid way of running BLAST; however, in some instances, error-checking can prove cumbersome. Advanced users can also try different BLAST output options, such as output in tabular format, which can at times eliminate the need for result parsing. Note that tabular output will report on a fixed number of attributes associated with the HSPs (*see* BLAST documentation). A number of BLAST parsing programs are freely available on the Internet. In addition, large, well-established packages such as BioPerl (http://www.bioperl.org), BioPython (http://www.biopython.org), and BioJava (http://www.biojava.org), associated with the PERL, Python (http://www.python.org/), and JAVA (http://www.java.com) languages, respectively, also make available modules that deal with BLAST results.
3. Often times, sequencing centers place restrictions on the use of complete genome sequencing data before its publication. Therefore, it is best to adhere to any agreements

or contracts put forth by the sequencing centers, and ask permission from the principal investigators before using the data.

4. In the author's experience, gene/protein identifiers are used in a sequence-specific context by the sequencing centers, meaning two or more sequences can possess the same identifiers. Users should ensure that all identifiers are unique, to prevent errors in the results.
5. It is always advisable to use a low E-value cutoff to avoid including possible false positives. For some genomes, especially those of higher eukaryotes, higher E-value cutoffs might be required to capture accurate sequence matches. An empirical survey of published literature reveals that authors usually trust and accept sequence matches with E-values of 10^{-5}.
6. Identification of correct functional links using fusion sequences is greatly affected by the presence of certain "promiscuous" domains (such as ATP-binding cassette domains). If possible, sequences with such domains should be identified and discarded during analysis, or placed in a separate low-confidence group. Other criteria for enhancing the value of the match can also serve to strengthen results. For instance, accepted matches that contain a high percentage of identical amino acids will certainly increase confidence in the results. Using a strong BLAST E-value cutoff can also prove beneficial in many instances (*see* **Note 5**).

References

1. Gardner, M. J., Hall, N., Funq, E., et al. (2002) Genome sequence of the human malaria parasite *Plasmodium falciparum*. *Nature* **419,** 498–511.
2. Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D., and Yeates, T. O. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA* **13,** 4285–4288.
3. Gaasterland, T. and Ragan, M. A. (1998) Microbial genescapes: phyletic and functional patterns of ORF distribution among prokaryotes *Microb. Comp. Genomics* **3,** 199–217.
4. Marcotte, E. M., Pellegrini, M., Ng, H. -L., Rice, D. W., Yeates, T. O., and Eisenberg, D. (1999) Detecting Protein Function and Protein-Protein Interactions from Genome Sequences. *Science* **285,** 751–753.
5. Date, S. V. and Marcotte, E. M. (2003) Discovery of uncharacterized cellular systems by genome-wide analysis of functional linkages. *Nat. Biotechnol.* **21,** 1055–1062.
6. Butland, G., Peregrin-Alvarez, J. M., Li, J., et al. (2005) Interaction Network Containing Conserved and Essential Protein Complexes in *Escherichia coli*. *Nature* **433,** 531–537.
7. Peregrin-Alvarez, J. M., Tsoka, S., and Ouzounis, C. A. (2003) The phylogenetic extent of metabolic enzymes and pathways. *Genome Res.* **13,** 422–427.
8. Date, S. V. and Stoeckert, C. J. (2006) Computational modeling of the *Plasmodium falciparum* interactome reveals protein function on a genome-wide scale. *Genome Res.* **4,** 542–549.
9. Lee, I., Date, S. V., Adai, A. T., and Marcotte, E. M. (2004) A probabilistic functional network of yeast genes. *Science* **306,** 1555–1558.

10. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool. *J. Mol. Biol.* **215,** 403–410.
11. Lopez, R., Silventoinen, V., Robinson, S., Kibria, A., and Gish, W. (2003) WU-Blast2 server at the European Bioinformatics Institute. *Nucleic Acids Res.* **31,** 3795–3798.
12. Wu, J., Kasif, S., and DeLisi, C. (2003). Identification of functional links between genes using phylogenetic profiles. *Bioinformatics* **19,** 1524–1530.
13. Shannon, C. E. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.* **27,** 379–423; 623–656.
14. Krober, B. T. M., Farber, R. M., Wolpert, D. H., and Lapedes, A. S. (1993) Covariation of mutations in the V3 loop of human immunodeficiency virus type I envelope protein: an information theoretic analysis. *Proc. Nat. Acad. Sci. USA* **90,** 7176–7180.
15. Huynen, M., Snel, B., Lathe, W., and Bork, P. (2000) Predicting protein function by genomic context: quantitative evaluation and qualitative inferences. *Genome Res.* **10,** 1204–1210.
16. Verjovsky Marcotte, C. J. and Marcotte, E. M. (2002) Predicting functional linkages from gene fusions with confidence. *Appl. Bioinforma.* **1,** 1–8.
17. Kanehisa, M., Goto, S., Hattori, M., et al. (2006) From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.* **34,** D354–D357.
18. Jansen, R., Yu, H., Greenbaum, D., et al. (2003) A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* **302,** 449–453.
19. Adai, A. T., Date, S. V., Wieland, S., and Marcotte, E. M. (2004) LGL: Creating a map of protein function with an algorithm for visualizing very large biological networks. *J. Mol. Biol.* **340,** 179–190.

8

Bioinformatics Tools for Modeling Transcription Factor Target Genes and Epigenetic Changes

Ramana V. Davuluri

Summary

The combinatorial control of gene regulatory switches involves both transcription factor (TF) complexes and associated epigenetic modifications to the chromatin template. The novel high-throughput technologies, such as Chromatin ImmunoPrecipitation ChIP-chip, have enabled genome-wide in vivo identification of TF target regulatory regions and related epigenetic modifications, which led to the view of highly dynamic TF–DNA interactions in activated or repressed promoters. Consequently, modeling and elucidating the combinatorial interaction of TFs and corresponding *cis*-regulatory modules in target promoters is of paramount interest. An estimated 5% of the genes in mammalian genomes code for TF proteins, and computational modeling of *cis*-regulatory logic would rapidly increase the pace of experimental confirmation of TF target promoters at the bench. The purpose of this chapter is to discuss the use of different bioinformatics tools for predicting the target genes of TFs of interest in mammalian genomes, and the application of these methods in the analysis of ChIP-chip experimental data. The author describes most commonly used databases and prediction programs that are available on the World Wide Web and demonstrate the use of some of these programs by an example. A list of these programs is provided along with their web Uniform Resource Locator (URLs) and guidelines for successful application are suggested.

Key Words: *Cis*-regulatory logic; computational modeling; CpG island; target promoter; transcriptional module; ChIP-chip.

1. Introduction

Transcriptional regulation of protein-coding genes involves a number of different levels of organization in the cell nucleus. The transcription machinery is responsible for the decoding and expression of genes in a regulatory network made up of various coregulatory complexes that are interconnected to control RNA synthesis from a given promoter ***(1)***. The key players of the transcriptional regulation are the transcription factors (TFs) that form complexes with other

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

interacting proteins and bind to sequence specific *cis*-regulatory elements or TF-binding sites (TFBSs) in gene promoters. These clusters of TFBSs, always of different types, and some represented multiple times, which occur synergistically in gene regulatory regions, are known as *cis*-regulatory modules ***(2)***. Emerging evidence suggests that alternative states of promoter activity (activation or repression) are influenced by interconnected assembly of transcriptional regulatory networks and epigenetic modifications at a chromatin template ***(3)***.

A key component in this epigenetic machinery is the occurrence of histone modifications around the promoter region of a gene ***(4–6)***, and different combinations of histone modifications may act synergistically or antagonistically to affect gene expression ***(7)***. It is known that acetylation of lysine 9 at histone H3 (H3-K9) is linked to transcriptional activation ***(8)***, whereas dimethylation of the same lysine seems to specify transcriptional repression ***(9)***. Regarding this, coregulatory proteins, which often possess chromatin modulating activities, appear to act cooperatively with partner TFs to establish patterns of gene expression, and thus, provide considerable functional flexibility in specifying transcriptional activation or repression ***(3,10)***.

Consequently, modeling the *cis*-regulatory modules in the activated or repressed target promoters of specific TFs is required to elucidate the transcriptional regulatory machinery. The first step in modeling the *cis*-regulatory modules is specifically identifying the TFBSs in the target promoters of TFs. Which gene promoters are targets of a given TF is partly determined by the DNA-binding domain of the TF protein. This domain allows the TF to bind to its specific TFBS in the target gene promoter. Extensive molecular research has provided a wealth of such information about experimentally characterized gene promoter sequences, TFs, and their binding sites (TFBS). Databases such as JASPAR ***(11)***, TRANSFAC ***(12)***, TRRD ***(13)***, and TFD ***(14)*** provide information about TFs and experimentally known TFBSs. In theory, the availability of these resources along with gene promoter databases, such as MPromDb ***(15)***, DBTSS ***(16)***, TRED ***(17)***, and EPD ***(18)***, should have made the task of finding TF target promoters a straightforward approach. For example, one can scan the promoter sequences in a genome of interest for the location of TFBS by using programs, such as MATCH ***(19)***, which uses TRANSFAC position weight matrices (PWMs). These PWM-based scanning programs are extremely useful for the identification of potential TFBS in a small promoter region around the Transcription Start Site (TSS) of a gene of interest, but produce too many false-positive predictions when applied to multiple promoters at genome level.

Consequently, determination of the TF targets is a daunting task ***(20)***. Recent programs have greatly improved in their TFBS prediction accuracy by incorporating sequence conservation information through phylogenetic footprinting ***(21–25)*** and by modeling the *cis*-regulatory modules ***(3,26–28)***.

The drawback of these integrative approaches is that these programs tend to miss many functional TFBSs that show very little sequence conservation even across modestly distant species, because of single-nucleotide substitutions and small indels within the regulatory regions.

Novel high-throughput technologies, such as ChIP-chip, have enabled genome-wide identification of the epigenetic mechanisms and protein–DNA interactions that effect gene expression ***(29)***. In ChIP-chip experiments, chromatin immunoprecipitation of specific protein/DNA complexes followed by microarray analysis is performed to probe a promoter microarray panel (e.g., CpG-island microarray panel ***[30]***). In recent years, the author ***(31–34)*** and others ***(35)*** have successfully used ChIP-chip assays to find the target genes of TFs in mammalian systems. The major focus of this chapter is to introduce different bioinformatics tools that identify TFBS in a set of genomic sequences, and discuss the application of these methods in the high-level analysis of ChIP-chip experimental data.

2. Materials

The user must have access to a computer with Internet access; for example, a PC running Microsoft Windows or Linux, an Apple Macintosh, or a UNIX workstation. The user should be familiar with the use of Netscape Navigator or Microsoft Internet Explorer, and the R statistical package http://www.r-project.org/. If the R programming package is not readily available the user can download the R base package from R-project website (through http://CRAN.R-project.org). The classification packages "rpart" and "randomForest" should be downloaded and installed in R. The user-friendly commercial CART software from Salford-systems (http://www.salford-systems.com) and the professional version of TRANSFAC from Genomatix (http://www.genomatix.de) would be helpful, but not necessary. The list of commonly used TFBS prediction programs based on PWM and phylogenetic footprinting approaches are provided in **Table 1**.

3. Methods

First an overview of the methodology is provided in **Subheading 3.1.**, then a worked example is presented in **Subheading 3.2.**

3.1. An Overview of In Silico Identification of TF Target Promoters

Quite a few methods are available to scan for TFBSs in a candidate promoter sequence. The simplest method of searching for a TFBS is by its consensus sequence of preferred nucleotides at specific positions of the binding site ***(36)***. Perhaps the most widely used method is the PWM approach, wherein a candidate TFBS is represented by a matrix of nucleotide scores reflecting the likelihood of each nucleotide at specific position ***(37)***. Although consensus sequence and

Table 1
Web URLs of Promoter, TF Databases, and TFBs Prediction Programs

Program name	Description	Organism	Web URL	References
TRANSFAC	Database of TFs, their genomic, binding sites and DNA-binding profiles	Eukaryotes	http://www.gene-regulation.com	***12***
JASPAR	Database of TF–DNA binding preferences, modeled as matrices	Eukaryotes	http://mordor.cgb.ki.se/cgi-bin/jaspar2005/jaspar_db.pl	***11***
MPromDb	Database of mammalian gene promoters	Mammals	http://bioinformatics.med.ohio-state.edu/MPromDb	***15***
OMGProm	Database of orthologous mammalian gene promoters	Human, mouse, and rat	http://bioinformatics.med.ohio-state.edu/OMGProm	***61***
DBTSS	Database of transcription start sites	Human, mouse, zebrafish, malaria, and schyzon	http://dbtss.hgc.jp	***16***
EPD	Database of eukaryotic gene promoters	Eukaryotes	http://www.epd.isb-sib.ch	***67***
UCSC	Genome browser at UCSC	Genome sequences and annotations including human–mouse–rat conserved blocks	http://genome.ucsc.edu	***68***
Match	Program to search for TFBSs using TRANSFAC PWMs	Eukaryotes	http://www.gene-regulation.com/pub/programs.html#match	***19***
TESS	Program to predict TFBSs	Eukaryotes	http://www.cbil.upenn.edu/cgi-bin/tess/tess	–
MatInspector	Program to search for TFBSs using TRANSFAC PWMs	Eukaryotes	http://www.genomatix.de/products/MatInspector/MatInspector1.html	***50***

rVISTA	Program to search for TFBSs by combining TRANSFAC PWMS and comparative sequence analysis	Human and mouse	http://genome.lbl.gov/vista/rvista/about.shtml	***57***
FootPrinter	Program to predict TFBSs by phylogenetic footprinting method	Many species with information of phylogenetic tree	http://genome.cs.mcgill.ca/cgi-bin/ FootPrinter3.0/FootPrinterInput2.pl	***59***
oPOSSUM	Program to analyze overrepresented TFBSs within a given set of promoters	Human and mouse	http://www.cisreg.ca/oPOSSUM2/opossum2.php	–
CisMols	Program to identify *cis*-regulatory modules in coexpressed gene promoters	Human and mouse	http://cismols.cchmc.org/peak-web/index.jsp	–
ConSite	Program to predict TFBSs using orthologous human and mouse genomic sequences	Human and mouse	http://mordor.cgb.ki.se/cgi-bin/CONSITE/consite	***58***

PWM-based models do not capture the complexity of TF–DNA interactions and produce too many false predictions at genome scale, these simple and easily interpretable models provide a very good approximation to reality ***(38)***. To reduce the number of predictions found by chance, recent methods have incorporated additional information, such as use of complex sequence motif models ***(39–41)***, conservation of TFBSs in orthologous promoters of closely related species ***(42–45)***, and clustering of binding sites in promoters of coregulated genes ***(3,24,46)***. In this protocol, a combination of sequence conservation and clustering of TFBSs of known PWMs is described in predicting and classifying the target promoters (*see* **Note 1**). Readers are encouraged to read recent reviews ***(47–49)*** for practical strategies to scan for TFBSs.

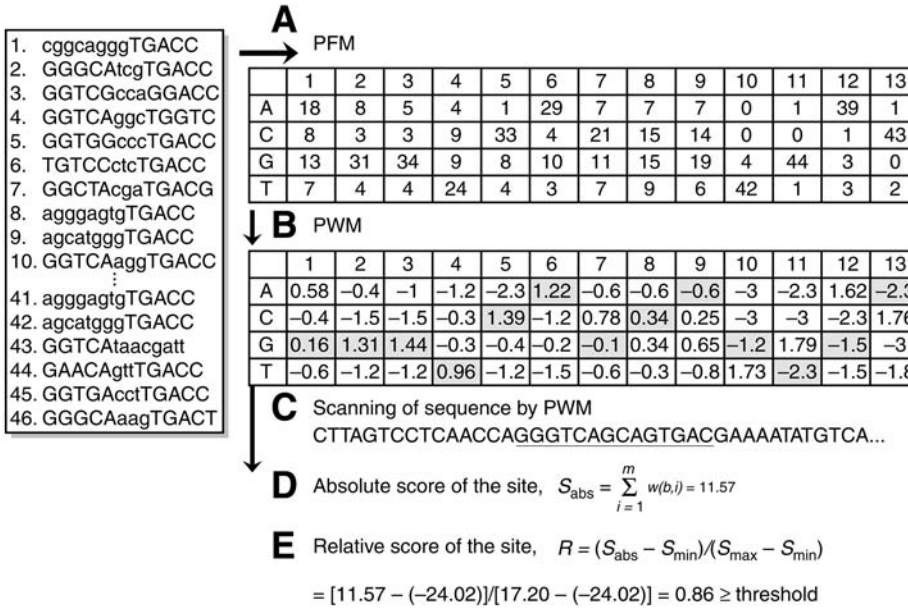
3.1.1. Identifying Candidate TFBSs by PWM Approach

A number of databases of experimentally supported TFBSs have been assembled **(Table 1)**. The largest and perhaps most widely used databases are TRANSFAC ***(12)*** and JASPAR ***(11)***, which catalog eukaryotic TFs, associated binding sites, and PWMs. Similarly, PWM-based sequence scanning programs, such as MatInspector ***(50)***, MATCH ***(19)***, and MATRIX SEARCH ***(51)***, can be used to search the query sequences for candidate TFBSs by matching the corresponding PWMs. These programs are quite similar in the use of PWM databases (e.g., TRANSFAC or JASPAR) and statistically principled methods in scoring the sites.

Choosing a cutoff threshold for the PWM score is the main requirement in determining whether a sequence site is a putative TFBS or not, and the number of TFBS predictions in a candidate sequence is inversely proportional to the cutoff values. A basic procedure to scan a query sequence using PWM is illustrated in **Fig. 1**. MATCH uses the matrix library collected in the TRANSFAC database. MATCH has built-in optimized matrix cutoff values (called profiles), which were precalculated to provide three different search modes of varying stringency. The user can choose one of these three predefined profiles: (1) minFP—cutoffs minimizing false-negative rate, (2) minFN—cutoffs minimizing false-negative rate, and (3) minSum—cutoffs minimizing the sum of both errors. The use of minSum profile is suggested, because sequence conservation is added as an additional criterion to minimize the false-positive predictions in the next step.

3.1.2. Identification of Conserved TFBSs in Orthologous Promoters

As PWM-based methods tend to produce an overwhelming number of false-positives, phylogenetic footprinting or comparative genomics approach has been widely used by both experimental and computational biologists to aid regulatory element identification by examining orthologous sequences from multiple species ***(52)***. Recent studies ***(28,53,54)*** have identified blocks of highly conserved regions



	1	2	3	4	5	6	7	8	9	10	11	12	13
A	18	8	5	4	1	29	7	7	7	0	1	39	1
C	8	3	3	9	33	4	21	15	14	0	0	1	43
G	13	31	34	9	8	10	11	15	19	4	44	3	0
T	7	4	4	24	4	3	7	9	6	42	1	3	2

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0.58	–0.4	–1	–1.2	–2.3	1.22	–0.6	–0.6	–0.6	–3	–2.3	1.62	–2.3
C	–0.4	–1.5	–1.5	–0.3	1.39	–1.2	0.78	0.34	0.25	–3	–3	–2.3	1.76
G	0.16	1.31	1.44	–0.3	–0.4	–0.2	–0.1	0.34	0.65	–1.2	1.79	–1.5	–3
T	–0.6	–1.2	–1.2	0.96	–1.2	–1.5	–0.6	–0.3	–0.8	1.73	–2.3	–1.5	–1.8

Fig. 1. TFBS prediction by PWM approach: a schematic representation of position frequency matrix (PFM) and PWM construction. The PWM is used to scan the promoter sequences to score the TFBSs. Experimentally known EREs from Jin et al. ***(24)*** were used to prepare the PFM. PFM is converted to PWM, which is in log scale for efficient computational analysis.

in orthologous human and mouse promoter sequences. Systematic analyses of sequence conservation in known human TFBSs was performed to demonstrate the use of sequence conservation as a useful criterion to identify putative binding sites ***(55,56)***. Several phylogenetic footprinting methods have been developed for identifying putative binding sites for TFs with known PWMs.

Alignment-based methods, such as rVISTA ***(57)*** and ConSite ***(58)***, first compute a multiple sequence alignment of the set of orthologous sequences and then scan for TFBSs in the conserved regions using PWMs. If the regulatory regions are too divergent and fail to align properly, *de novo* programs, such as FootPrinter ***(42,59)***, can be used to identify sets of subsequences that exhibit a high degree of conservation given a set of orthologous input sequences and the phylogenetic tree relating them. Blanco et al. ***(60)*** proposed a new approach in which they first translate the nucleotide sequence of the promoter into an alphabet of different symbols representing different TFs of known PWMs. These TF-maps of two related promoters (e.g., of orthologous or coexpressed genes) were then aligned using global pair-wise alignment method. In this protocol, the use of alignment-based methods or

TF-map alignments of orthologous promoters is suggested. The effectiveness of either of the approaches largely depends on the quality and availability of orthologous promoter sequences. The use of OMGProm database ***(61)***, which contains the promoters of orthologous mammalian genes and their sequence alignments is suggested.

3.1.3. Classification of Target Promoters From Nontargets and Inferring cis-Regulatory Modules (TFBS Clusters) Using Decision Tree Methodology

TF interaction is an important aspect of mammalian gene regulation. Through the fine tuning of different partners, a specific TF could involve in different cellular processes and achieve opposite downstream effects by either activating or repressing the direct target promoters ***(3)***. Different methods to infer *cis*-regulatory modules in a given set of target promoters have been developed ***(3,24,46,62,63)***. Most of the methods rely on discriminating a set of target promoters from nontarget promoters by using TFBSs or sequence motifs as feature variables in classification function. The best discriminating feature variables (e.g., TFBSs) are then extracted to infer the *cis*-regulatory modules. In this protocol, the use of decision tree approaches is recommended for their simplicity and interpretability.

Tree-based statistical methods have become increasingly popular since the publication of the CART monograph ***(64)***. These approaches have many advantages over discriminant analysis, as tree-based models are easy to interpret, are nonparametric, and make no assumptions regarding the covariance structure of the two groups. CART analysis provides a better understanding of the dependence of the response variables (y_i) (promoter status—target or nontarget in the present case) on the structure of the relationships of potential explanatory variables (x_i) (e.g., TFBSs—present or not present in a given promoter) and their combinations, together with their high-level interactions. If (y_i) is binary, CART produces a classification tree, whereas if the response variable is continuous, a regression tree is produced. In essence, CART uses recursive partitioning and asymmetric stratification to develop tree-like models. CART splits the data at a parent node by determining a cutoff value along the range of values for an explanatory variable, thus producing two child nodes with greater homogeneity (purity) than the parent node.

Child nodes are recursively treated as parent nodes, thereby continuously splitting the data until a stopping criterion is reached and a set of terminal nodes are produced, which in total resemble an inverted tree. Overfitted trees are grown, and then pruning trims the trees to a more optimal size using test samples or cross-validation. Each terminal node is assigned a class that is determined by the

class representation in that group. The resulting model is a highly interpretable decision tree, which helps design further experiments. Some of the principal limitations of CART are low accuracy (because of the use of piece-wise, constant approximations) and high variance or instability. In particular, when the number of variables (TFBSs) is much larger than the number of observations (promoters), CART would fail to give a robust classification model (*see* **Note 3**).

In order to limit the number of variables for CART analysis, one can use the Random Forest program ***(65)*** to preselect the most discriminative variables from a large number of input variables. Random forest is an ensemble of many decision trees, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. To classify a new object from an input vector, the algorithm applies the input vector to each tree of the forest. Each tree is a separate classification model, and the tree "votes" for that class. The forest then chooses the classification having the most votes over all of the trees in the forest. The forest error rate depends on the correlation between any two trees in the forest (increasing the correlation increases the forest error rate) and the strength of each individual tree in the forest (a tree with a low error rate is a strong classifier and increasing the strength of the individual trees decreases the forest error rate).

Random Forest can handle thousands of input variables without variable selection and gives estimates of what variables are important in the classification. Although Random Forest is a robust classifier, the black box nature of the algorithm makes it impracticable to infer the decision rules from thousands of trees. In the present case, it is critical to understand the interaction of variables (TFs) that provide the predictive accuracy. Hence, the use of Random Forest for variable selection followed by application of CART algorithm is recommended.

The commercially available CART program ***(66)*** is perhaps the best and is user-friendly, and the authors have used it in their earlier studies ***(3,24)***. If the commercial program is not available, the user may use rpart, a free implementation of CART in the R statistical package. Similarly, the freely available implementation of Random Forest in R can be used for variable selection. The author suggests "Gini" method as the splitting method for growing the tree and the 10-fold cross-validation to obtain the optimal minimal tree. TFBSs predicted by MATCH and conserved in the human and mouse orthologous promoters can be used as predictor variables, wherein each binding site may be considered as a binary variable, such that it was either 1 or 0, depending on its presence or absence within a specified region.

3.2. Worked Example

Various methods are discussed to predict TFBSs in a given promoter and decision tree classification methods in the previous sections. Now will be

discussed which programs to choose and how to use those programs in practice. Given two sets of genome coordinates (usually genomic locations of target probes and nontarget probes from ChIP-chip experimental data analysis) the following steps are recommended in classifying target promoters from nontargets and inferring the *cis*-regulatory modules. Several alternatives for targets and nontarget sets are possible (e.g., acetylated promoters vs methylated promoters, target promoters of specific TF vs nontarget promoters, and methylated- vs unmethylated-CpG islands).

3.2.1. Procedures

1. Retrieve human and mouse orthologous sequence regions by extending 500 bp at both ends of each probe of the input set of genome coordinates. Each of the retrieved sequences would then be of length 1 kb plus the probe length (60 bp in case of Agilent promoter array Agilent Technologies [http://www.home.agilent.com.]), with a corresponding orthologous region of similar length in human or mouse. The user can use either OMGProm or USCS genome browser to retrieve the sequences.
2. Use MATCH to predict the TFBSs in each of the sequences by using the minSum cutoff profile.
3. Consider the conserved TFBSs by comparing the MATCH predictions in orthologous promoter pairs. *See* **Note 2** for alternative approaches to predict TFBSs.
4. Choose a primary TF of interest and locate its conserved binding sites. For example, Estrogen Recepter (ER)-α would be primary TF of interest if the ChIP-chip data was obtained by using antibody against ER-α TF. If multiple TFBSs of primary TF are predicted within a given sequence region, choose the TFBS closest to the center of the probe.
5. Locate all the TFBSs within −220- to +220-bp region of primary TFBS for each sequence. Prepare a data matrix (x_{ij}), in which *i*-th row (object) and *j*-th column (variable) correspond to *i*-th promoter and *j*-th TF, respectively. The data matrix is binary in nature, such that $x_{ij} = 1|0$ depending on *j*-th TF has its binding site located or not located in *i*-th promoter. Similarly, prepare the classification vector or response variable y_i, such that $y_i = 1|0$ depending on whether *i*-th promoter is target or nontarget. The user may also use the actual counts (number of TFBSs present in the promoter for each TF) in the data matrix, in which case the data matrix is not binary but quantitative in nature.
6. Run RandomForest program in R console by using "randomForest" command (e.g., ER.rf <– randomForest[*x* = X, *y* = y, importance = T, proximity = T, ntree = 1000, strata = y, sampsize = c(23, 23)], where X is the data matrix of feature variables and y is the classification vector.
7. Use the function "importance" to extract the variable importance. For example, the R command "ER.imp <– importance(ER.rf)" gives a matrix of number of classes +2 columns (four columns for two class problem), in which the first two columns are the class specific mean decrease in accuracy, the third column is the mean decrease in accuracy, and the last column is the mean decrease in Gini index.

Select a subset of variables by removing a certain percentage (20–25%) of the least important variables by using mean decrease in accuracy and/or Gini index values in columns 3 and 4 of the importance matrix (*see* **Notes 3** and **4**).

8. Repeat **steps 8** and **9** by using the selected subset of variables 5–10 times, and finally select a subset of 10–20% of the most important variables.
9. Using the subset of variables selected in **step 9**, run CART to produce the decision tree that classifies the two sets of promoters. The splitting rules at each node can be interpreted as "if-then" statements to determine what TFBS are present in each class of promoters to determine the regulatory modules.

3.2.2. Results of Application to ER-α Targets

The above steps (except the Random Forest **steps 8** and **9**) have been successfully implemented in the earlier studies to classify ER-α targets from nontargets ***(24)*** and acetylated ER-α targets from methylated ER-α targets ***(3,24)***. A manuscript describing the above algorithm is currently under preparation. An automated version of the computational pipeline would soon be made available (*see* **Note 5**).

To demonstrate the above steps, the ER-α target data set from Cheng et al. ***(3)*** consisting of acetylated ER-α promoters (target set) and methylated ER-α promoters (nontarget set) are used. Briefly, ChIP-chip experiments were conducted by probing the 12 K CpG-island microarray ***(30)*** with series of different ChIP assays using antibodies against ER-α, acetyl-, and dimethyl-H3-K9 in MCF7 cells treated with E2 for 0, 3, 12, and 24 h. Integrated statistical and genome analysis of these data identified 92 ER-α target promoters, of which 40 were classified as acetylated (upregulated) and 28 as methylated (downregulated) targets. Retrieve human and mouse orthologous promoter sequences that correspond to these probes from OMGProm database, and ran MATCH program on both human and mouse sequences. First find the TFBSs of ER-α (primary TF of interest). **Table 2** gives the list of genes, and genomic coordinates of the sequences analyzed. Then locate all the TFBSs within −220 and +220 region around the predicted ERE, and prepare the data matrix as explained in **step 7**. **Table 3** presents part of the data matrix, which includes the top ranking TFs as determined by Random Forest variable importance (in **step 9**). The original data matrix contains all the TFs that have at least one TFBS in 20% of either of the promoter sets. **Figure 2** presents the plot of variable importance obtained in **step 9**. Then select the top 10 ranking variables, ranked according to the mean decrease in accuracy, for **step 10**. Here the number of variables selected was arbitrarily chosen; user should repeat **step 10** by varying this number. Using the subdata matrix that contains only the selected 10 variables run CART and/or rpart program.

Figure 3A presents a minimal cost tree constructed based on these TFBSs as the categorical predictor variables. The prediction rate based on 10-fold

Table 2
ER-a-Responsive Regions

Type	Gene-ID	Human sequence (−220 to +220 of ERE)	Mouse orthologous sequence
Ac	333	chr19:41051154-41051607(+)	chr7:25859980-25860432(−)
Ac	595	chr11:69162963-69163416(+)	chr7:139353963-139354415(()
Ac	652	chr14:53494338-53494791(−)	chr14:41471482-41471934(−)
Ac	1380	chr1:204014722-204015175(+)	chr1:194917960-194918412(−)
Ac	1745	chr2:172774828-172775281(+)	chr2:71226655-71227107(+)
Ac	2553	chr15:48434500-48434953(−)	chr2:126189214-126189666(−)
Ac	3099	chr2:74973752-74974205(+)	chr6:83116970-83117422(−)
Ac	4201	chr6:43088557-43089010(−)	chr17:44192432-44192884(+)
Ac	4207	chr19:19161832-19162285(−)	chr8:69294338-69294687(+)
Ac	4609	chr8:128818383-128818836(+)	chr15:61998428-61998880(+)
Ac	5018	chr14:22305623-22306076(+)	chr14:48877700-48878152(+)
Ac	6925	chr18:51407456-51407909(−)	chr18:69575125-69575577(+)
Ac	7057	chr15:37659195-37659648(+)	chr2:117624437-117624889(+)
Ac	7779	chr1:208140357-208140810(−)	chr1:191646305-191646757(+)
Ac	7779	chr1:208140357-208140810(−)	chr1:191646473-191646925(+)
Ac	8317	chr1:91679122-91679575(+)	chr5:106027962-106028414(+)
Ac	8615	chr4:77006595-77007048(+)	chr5:91473337-91473789(+)
Ac	9908	chr4:77006600-77007053(−)	chr5:91473341-91473793(−)
Ac	11273	chr16:28743036-28743489(+)	chr7:120551563-120552015(−)
Ac	25939	chr20:35013301-35013754(−)	chr2:156591900-156592352(−)
Ac	29090	chr18:69966999-69967452(+)	chr18:85118618-85119070(−)
Ac	51110	chr8:71743847-71744300(−)	chr1:13793443-13793895(−)
Ac	53373	chr12:112121480-112121933(+)	chr5:119740487-119740939(−)
Ac	54433	chr4:111094083-111094536(+)	chr3:128764022-128764474(−)
Ac	54737	chr13:19105862-19106315(+)	chr14:51190475-51190927(+)
Ac	55920	chr1:17511227-17511680(−)	chr4:139582591-139583043(+)
Ac	60314	chr12:51979988-51980441(+)	chr15:102392356-102392808(+)
Ac	79694	chr6:96132139-96132592(+)	chr4:26481957-26482409(−)
Ac	79980	chr20:34833961-34834414(−)	chr2:156462406-156462858(−)
Ac	80256	chr9:35105594-35106047(−)	chr4:42960161-42960613(−)
Ac	80256	chr9:35105594-35106047(−)	chr4:42961363-42961815(−)
Ac	81603	chr10:104391986-104392439(+)	chr19:46047350-46047802(+)
Ac	84447	chr11:64658309-64658762(−)	chr19:5835661-5836113(+)
Ac	116138	chr6:43089208-43089661(+)	chr17:44191823-44192275(−)
Ac	127933	chr1:159199138-159199591(+)	chr1:170189611-170190071(−)
Ac	136319	chr7:135118928-135119381(−)	chr6:35633646-35634098(−)
Ac	144608	chr12:14847000-14847453(+)	chr6:137599279-137599731(+)
Ac	153364	chr5:89806716-89807169(−)	chr13:77784071-77784523(+)
Ac	284403	chr19:41237537-41237990(+)	chr7:25693030-25693482(−)

(Continued)

Table 2 ***(Continued)***

Type	Gene-ID	Human sequence (−220 to +220 of ERE)	Mouse orthologous sequence
Me	119	chr2:70907791-70908244(−)	chr6:86456544-86456996(+)
Me	2313	chr11:128068485-128068938(+)	chr9:32463271-32463723(−)
Me	4035	chr12:55809318-55809771(+)	chr10:127356571-127357023(−)
Me	5048	chr17:2444402-2444855(+)	chr11:74448879-74449331(−)
Me	5101	chr13:66702595-66703048(−)	chr14:88356851-88357303(−)
Me	5783	chr4:87872450-87872903(+)	chr5:102458949-102459401(+)
Me	7702	chr11:9438737-9439190(+)	chr7:103914186-103914638(+)
Me	7745	chr6:28213421-28213874(+)	chr13:21014835-21015287(−)
Me	8667	chr8:117846957-117847410(−)	chr15:51873755-51874207(−)
Me	9774	chr6:136651331-136651784(−)	chr10:20240553-20241005(+)
Me	11096	chr21:27259775-27260228(−)	chr16:84999666-85000118(−)
Me	53335	chr2:60689565-60690018(−)	chr11:23975167-23975619(+)
Me	55064	chr9:4656506-4656959(−)	chr19:28220480-28220932(−)
Me	79661	chr15:73426510-73426963(+)	chr9:57261658-57262110(−)
Me	80207	chr19:50779773-50780226(−)	chr7:16096867-16097319(+)
Me	80309	chr2:228872394-228872847(−)	chr1:83738473-83738925(−)
Me	85015	chr6:100069931-100070384(−)	chr4:21845309-21845761(+)
Me	116092	chr20:43853917-43854370(+)	chr2:164202655-164203107(+)
Me	140775	chr17:18159358-18159811(+)	chr11:60503208-60503660(+)
Me	283078	chr10:28074133-28074586(−)	chr18:7046824-7047276(+)
Me	200558	chr2:68606356-68606809(+)	chr6:88107914-88108471(+)
Me	92014	chr9:37894330-37894344(−)	chr4:45324576-45325138(+)
Me	7072	chr2:70387601-70387615(−)	chr6:86876812-86877270(+)

Obtained from Cheng et al. ***(3)***. The locations of the human sequence around the consensus ERE and the mouse orthologous region are given in columns 3 and 4. The genomic coordinates are according to the human (May 2004 Build—hg17) and the mouse (March 2005—mm6) assemblies.

cross-validation was 86% for acetylated targets and 84% for methylated targets. Based on the splitting rules at each node and applying the "if then" rules in the CART tree, three *cis*-regulatory modules i.e., ERE + MYC, ERE + MYB, and ERE + E47 + CETS168, were identified for upregulated (i.e., more acetylated) targets and four modules (ERE + HNF3α, ERE + AP3, ERE + E2A, and ERE + E47) were identified for downregulated (i.e., more methylated) targets **(Fig. 3B)**. Overall, CART and Random Forest analyses identified seven distinct *cis*-regulatory modules for up- or downregulated ER-α target genes. The user should try different top ranking variables (TFs) to construct various CART trees and should focus on those modules that are predicted consistently.

4. Notes

1. Despite great progress, TFBS prediction by computational approaches alone is still far from perfect. The existing programs that combine PWM and comparative

Table 3
Sample Data Matrix Representing the Presence or Absence of Binding Sites of Top Ranking TFs

Class	MYCMAX	MYC	VMYB	GATA2	P53	TCF11	EFC	CETS168	MAZ	STAT4	CRX	SOX5	FOXD3	HNF3α	E2A	AP3	AML1	E47	TCF4	HIF1
Ac	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Ac	1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0
Ac	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0
Ac	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
Ac	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ac	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Ac	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Ac	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0
Ac	1	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
Ac	1	1	1	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0
Ac	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Ac	0	0	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0
Ac	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Ac	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
Ac	1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1
Ac	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Ac	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1
Ac	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ac	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ac	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Ac	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
Ac	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0
Ac	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0
Ac	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

Ac	0	0	0	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	
Ac	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	
Ac	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	
Ac	0	0	0	1	1	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	
Ac	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	
Ac	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Ac	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	
Ac	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	
Ac	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
Ac	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	
Ac	1	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	
Ac	1	1	0	1	1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	
Ac	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	
Me	0	0	0	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	
Me	0	0	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	
Me	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	
Me	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	
Me	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	
Me	0	0	1	1	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0	
Me	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	
Me	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	
Me	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	
Me	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	
Me	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	
Me	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	1	0	0	
Me	0	0	0	0	1	0	0	0	1	1	0	1	1	1	0	1	1	1	0	0	
Me	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	0	0	

(Continued)

Table 3 ***(Continued)***

Class	MYCMAX	MYC	VMYB	GATA2	P53	TCF11	EFC	CETS168	MAZ	STAT4	CRX	SOX5	FOXD3	HNF3α	E2A	AP3	AML1	E47	TCF4	HIF1
Me	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Me	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0
Me	0	0	0	1	1	1	0	0	1	1	1	1	0	0	0	0	0	1	1	0
Me	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
Me	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0
Me	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
Me	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
Me	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
Me	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
Me	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Me	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	0

The first column is the class variable (y), which is a categorical variable with two values; Ac–Acetylated promoter and Me–Methylated promoter. The rest of the columns represent different Transcription Factor Binding Sites (TFBS), used as feature (predictor) variables for running Random Forest and CART/rpart. It is a binary matrix, where 1 and 0 stand for the presence and absence of TFBS (column) in the corresponding promoter (row).

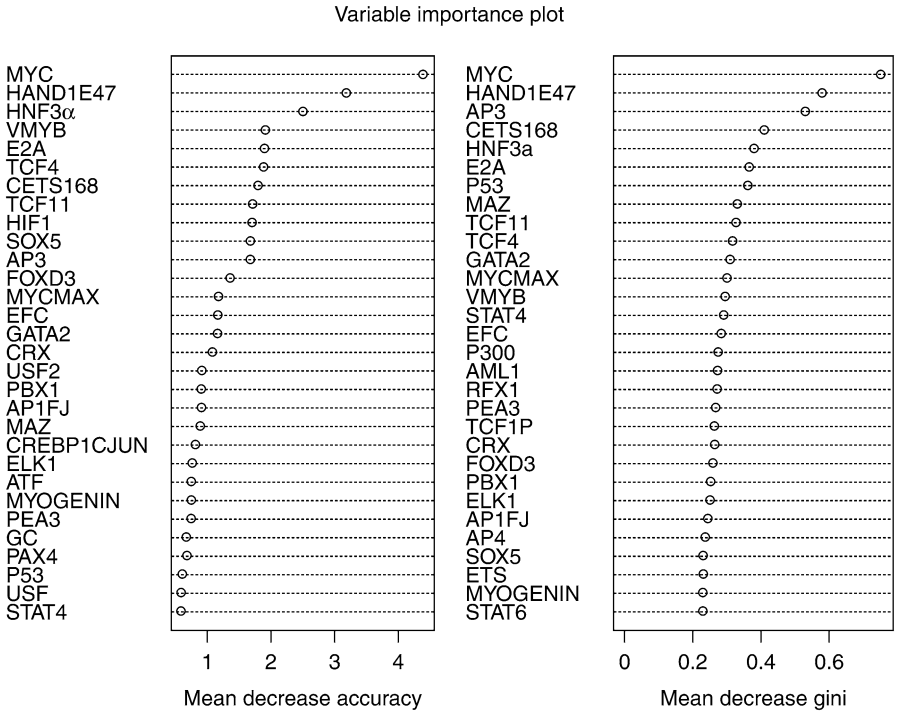


Fig. 2. Variable importance: relative importance of top ranking variables given by Random Forest analysis, ranked in the decreasing order of importance with respect to mean decrease in accuracy and Gini measures.

genomics approaches have reached a reasonable sophistication in identifying TFBSs of known PWMs ***(48)***. Using combination of different programs and taking the consensus predictions for considering the reliable predictions is suggested. But, the author expects some amount of noise in terms of false predictions and missed real TFBS within a given promoter. Further, inclusion of novel TFBS is not considered in the approaches suggested in this chapter, although one can make PWM and include it as a new variable. However, even the partial predictions are of immense value to design the experiments that can determine the regulatory modules faster than would be possible by experimental methods alone.

2. Recent programs, such as rVISTA ***(57)*** and ConSite ***(58)***, incorporate both sequence conservation across orthologous promoters and high-quality PWM models in producing more reliable TFBSs predictions.
3. Dimensionality reduction is an important problem in pattern recognition. In most of the experimental situations, lot more number of features/variables (TFs) are available than the number of cases (promoters). Selecting the appropriate number of features to build the classifier is an important problem, and Random Forest helps to reduce the dimensionality of feature space for effective classification ***(65)***.

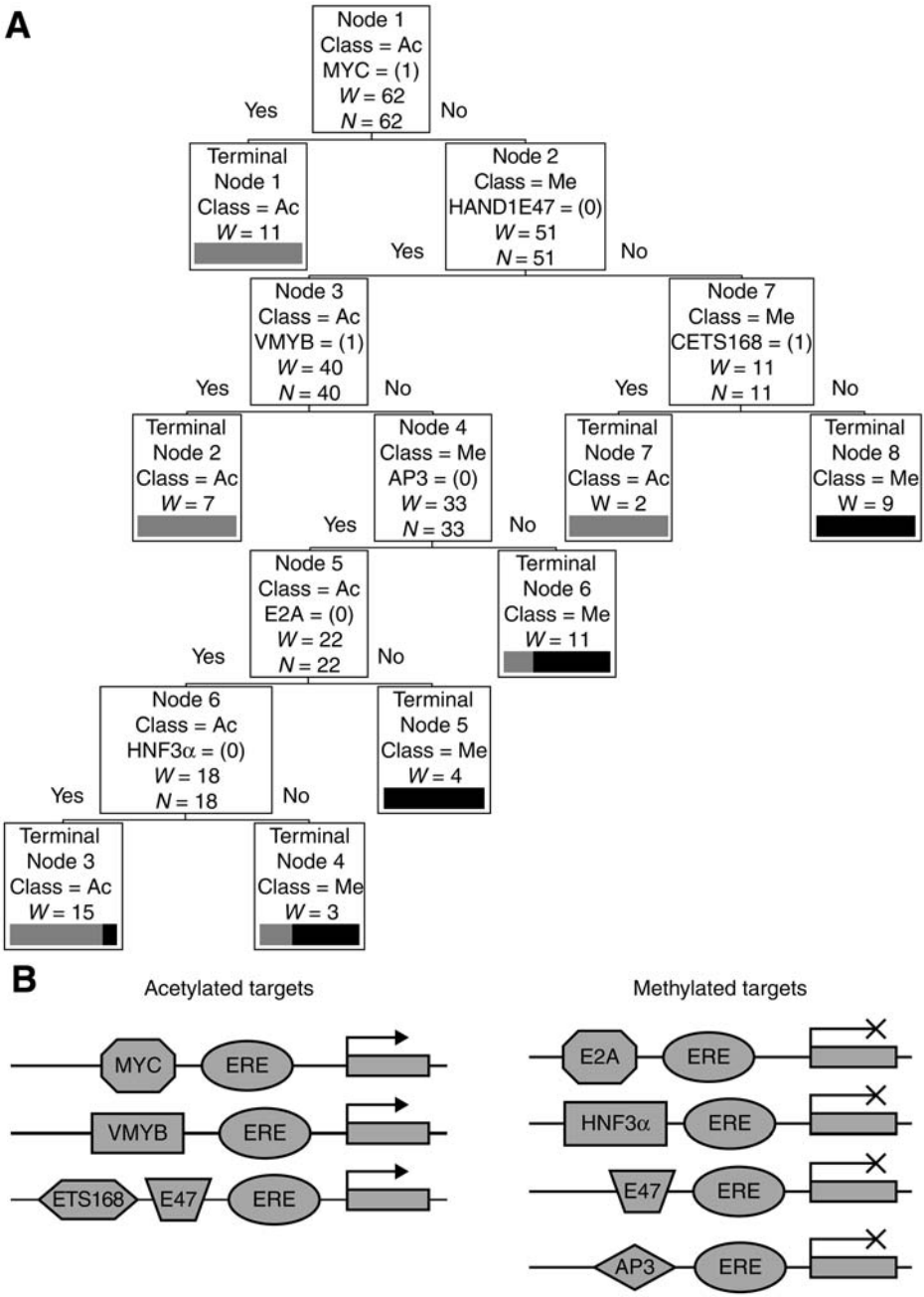


Fig. 3. **(A)** CART Tree: a CART model that discriminates between ER-α upregulated (more acetylated) and downregulated (more methylated) targets. **(B)** *Cis*-regulatory modules: Modules of ER-α upregulated (acetylated) and downregulated (methylated) target genes identified by the CART model.

The user should run the Random Forest program repeatedly, each time dropping a certain percentage (e.g., 20%) of the least important variables. Finally, the user should focus on the top ranking variables for constructing the CART tree. Even if some of these top ranking variables are not included in the initial CART tree, running CART repeatedly by selecting different combinations of the variables would give a better idea of possible TFBS modules. The list of ranked feature variables **(Fig. 2)** is valuable information for prioritizing the TFs for further experimental verification.

4. The following sequence of R commands ranks the variables in the decreasing order of mean decrease in accuracy (rf.imp[,3]) and Gini index (rf.imp[,4]), and selects the union of top 10 ranking variables either according to mean decrease in accuracy or Gini index.
   a. rank1 <- rank(rf.imp[,3]).
   b. rank2 <- rank(rf.imp[,4]).
   c. rankTF <- rep(F, length(rank1)).
   d. rankTF[rank1>=(ncol(x) − 10) | rank2>=(ncol(x) − 10)]<-T.
   e. X.sel <- x[, rankTF].
5. An automated pipeline would soon be made available at http://bioinformatics.med.ohio-state.com. The users may contact the author for R code.

References

1. Hochheimer, A. and Tjian, R. (2003) Diversified transcription initiation complexes expand promoter selectivity and tissue-specific gene expression. *Genes Dev*. **17,** 1309–1320.
2. Longabaugh, W. J., Davidson, E. H., and Bolouri, H. (2005) Computational representation of developmental genetic regulatory networks. *Dev. Biol*. **283,** 1–16.
3. Cheng, A. S., Jin, V. X., Fan, M., et al. (2006) Combinatorial Analysis of Transcription Factor Partners Reveals Recruitment of c-MYC to Estrogen Receptor-alpha Responsive Promoters. *Mol. Cell* **21,** 393–404.
4. Kurdistani, S. K. and Grunstein, M. (2003) Histone acetylation and deacetylation in yeast. *Nat. Rev. Mol. Cell Biol*. **4,** 276–284.
5. Metivier, R., Penot, G., Hubner, M., et al. (2003) Estrogen receptor-alpha directs ordered, cyclical, and combinatorial recruitment of cofactors on a natural target promoter. *Cell* **115,** 751–763.
6. Xu, J. and Li, Q. (2003) Review of the in vivo functions of the p160 steroid receptor coactivator family. *Mol. Endocrinol*. **17,** 1681–1692.
7. Jenuwein, T. and Allis, C. D. (2001) Translating the histone code. *Science* **293,** 1074–1080.
8. Roh, T. Y., Cuddapah, S., and Zhao, K. (2005) Active chromatin domains are defined by acetylation islands revealed by genome-wide mapping. *Genes Dev*. **19,** 542–552.
9. Peters, A. H., Kubicek, S., Mechtler, K., et al. (2003) Partitioning and plasticity of repressive histone methylation states in mammalian chromatin. *Mol. Cell* **12,** 1577–1589.
10. McKenna, N. J. and O'Malley, B. W. (2002) Combinatorial control of gene expression by nuclear receptors and coregulators. *Cell* **108,** 465–474.

11. Vlieghe, D., Sandelin, A., De Bleser, P. J., et al. (2006) A new generation of JASPAR, the open-access repository for transcription factor binding site profiles. *Nucleic Acids Res*. **34,** D95–D97.
12. Wingender, E., Chen, X., Fricke, E., et al. (2001) The TRANSFAC system on gene expression regulation. *Nucleic Acids Res*. **29,** 281–283.
13. Kolchanov, N. A., Ignatieva, E. V., Ananko, E. A., et al. (2002) Transcription Regulatory Regions Database (TRRD): its status in 2002. *Nucleic Acids Res*. **30,** 312–317.
14. Ghosh, D. (2000) Object-oriented transcription factors database (ooTFD). *Nucleic Acids Res*. **28,** 308–310.
15. Sun, H., Palaniswamy, S. K., Pohar, T. T., Jin, V. X., Huang, T. H., and Davuluri, R. V. (2006) MPromDb: an integrated resource for annotation and visualization of mammalian gene promoters and ChIP-chip experimental data. *Nucleic Acids Res*. **34,** D98–D103.
16. Suzuki, Y., Yamashita, R., Sugano, S., and Nakai, K. (2004) DBTSS, DataBase of Transcriptional Start Sites: progress report 2004. *Nucleic Acids Res.* **32(Database issue),** D78–D81.
17. Zhao, F., Xuan, Z., Liu, L., and Zhang, M. Q. (2005) TRED: a Transcriptional Regulatory Element Database and a platform for in silico gene regulation studies. *Nucleic Acids Res*. **33,** D103–D107.
18. Schmid, C. D., Praz, V., Delorenzi, M., Perier, R., and Bucher, P. (2004) The Eukaryotic Promoter Database EPD: the impact of in silico primer extension. *Nucleic Acids Res*. **32(Database issue),** D82–D85.
19. Kel, A. E., Gossling, E., Reuter, I., Cheremushkin, E., Kel-Margoulis, O. V., and Wingender, E. (2003) MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res*. **31,** 3576–3579.
20. Tompa, M., Li, N., Bailey, T. L., et al. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol*. **23,** 137–144.
21. Li, X., Zhong, S., and Wong, W. H. (2005) Reliable prediction of transcription factor binding sites by phylogenetic verification. *Proc. Natl. Acad. Sci. USA* **102,** 16,945–16,950.
22. Lenhard, B., Sandelin, A., Mendoza, L., Engstrom, P., Jareborg, N., and Wasserman, W. W. (2003) Identification of conserved regulatory elements by comparative genome analysis. *J. Biol*. **2,** 13.
23. Sinha, S., Blanchette, M., and Tompa, M. (2004) PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics* **5,** 170.
24. Jin, V. X., Leu, Y. W., Liyanarachchi, S., et al. (2004) Identifying estrogen receptor alpha target genes using integrated computational genomics and chromatin immunoprecipitation microarray. *Nucleic Acids Res*. **32,** 6627–6635.
25. Siddharthan, R., Siggia, E. D., and van Nimwegen, E. (2005) PhyloGibbs: a gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput. Biol*. **1,** E67.
26. Smith, A. D., Sumazin, P., Xuan, Z., and Zhang, M. Q. (2006) DNA motifs in human and mouse proximal promoters predict tissue-specific expression. *Proc. Natl. Acad. Sci. USA* **103,** 6275–6280.

27. Blanchette, M., Bataille, A. R., Chen, X., Poitras, C., et al. (2006) Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. *Genome Res.*
28. Jin, V. X., Singer, G. A., Agosto-Perez, F. J., Liyanarachchi, S., and Davuluri, R. V. (2006) Genome-wide analysis of core promoter elements from conserved human and mouse orthologous pairs. *BMC Bioinformatics* **7,** 114.
29. van Steensel, B. (2005) Mapping of genetic and epigenetic regulatory networks using microarrays. *Nat. Genet.* **37(Suppl),** S18–S24.
30. Heisler, L. E., Torti, D., Boutros, P. C., et al. (2005) CpG Island microarray probe sequences derived from a physical library are representative of CpG Islands annotated on the human genome. *Nucleic Acids Res.* **33,** 2952–2961.
31. Mao, D. Y., Watson, J. D., Yan, P. S., et al. (2003) Analysis of Myc bound loci identified by CpG island arrays shows that Max is essential for Myc-dependent repression. *Curr. Biol.* **13,** 882–886.
32. Yan, P. S., Shi, H., Rahmatpanah, F., et al. (2003) Differential distribution of DNA methylation within the RASSF1A CpG island in breast cancer. *Cancer Res.* **63,** 6178–6186.
33. Wells, J., Yan, P. S., Cechvala, M., Huang, T., and Farnham, P. J. (2003) Identification of novel pRb binding sites using CpG microarrays suggests that E2F recruits pRb to specific genomic sites during S phase. *Oncogene* **22,** 1445–1460.
34. Weinmann, A. S., Yan, P. S., Oberley, M. J., Huang, T. H., and Farnham, P. J. (2002) Isolating human transcription factor targets by coupling chromatin immunoprecipitation and CpG island microarray analysis. *Genes Dev.* **16,** 235–244.
35. Odom, D. T., Zizlsperger, N., Gordon, D. B., et al. (2004) Control of pancreas and liver gene expression by HNF transcription factors. *Science* **303,** 1378–1381.
36. Day, W. H. and McMorris, F. R. (1993) A consensus program for molecular sequences. *Comput. Appl. Biosci.* **9,** 653–656.
37. Stormo, G. D. (2000) DNA binding sites: representation and discovery. *Bioinformatics* **16,** 16–23.
38. Benos, P. V., Lapedes, A. S., and Stormo, G. D. (2002) Is there a code for protein-DNA recognition? Probab(ilistical)ly. *Bioessays* **24,** 466–475.
39. Ben-Gal, I., Shani, A., Gohr, A., et al. (2005) Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics* **21,** 2657–2666.
40. Zhou, Q. and Liu, J. S. (2004) Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics* **20,** 909–916.
41. Audic, S. and Claverie, J. M. (1997) Detection of eukaryotic promoters using Markov transition matrices. *Comput. Chem.* **21,** 223–227.
42. Fang, F. and Blanchette, M. (2006) FootPrinter3: phylogenetic footprinting in partially alignable sequences. *Nucleic Acids Res.* **34,** W617–W620.
43. Lardenois, A., Chalmel, F., Bianchetti, L., Sahel, J. A., Leveillard, T., and Poch, O. (2006) PromAn: an integrated knowledge-based web server dedicated to promoter analysis. *Nucleic Acids Res.* **34,** W578–W583.

44. Berezikov, E., Guryev, V., and Cuppen, E. (2005) CONREAL web server: identification and visualization of conserved transcription factor binding sites. *Nucleic Acids Res*. **33,** W447–W450.
45. Corcoran, D. L., Feingold, E., and Benos, P. V. (2005) FOOTER: a web tool for finding mammalian DNA regulatory regions using phylogenetic footprinting. *Nucleic Acids Res*. **33,** W442–W446.
46. Das, D., Banerjee, N., and Zhang, M. Q. (2004) Interacting models of cooperative gene regulation. *Proc. Natl. Acad. Sci. USA* **101,** 16,234–16,239.
47. MacIsaac, K. D. and Fraenkel, E. (2006) Practical strategies for discovering regulatory DNA sequence motifs. *PLoS Comput. Biol*. **2,** E36.
48. Vavouri, T. and Elgar, G. (2005) Prediction of cis-regulatory elements using binding site matrices—the successes, the failures and the reasons for both. *Curr. Opin. Genet. Dev*. **15,** 395–402.
49. Bulyk, M. L. (2003) Computational prediction of transcription-factor binding site locations. *Genome Biol*. **5,** 201.
50. Cartharius, K., Frech, K., Grote, K., et al. (2005) MatInspector and beyond: promoter analysis based on transcription factor binding sites. *Bioinformatics* **21,** 2933–2942.
51. Chen, Q. K., Hertz, G. Z., and Stormo, G. D. (1995) MATRIX SEARCH 1.0: a computer program that scans DNA sequences for transcriptional elements using a database of weight matrices. *Comput. Appl. Biosci*. **11,** 563–566.
52. Wasserman, W. W. and Sandelin, A. (2004) Applied bioinformatics for the identification of regulatory elements. *Nat. Rev. Genet*. **5,** 276–287.
53. Suzuki, Y., Yamashita, R., Shirota, M., et al. (2004) Sequence comparison of human and mouse genes reveals a homologous block structure in the promoter regions. *Genome Res*. **14,** 1711–1718.
54. Iwama, H. and Gojobori, T. (2004) Highly conserved upstream sequences for transcription factor genes and implications for the regulatory network. *Proc. Natl. Acad. Sci. USA* **101,** 17,156–17,161.
55. Wasserman, W. W., Palumbo, M., Thompson, W., Fickett, J. W., and Lawrence, C. E. (2000) Human-mouse genome comparisons to locate regulatory sites. *Nat. Genet*. **26,** 225–228.
56. Liu, Y., Liu, X. S., Wei, L., Altman, R. B., and Batzoglou, S. (2004) Eukaryotic regulatory element conservation analysis and identification using comparative genomics. *Genome Res*. **14,** 451–458.
57. Loots, G. G. and Ovcharenko, I. (2004) rVISTA 2.0: evolutionary analysis of transcription factor binding sites. *Nucleic Acids Res*. **32,** W217–W221.
58. Sandelin, A., Wasserman, W. W., and Lenhard, B. (2004) ConSite: web-based prediction of regulatory elements using cross-species comparison. *Nucleic Acids Res.* **32,** W249–W252.
59. Blanchette, M. and Tompa, M. (2003) FootPrinter: A program designed for phylogenetic footprinting. *Nucleic Acids Res*. **31,** 3840–3842.
60. Blanco, E., Messeguer, X., Smith, T. F., and Guigo, R. (2006) Transcription factor map alignment of promoter regions. *PLoS Comput. Biol*. **2,** E49.

61. Palaniswamy, S. K., Jin, V. X., Sun, H., and Davuluri, R. V. (2005) OMGProm: a database of orthologous mammalian gene promoters. *Bioinformatics* **21,** 835–836.
62. Linhart, C., Elkon, R., Shiloh, Y., and Shamir, R. (2005) Deciphering transcriptional regulatory elements that encode specific cell cycle phasing by comparative genomics analysis. *Cell Cycle* **4,** 1788–1797.
63. Sinha, S., van Nimwegen, E., and Siggia, E. D. (2003) A probabilistic method to detect regulatory modules. *Bioinformatics* **19(Suppl 1),** I292–I301.
64. Breiman, L., Friedman, J., Stone, C. J. and Olshen, R. A. (1984) *Classification and regression trees*. CRC Press LLC, Boca Raton, FL.
65. Breiman, L. (2001) Random Forests. *Macine Learing* **45,** 5–32.
66. Steinberg, D. and Colla, P. L., (1995) *CART: Tree–Structured Nonparametric Data Analysis,* San Diego, CA: Salford systems.
67. Schmid, C. D., Perier, R., Praz, V., and Bucher, P. (2006) EPD in its twentieth year: towards complete promoter coverage of selected model organisms. *Nucleic Acids Res*. **34,** D82–D85.
68. Hinrichs, A. S., Karolchik, D., Baertsch, R., et al. (2006) The UCSC Genome Browser Database: update 2006. *Nucleic Acids Res*. **34,** D590–D598.

9

Mining Biomedical Data Using MetaMap Transfer (MMTx) and the Unified Medical Language System (UMLS)

John D. Osborne, Simon Lin, Lihua (Julie) Zhu, and Warren A. Kibbe

Summary

Detailed instruction is described for mapping unstructured, free text data into common biomedical concepts (drugs, diseases, anatomy, and so on) found in the Unified Medical Language System using MetaMap Transfer (MMTx). MMTx can be used in applications including mining and inferring relationship between concepts in MEDLINE publications by transforming free text into computable concepts. MMTx is in general not designed to be an end-user program; therefore, a simple analysis is described using MMTx for users without any programming knowledge. In addition, two Java template files are provided for automated processing of the output from MMTx and users can adopt this with minimum Java program experience.

Key Words: Analysis; biomedical; data mining; MMTx; NLP; parsing; UMLS.

1. Introduction

The explosion of biomedical information in electronic format has posed both opportunities and challenges to researchers wishing to analyze that information. The easy accessibility and size of the information allow a wide range of hypotheses to be tested and questions to be asked, but much of the data is in free text format and consequently difficult to organize and compare. The field of natural language processing has a variety of tools to deal with these types of problems, one of which (MetaMap Transfer [MMTx]) ***(1)*** is of particular interest to biomedical researchers. MMTx is one of the tools used by the National Library of Medicine (NLM) to import medical and biological vocabularies into the Unified Medical Language System (UMLS) database. A total of 143 vocabularies

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

are included in the 2006 release of UMLS including widely used vocabularies such as SNOMED™ International Statistical Classification of Diseases and Related Health Problems (ICD9), Medical Subject Headings (MeSH), the Formal Model of Anatomy, and many others. The NLM is making an effort to cover as widely as possible the biomedical domain, so in addition to the standard medical vocabularies, additional vocabularies covering drug codes, chemicals, adverse reactions, and nursing care standards are also included. In general, the coverage is large enough that most researchers should be able to find most commonly needed systems and concepts needed to map free text for their problem domain. A detailed understanding of MMTx is not required to use it, but it helps to understand the process in order to get the best results possible. A more detailed and extensive description can be found from the documents page (http://mmtx. nlm.nih.gov/ docs.shtml), but the salient points are summarized herein. **Figure 1** outlines the steps taken by MMTx as it maps components of free text to candidate concepts.

First the tokenization module organizes the input document into sections consisting of sets of sentences and tokens. This tokenizer will recognize the MEDLINE format (available for PubMed articles through NCBI) or free text automatically, so in most cases the users will need to do little, if any formatting of input data before running MMTx. The Part of Speech Tagger Client ***(2)*** then "tags" the tokens in order to identify which part of speech (such as a noun) the tokens belong to. These tagged tokens are then subject to LexicalLookup, the module that determines if any of the tagged elements belong to a particular lexicon. Adjacent tokens that are part of the same lexicon (for instance "July" and "5th") can then be treated as a single lexical element. A noun phrase parser then identifies noun phrases from these elements for which variants are calculated by table lookup. These variants are then used to identify matching strings from the UMLS Metathesaurus termed candidates. Each of the candidates is evaluated and assigned a score based on the extent of contiguity, central component involvement, cohesiveness, word order, and other factors. The final mapping module generates a list of UMLS Metathesaurus concepts that best cover the input noun phrase and associated scores representing the mapping result for input text.

The number of applications for a tool like MMTx is enormous. It can be data mining and inferring relationship between concepts in MEDLINE publications and other published data and is in general appropriate for any task that requires the transformation of free text biomedical data into categorized, comparable biomedical information. Published examples include extracting information about medical problems from clinical reports ***(3)***, detecting respiratory illness in patients from emergency department reports ***(4)***, and annotating enzyme classes with disease-related information ***(5)***. Although, not always easy to

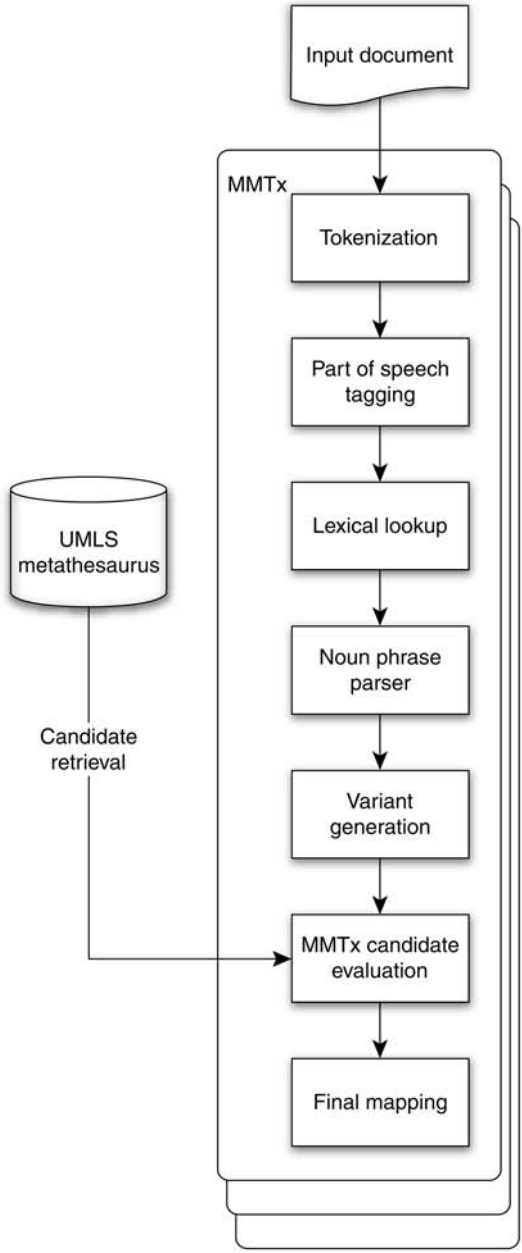


Fig. 1. Shows the breakdown of a document inputted into MMTx as it proceeds through the pipeline.

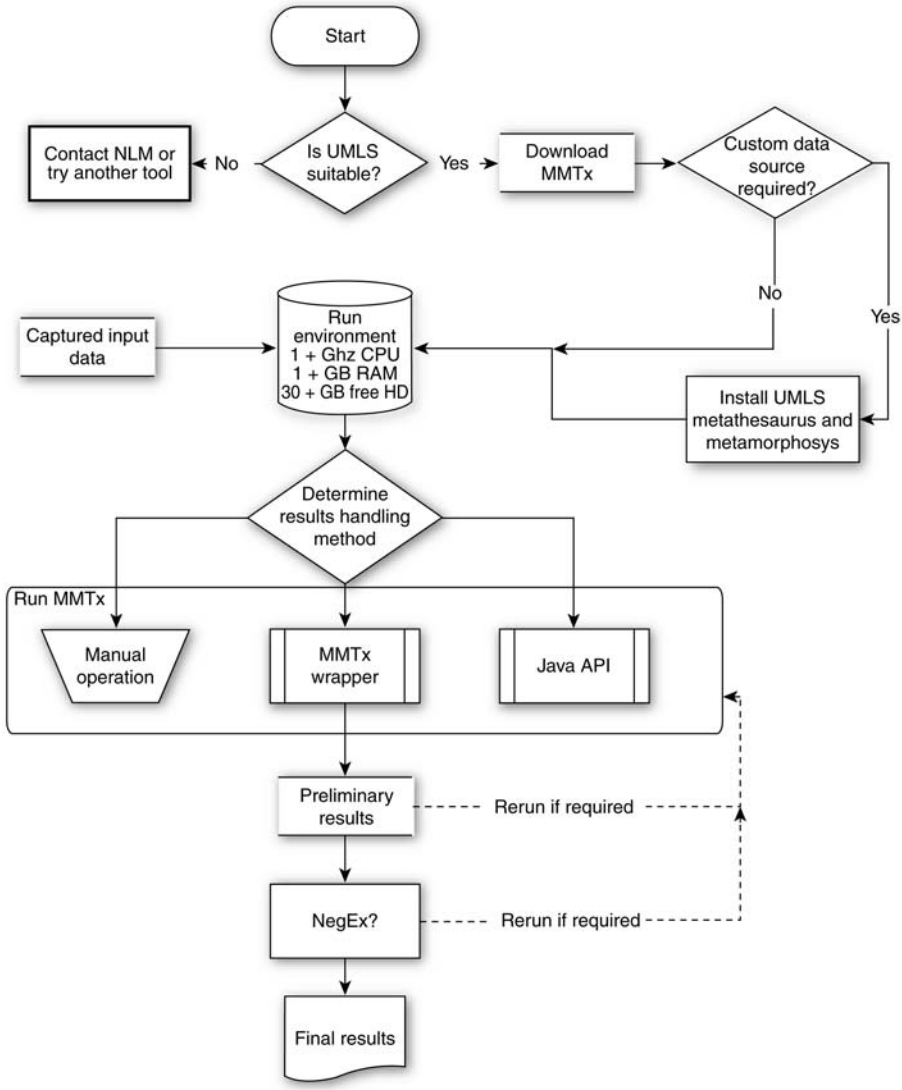


Fig. 2. Flowchart of typical MMTx workflow.

use, it is powerful enough that results can be analyzed with nothing more complex than a spreadsheet program. For demonstration purposes free text geneRIF data (found for most genes indexed on NCBI) will be mapped to the UMLS disease/disorder semantic network to infer relationships for testicular cancer. A flowchart showing the general usage of MMTx is shown in **Fig. 2**.

2. Materials

Data mining with MMTx will require a host machine with a least 1 GHz CPU speed, 1 GB of RAM, and at least 3–4 GB of hard disk space for most use cases. If the user is going to use custom data sets (*see* **Subheading 3.1.**), then the UMLS Metathesaurus and Metamorphosys will also need to be installed, raising the hard disk requirements to approx 40 GB.

UMLS Metamorphosys and MMTx are both Java-based programs and require Java virtual machine (JVM) to run. They have been tested with JVMs for Windows (XP, 2000, and NT), Linux, Solaris (8 and 9), or Macintosh OS X 10.3 or higher. Up-to-date requirements for the installation of the UMLS Metathesaurus can be found in the README.txt file distributed with each distribution of UMLS. The ideal running environment for ease of setup is probably one of the non-Windows systems with Java already installed. The command line examples are designed for a UNIX-like system. However, if NegEx (a program which detects negations of concepts in Text Mining) is going to be used, then a Windows system is required. Owing to their large size, obtaining UMLS data and programs is easiest with a fast Internet connection, otherwise UMLS can be ordered in DVD format. No Internet connection is required for Metamorphosys or MMTx while running. Users should also have their own data set in electronic format on the same machine on which UMLS is installed. UMLS does not need to be put into a relational database to be used, but if this is desired, then the host machine should have either mySQL or Oracle installed.

3. Methods

3.1. Determining the Suitability of UMLS for Input Data Set

Broadly speaking the UMLS is organized by vocabulary, by semantic type, and by individual atomic UMLS concepts. Vocabularies in UMLS are an organized set of concepts and relationships. Semantic types span vocabularies in UMLS and were created to categorize all concepts represented in the UMLS. It includes general categories such as “drugs” and “congenital abnormalities” that are commonly found in UMLS vocabularies. At the lowest level the UMLS has concepts, which describe the narrowest entity such as a particular drug or a specific disease. The first step in determining UMLS suitability is to determine what in UMLS terms one wants to match the input data with. First-time users will likely use MMTx to search against one of the preconfigured data sets of UMLS and then filter their matches against a particular vocabulary, semantic type or small concept set, to get the results needed. UMLS is preconfigured to make available most of its English language vocabulary sources. The only exceptions are listed in **Table 1** and are due to licensing restrictions set in place by the American Medical Association (AMA).

UMLS source abbreviation	Full source name
CDT5	Current Dental Terminology 2005 (CDT-5)
CPT01SP	Physicians' Current Procedural Terminology, Spanish Translation, 2001
CPT2005	Physicians' Current Procedural Terminology, 2005
HCDT5	HCPCS Version of Current Dental Terminology 2005 (CDT-5)
HCPCS05	Healthcare Common Procedure Coding System, 2005
HCPT05	HCPCS Version of Current Procedural Terminology (CPT), 2005
MTHCH05	Metathesaurus CPT Hierarchical Terms,2005
MTHHH05	Metathesaurus HCPCS Hierarchical Terms,2005

Table 1
The Input Options for Data Sent to UMLS

Short name	Long name	Default value	Purpose
_	--medlineCitations	False	The input is a collection of mediline citations
_	--mrcon	False	The inputis a collection of MRCON rows
_	--free Text	True	The input is free text
_	--fielded Text=	False	Is the input file/stdin fielded text?
_	textField=	2	For fieded text, which field containsthe text
_	--fieldedSeparator	1	For fielded text, which char is separator

3.1.1. Filtering Against a Specific Vocabulary

Filtering against a particular vocabulary set is advisable in cases wherein the user has a very specific need to map to that vocabulary, such as mapping free text from clinical reports to a medical coding system such as SNOMED-CT. Unless there is a compelling reason (such as a license restriction), it is preferable to use the entire Metathesaurus to give the best possible coverage. If a user is interested in a particular vocabulary, the quickest way to check for its presence is to check: (http://www.nlm.nih.gov/research/umls/sources_by_categories.html), which lists a relatively current list of English vocabularies in UMLS.

3.1.2. Filtering by Semantic Type

It is worthwhile to take advantage of the classification of concepts in the Metathesaurus into semantic types when the data to be extracted falls into large categories and there is no explicit requirement to map to a particular vocabulary. Semantic types (such as deformities, embryonic structures, chemicals, and so on) allow the data miner to capture data categories across vocabularies without knowing the details of either the vocabulary or the various concepts in UMLS. A webpage listing semantic types available in the UMLS can be found at http://mmtx.nlm.nih.gov/semanticTypes.shtml, and the definitive listing of semantic types for a particular distribution of UMLS can be found in nls/mmtx/data/2004/mmtx/semdef. A typical use case may be to mine MEDLINE for statistical associations between semantic types or concepts.

3.1.3. Filtering by Specific Concepts

In this case, input data is filtered by a small group of manually selected concepts that span one or more semantic types and vocabularies. One such use case that MMTx and UMLS have already been used for is mapping clinical reports to a manually selected set of 80 particular medical problems ***(6)***. If it is not known whether the UMLS Metathesaurus contains the specification of the desired concept/s, the UMLS Knowledge Source Server (UMLSKS) can be used to search for the concepts in UMLS as shown in **Fig. 3**. The UMLSKS is located at http://umlsks.nlm.nih.gov/kss/servlet/Turbine/template/admin,user, KSS_login.vm and its use requires registration and signing of the UMLS license agreement **(Note 1)**.

3.1.4. UMLS Lacks the Required Content

In cases wherein UMLS does not contain the required vocabulary, semantic network, or concepts, it is best to directly contact the UMLS development team. The team is generally receptive to reasonable requests, although the inclusion of any new vocabulary will have to be justified against the NLM mandate and will take a considerable amount of time. Semantic types and concepts can be added

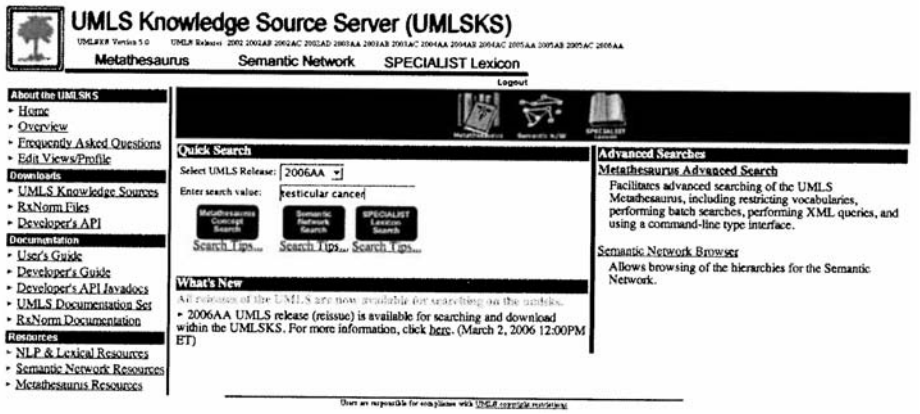


Fig. 3. Querying the UMLSKS. The UMLSKS can also be installed locally.

with less difficulty. The webpage for the UMLS users groups is located at https://list.nih.gov/archives/umlsusers-l.html.

3.2. Using a Custom Data Source

MMTx has three different models for candidate retrieval from Metathesaurus that differ in extent of filtering applied. However, most users should be satisfied using these models, in some cases a user may need to create a custom data set. A typical use case might be for commercial applications using the Metathesaurus that must exclude particular vocabularies owing to licensing restrictions. Using a custom data will require that both the UMLS Metathesaurus and Metamorphosys be downloaded and installed.

It should be kept in mind that a custom data set is not required to restrict results to appropriate concepts. Whereas creating a custom data source does give the user the ability to prevent matches to unsuitable concepts, a postmatch filtering can still be done to eliminate unwanted concepts from particular vocabularies or semantic types. A custom data source is generally only warranted to avoid licensing problems or if the information to be extracted from the input data cannot easily be described by the UMLS semantic network or particular vocabularies. Because of the release of MMTx 2.4.B, candidate concepts can be filtered by both semantic type and by vocabulary from command line of MMTx, making custom data sets less critical. It is advised against creating a custom data set if the concepts of interest are small in number, they can be described by particular vocabularies, or if they are restricted to certain semantic network types. The custom data set can always be created later. The creation of a custom data set will not be covered herein, but for those who are interested in the details on custom data sets creation, the information can be found at http://mmtx.nlm.nih.gov/DataFileBuilder.pdf.

3.3. Downloading

Downloading MMTx requires a login to the UMLSKS, which in turn requires the acceptance of a license agreement (*see* **Note 1**). The site for download is http://mmtx.nlm.nih.gov/Download/download.shtml, which is password protected.

3.3.1. Downloading of MMTx

Although MMTx is significantly smaller than the Metathesaurus, it is still almost 2.5 GB in size when including the standard data sources. Therefore, the NLM recommendation to use a fast Internet connection (and not a 802.11g wireless connection) should be taken seriously. The current version of MMTx as of July 2006 is 2.4 MB. Both the smaller executable file (mmtx_V2.4.B_exec.jar) and the larger data file (mmtx_V2.4.B_data.jar) should be downloaded and placed in the same directory. The data file contains only the strict model of the Metathesaurus.

3.3.2. Downloading Metathesaurus and Metamorphosys (Optional)

Users interested in using MMTx with a custom data source or curious about the "guts" of UMLS should install the Metathesaurus and Metamorphosys on a local system. Download this by selecting "UMLS Knowledge Sources" from the download section of the UMLSKS (*see* **Fig. 3**). All the files for a release including the data files (the .nlm files), the checksum files (MD5 and CHK), and the installation program Metamorphosys (mmsys.zip) should be downloaded into a single directory to a partition with at least 23 GB of space, preferably 30 or more. In general, only a single release (ex., 2006AA) needs to be downloaded, unless there is interest in changes in UMLS or its component vocabularies over time. In most cases this is unlikely to be true.

3.4. Installation

The installation of both the MMTx and the UMLS requires Java to extract the jar files or to run Metamorphosys to set install the Metathesaurus. The latest release of MMTx has been tested with Java 1.4 and some trouble was encountered running it with Java 1.5. It would be recommended that "java-version" be run on the command line to confirm that there is a compatible JVM before continuing. The JVM from Sun, Mac, and Blackdown are all known to work with UMLS. Installation of Java is beyond the scope of this chapter, but details can be found on the Sun website (http://www.java.com/en/download/manual.jsp).

3.4.1. Installation of UMLS—MMTx Tool

Installation of UMLS is also relatively straightforward and full instructions can be found at http://mmtx.nlm.nih.gov/install.shtml. For some notes on the installation process, *see* **Note 2**.

3.4.2. Installation of UMLS Metathesaurus and Metamorphosys (Optional)

Although not required unless a user is planning to create a custom data set, it is recommended to install it anyway. Running Metamorphosys and seeing the list of vocabularies in UMLS will give one a sense of the scale of UMLS and potentially pointers to useful areas that might otherwise go undiscovered. Installation is relatively straightforward, for complete and up-to-date information *see* http://www.nlm.nih.gov/research/umls/meta6.html where the latest requirements and instruction are detailed. There are only a couple of potential problems with installation. First, it is important that the checksum files be downloaded into the same directory as the other files because Metamorphosys will need to confirm that the downloaded files are intact. Second, one of the trickier aspects of Metamorphosys is that it is not always clear which vocabularies are being included and which are being excluded during source selection. Sources that are highlighted in blue are selected for exclusion by default, not for inclusion. This is somewhat counterintuitive and because installation of UMLS can take over an hour on some platforms, the radio buttons over the main menu should be checked to make sure the correct subset is selected. It is also worthwhile to save one's configuration file because the selection of the subset of UMLS to install tends to be fluid. It is often the case that the configuration chosen is not quite optimal (a source for inclusion or exclusion is often overlooked) so save the configuration to a file before beginning the subset process. Finally, the option to write the UMLS Metathesaurus as a database SQL script (either Oracle or mySQL) is turned off, so if one wants to put UMLS into a relational database, then these options need to be selected before subsetting.

3.5. Setup the Running Environment

If not already present, copy all the input data to the host machine on which MMTx has been installed. Everything can be run from a single directory.

3.5.1. Formatting Input Data

This step should be done if the user is planning running MMTx from the command line. If one is going to use the JAVA Application Programming Interface (Java API) to run and process MMTx (not recommended for non-programmer) then this input transformation step can be skipped because the programmer (not MMTx) will be responsible for parsing the input data.

MMTx is extremely flexible in terms of the format of data it can accept. It will take in any free text, and MEDLINE data can be parsed directly by MMTx and so can particular fields in a text file delimited by arbitrary separators. For a listing on how to handle the data type, *see* **Table 1**. A user should be able to

format input data using any commonly available spreadsheet program (such as Excel), provided the spreadsheet software can support all the lines of input data that are required to be read in. For example, the geneRIF database is being processed, an example of the format of being:

139 2827859 15501399 2005-05-14 12:17 T-cell recognition of the outer-surface protein A (OspA) epitope is important in the induction of autoimmunity in treatment-resistant Lyme arthritis (OspA).

The tabs can be substituted with "|"s in any text editor that supports find and replace type operations so that MMTx will have an easier time with the data (*see* **Note 3**). In Microsoft Word the data could be transformed as follows:

1. Open the database file (generif_basic) in Word.
2. Use Ctrl-C to highlight and copy a single tab character.
3. Select Edit → Find.
4. Select the Replace tab.
5. Select the "Find what" textbox and type Ctrl-V to paste in a tab.
6. Select the "Replace with" textbox type "|" without quotes.
7. Select "Replace All."
8. Save the newly formatted database.

This will result in text in the default MMTx format as shown next. 139| 2827859| 15501399|2005-05-14 12:17|T-cell recognition of the OspA epitope is important in the induction of autoimmunity in treatment-resistant Lyme arthritis (OspA).

3.6. Running and Handling Results From MMTx

Perhaps the most difficult component of data mining with MMTx is handling the overwhelming amount of data that will be generated from the original input data. This problem is complicated by the fact that MMTx is not really designed to be an end-user program. MMTx is focused more on the production of machine-readable data for analysis by software tools, and not for direct interruption by an end-user. Potential users must therefore overcome a fairly work-intensive initial barrier before they can assess the utility of MMTx and UMLS. The simple example herein should avoid some of this but the focus on the generation of machine-readable data means that the best way to handle MMTx generated data is programmatically, by either handling the output of MMTx directly or preferably through the Java API. Regardless of whether software tools are used to process and analyze the results, ultimately a human is needed for the final analysis.

3.6.1. Choosing a Data Model

Regardless of whether the analysis will be software assisted, one consideration remains the same—choosing a data model. As discussed in **Subheading 3.3.**, there are three different data models. The default "strict" model utilizes the highest

amount of filtering and is useful when accuracy is desired; the "moderate" model is similar to the strict model but lacks syntactic filtering and is best suited to evaluate input text as a whole rather than as discrete phrases. Finally, the relaxed model provides minimal filtering of its component strings and is best used for exploring. In cases wherein accuracy is important, select a strict data model. When doing exploratory fishing for associations, it is suggested to start with a relaxed data model and move to a moderate data model if the results are undesirable. If one is going to use a different data model one will have to download it from the download page mentioned in the **Subheading 3.3.1.**

3.6.2. The Perils of Filtering With MMTx

Filtering of data is an option with MMTx, but often it can cause more problems than it solves. However, filtering may remove undesirable matches, it also hides the fact that such matches occurred. High-scoring matches may get past one's filter, but one will not know how to remove them if one has no ranking information. By fully mapping the text it is possible to programmatically remove high-scoring but low-ranking matches that are counterproductive to the data mining at hand. For the same reason it is cautioned against removing sources from contention unless it is for licensing purposes or the source causes more problems than it creates for one's mappings.

3.6.3. Running From the Command Line (for Nonprogrammers)

The details of running MMTx are found at http://mmtx.nlm.nih.gov/runMMTx.shtml and two examples are provided with MMTx usage details shown on a separate webpage at http://mmtx.nlm.nih.gov/semanticTypes.shtml.

The actual text is the fifth column, which is of interest in mapping. The other columns could be eliminated by using a spreadsheet or the UNIX "cut" option, but in this case one can use the MMTx input parameters to handle fielded text. One will also select the option "show_cuis" which is turned off by default. This allows to actually determine if one has mappings to the concepts of interest without having to manually investigate the text. One can also turn off the candidates and mappings (-c=false and -m=false) to reduce the amount of output and use the sections option to specify the entire line from which geneRIF was derived. The new version of MMTx (MMTx 2.4B) takes a semantic type as an argument (—restrict_to_sts=neop) not yet specified in the documentation. By specifying the abbreviation for neoplastic process one can restrict one's results appropriately. The actual command to run MMTx appears as follows:

MMTx —fieldedText —textField=5 —fieldSeparator='|' —fileName=generifs_basic —show_cuis -c=false -m=false –-sections —restrict_to_sts=neop > outputfile.txt

In this command the input file is specified (—fileName=generifs_basic) in fieldText format (—fieldedText), separated by the "|" character (—field

Separator='|') and the text of interest is found in the fifth field (—textField=5). The concept unique identifiers are displayed in UMLS (—show_cuis), which can come in handy when mapping results across vocabularies. The list of candidates for a map are not displayed (-c=false) nor are the intermediate mappings (-m=false) in order to reduce the volume of output directed to outputfile.txt.

The abbreviations for semantic types are found in (http://mmtx.nlm.nih.gov/semanticTypes.shtml), it is not possible to use the numerical or full-length format when specifying the semantic type. One of the points to keep in mind when running MMTx is that it is CPU bound and usually has a relatively large running time. So it is worthwhile to examine the early results of the run by looking at the outputfile.txt to ensure that the results being achieved are useful. If machine processing is desired at a later point, the –f (fielded output) or –q (machine output) options can be used. However, neither option has any flexibility in customizing the output; they do not include the concept unique identifier (CUI) for the actual mapping result and so might be of limited utility.

3.6.4. Java API (Java Programmers Only)

Using the Java API is the optimal way of handling MMTx. With a little bit of work the processing of the input data can be precisely controlled, which includes using any other metadata from the data source at processing time in evaluating mapping candidates. It also allows for an exact specification of the output format for easy analysis. A description of the API can be found http://mmtx.nlm.nih.gov/MMTxAPI_V2.3.pdf. Below is a template for constructing the Java API to process geneRIFs, consisting of two separate source files. These are also available on the web at: http://download.bioinformatics.northwestern.edu/download/mmtx/mmtx_java_example_template.tar.gz

The first file (MMTxGeneRIF.java) is a subclass of MMTxAPILite that will handle the processing. It is in this file in which one's evaluation of the input phrases should occur, because at this point one will have access scoring results in MMTx plus access to any additional metadata in the input phrase one wants to make use of in evaluating the candidate. The text in bold, "Undesired phrase here" can be replaced with whatever is appropriate to remove undesired mappings. The last lines of this file prints out a candidate CUI, mapped phrase, and score. Adding in an empirical derived cutoff score to remove bad mappings can further reduce candidates and the output format can also be adjusted herein.

```
MMTxGeneRIF.java
import java.util.*;
import gov.nih.nlm.nls.nlp.textfeatures.*; // -Included in MMTx.jar
import gov.nih.nlm.nls.mmtx.MMTxAPILite;
```

```
public class MMTxGeneRIF extends MMTxAPILite{
  private static ArrayList _genediseasemaps=new ArrayList(); //Our data
  structure
  private static GeneRIF _currentRIF = null;
  public MMTxGeneRIF(String[] args) { super.init(args); }
  public MMTxGeneRIF(){};
  public void processMappings(Sentence pSentence) throws Exception {
    Vector phrases = pSentence.getPhrases();
    for(Iterator g=phrases.iterator();g.hasNext();){
      Phrase aPhrase = (Phrase) g.next();
      if(aPhrase.getOriginalString().indexOf("Undesired phrase here"))
      continue;
        ArrayList topMappings=aPhrase.getBestFinalMappings();
        if(topMappings !=null ) {
          for(Iterator q=topMappings.iterator();q.hasNext();){
            FinalMapping topMapping = (FinalMapping)q.next();
            ArrayList cuis = topMapping.getCandidates();
              for(Iterator l = cuis.iterator(); l.hasNext();){
                Candidate c = (Candidate)l.next();
                System.out.println(c.getCUI()+"|"+
                aPhrase.getOriginalString()+
                "|"+c.getCandidateScore());
                                    }  }  }  }  }  }
```

The other program (MapClient.java) is responsible only for parsing the input data and calling the MMTxGeneRIF process document function. Users will have to find or write their own parser for whatever data they are inputting.

MapClient.java

```
import edu.northwestern.bcore.dotools.SetupSingleton;
import edu.northwestern.bcore.dotools.UMLSFileTools;
import java.util.*;

public class MapClient {
  public static void main(String[] args){
    MMTxGeneRIF myMMTx = null;
    try {
      myMMTx= new MMTxGeneRIF();
      Hashtable inputphrases = parseFile(args[0]);
      //Write out our document to process
      for(Iterator it = inputphrases.keySet().iterator();it.hasNext();){
```

```
            String input = (String)it.next();
            ArrayList al = (ArrayList)inputphrases.get(input);
            for(Iterator it2=al.iterator();it2.hasNext();){
              synchronized(rif) { myMMTx.processDocument(input);}
            }
          }
          myMMTx.cleanup();
        } catch (Exception e) {//Error handling code here}
      }
    }
```

3.6.5. MMTx Wrapper (Non-Java Programmer Option)

This is similar to running MMTx on the command line, except MMTx is wrapped by an external program that processes the output as it is generated. This gives significantly more power than running MMTx on the command line and allows any language to be used as the processing tool. The disadvantage is the extra programming work involved. The options are too varied and will not be covered here.

3.7. Filtering and Reprocessing Preliminary Results (Command Line)

The output generated by MMTx can be quickly filtered on a UNIX system (*see* **Note 4**) by the following command:

```
cat outputfile.txt | grep -P 'C0153594|C0855197|Section' > results.txt
```

This will leave only the original geneRIF input (the MMTx Section) and below it any hits for "Testicular malignant germ cell tumor" and "Malignant neoplasm of testis". A visual inspection at this point may reveal problems with the current filtering (**Figs. 4** and **5**). As instance calcium ions (Ca^{2+}) are parsed into a Ca token that is recognized as cancer. A geneRIF discussing calcium and the testes may be flagged as cancer accidentally. Using abbreviations like this can be turned off (the —no_acros_abbrs flag) but more may be lost than gained. It is up to the researcher to customize the filtering for the data set, it will be an interactive, learning process.

3.8. Analysis

Ultimately, at some point the value gained for handling exceptionally bad mapping cases will be less than the effort required to handle them. It is at this point that the user is done. An additional step a user may want to take is to run the data through NegEx ***(7)***, which detects negation expressions in text. For instance, the program outlined earlier will map all geneRIFs mentioning testicular data. This means that geneRIFs to the effect of "This gene is not involved in testicular cancer" will show an association. NegEx can detect and remove these.

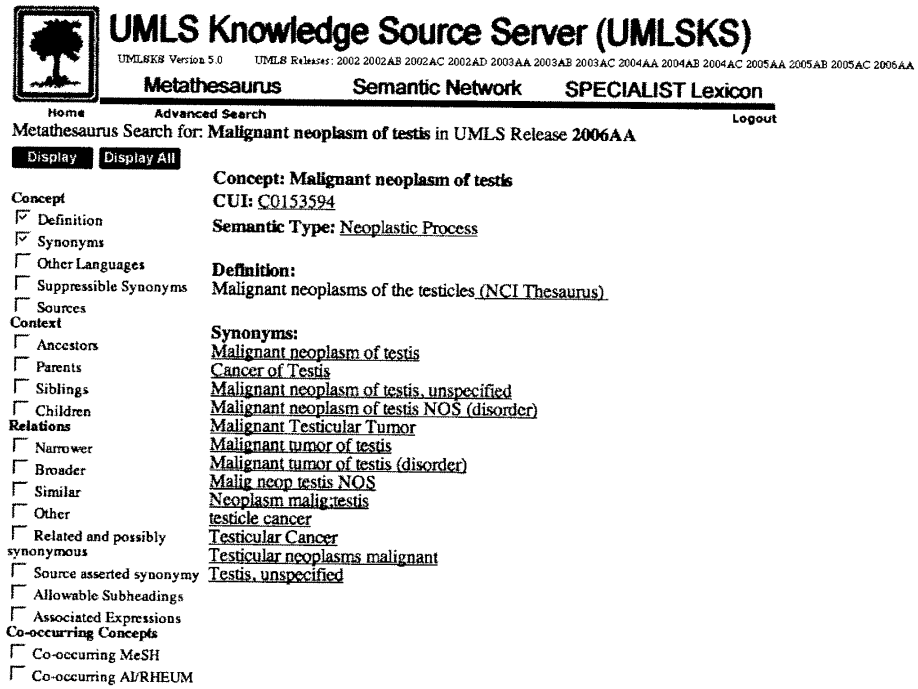


Fig. 4. Input testicular cancer will show two retrieved concepts, "malignant neoplasm of testes" and "testicular malignant germ cell tumor" one or both of which may be of interest to the researcher. The UMLS handles eponyms, synonyms including spelling variants (tumor/tumour) when selecting matching terms.

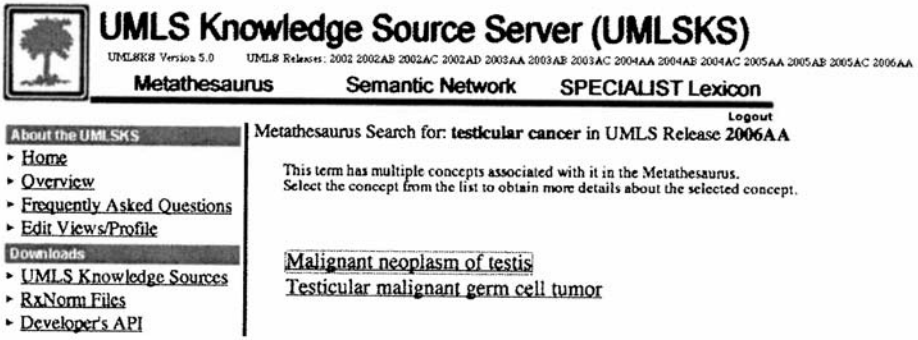


Fig. 5. The concept "Malignant neoplasm of testis" is detailed, including synonyms and a definition. The CUI uniquely identifies a more or less distinct concept in UMLS.

4. Notes

1. *Registration and License Information:* A complete description of the UMLS license agreement can be found here (http://www.nlm.nih.gov/research/umls/license.html). In general, one is free to use and incorporate UMLS as needed for research applications but care should be taken when working with any of the particular vocabularies in UMLS that are subject to their own licensing requirements. For instance the SNOMED vocabulary cannot be redistributed in commercial applications nor used for research outside the United States without a separate license agreement. The UMLS license also has a minimal reporting requirement; licensees are required to fill out a short electronic report once a year.
2. *Installation of MMTx:* After both the mmtx executable and the mmtx data file have been downloaded to the same directory, it is best to manually unjar both files. Problems have consistently been had with the installation script detecting and unjarring the data file.
3. *Command Line MMTx:* No success has been had in handling tab delimited data with MMTx, but "|" delimited data (the default) works just fine.
4. *Perl Expressions (-P) in grep:* Require the presence of GNU grep (*see* http://www.gnu.org/software/grep/). Systems without GNU grep may not support it.

References

1. Aronson, A. R. (2001) Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc. AMIA Symp.* 17–21.
2. Smith, L., Rindflesch, T., and Wilbur, W. J. (2004) MedPost: a part-of-speech tagger for bioMedical text. *Bioinformatics* **20(14),** 2320–2321.
3. Meystre, S. and Haug, P. J. (2005) Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation. *J. Biomed. Inform.* **5,** 5.
4. Chapman, W. W., Fiszman, M., Dowling, J. N., Chapman, B. E., and Rindflesch, T.C. (2004) Identifying respiratory findings in emergency department reports for biosurveillance using MetaMap. *Medinfo* **11(Pt 1),** 487–491.
5. Hofmann, O. and Schomburg, D. (2005) Concept-based annotation of enzyme classes. *Bioinformatics* **21(9),** 2059–2066.
6. Meystre, S. and Haug, P. J. (2005) Evaluation of Medical Problem Extraction from Electronic Clinical Documents Using MetaMap Transfer (MMTx). *Stud. Health Technol. Inform.* **116,** 823–828.
7. Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., and Buchanan, B. G. (2001) A simple algorithm for identifying negated findings and diseases in discharge summaries. *J. Biomed. Inform.* **34(5),** 301–310.

10

Statistical Methods for Identifying Differentially Expressed Gene Combinations

Yen-Yi Ho, Leslie Cope, Marcel Dettling, and Giovanni Parmigiani

Summary

Identification of coordinate gene expression changes across phenotypes or biological conditions is the basis of the ability to decode the role of gene expression regulatory networks. Statistically, the identification of these changes can be viewed as a search for groups (most typically pairs) of genes whose expression provides better phenotype discrimination when considered jointly than when considered individually. Such groups are defined as being jointly differentially expressed. In this chapter several approaches for identifying jointly differentially expressed groups of genes are reviewed of compared on a set of simulations.

Key Words: High-order interactions; liquid correlation; microarray data; entropy; joint differential expression; correlation.

1. Introduction

Gene-expression microarrays quantify the levels of thousands of RNA transcripts simultaneously ***(1)***. A common experimental design is the comparison of samples from different phenotypes or biological conditions, with the goal of identifying differences in expression. Standard analysis approaches are constructed considering each gene in turn and investigating the hypothesis that the one-dimensional (1D) gene-specific distributions are the same across conditions ***(2,3)***. In biological processes, RNA transcript levels interact with each other, and it is of interest to consider more than one gene at a time, to explore functional relationships between genes that are associated with phenotypes. Statistically, this means testing more general hypotheses formulated in terms of joint distributions of pairs or larger subgroups of genes ***(4)***.

The following artificial examples illustrate two archetypical cases of joint differential expression. **Figure 1** shows two genes with joint association on the

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

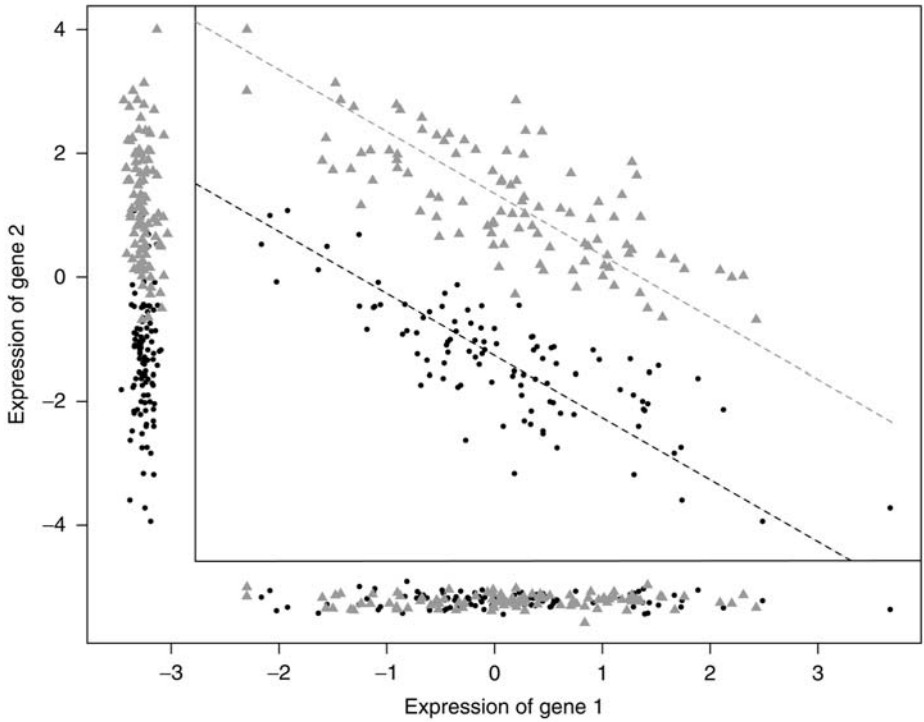


Fig. 1. Artificial example of joint differential expression (shift) of a pair of genes. The axes are the measured expression levels of the two genes. Grey and black circles represent samples from two phenotypes (say normal and cancer tissue). The inner panel reflects the joint distribution; the outer margins display the univariate marginal distributions. The dashed lines represent the first principal components, conditional on the phenotype. For this data $S_{\text{cross}} = 0.01$ and $S_{\text{shift}} = -0.29$.

phenotype: if the sum of their expression levels exceeds three, mostly the grey (▲) phenotype is observed. However, neither of the two genes alone shows a strong association with the phenotype. The sides (margins) of the figure show the 1D gene-specific distributions (technically referred to as marginal distributions). These would have been used in a one-gene-at-a-time testing approach, and thus both genes would have been unlikely to be selected. This pattern is generated by the combination of a relatively high correlation between the genes, and a shift in the sum of the expression levels (in this case owing mostly to gene 2) across phenotypes. Therefore, this will be referred to as a "shift" pattern. A biological mechanism leading to this pattern may occur when two genes are substitutes in a molecular process that is closely linked to the phenotype. A complementary case occurs when two genes cluster around two positively sloped lines, with a shift in their difference across phenotypes.

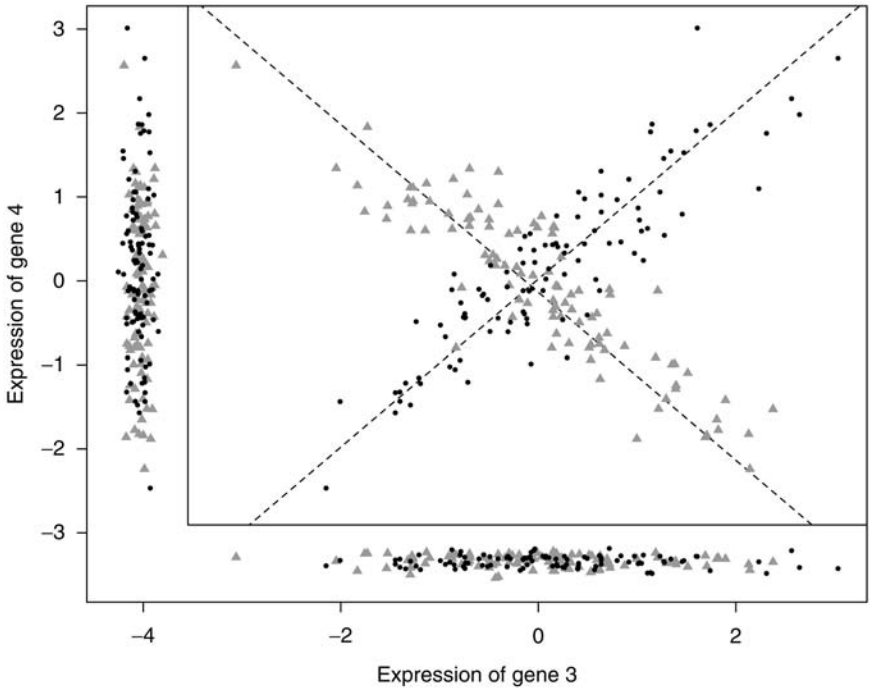


Fig. 2. Artificial example of joint differential expression (cross) of a pair of genes. The format is the same as that of **Fig. 1**. For this data set $S_{\text{cross}} = 1.64$ and $S_{\text{shift}} = -0.11$.

A second example is shown in **Fig. 2**. Herein there is no obvious demarcation in space, and again, neither gene alone predicts the phenotype. However, in samples from the grey phenotypes, the expression of the two genes are negatively correlated, whereas a positive correlation occurs in samples from the black (●) phenotype. Therefore, this will be referred to as a "cross" pattern. Biologically, this pattern could occur when two genes are involved in a common process in one phenotype, but perform separate or complementary activities in the other. Alternatively, it could reflect an "on/off" mechanism. If both genes are off (expression values <0), or both genes are on (expression value >0), the black phenotype is observed, whereas if only one of the genes is on, the grey phenotype is observed.

Both cross and shift patterns may only be identified when both genes are considered at the same time. This motivates defining joint differential expression as a departure from the null hypothesis of identical joint distributions, coupled with a weaker or no departure from the null hypothesis in the 1D marginal distributions. This definition is not entirely precise, because the term "weaker" needs to be specified. It is designed to guide the search toward gene pairs (or more broadly sets) that would not be identified by one-gene-at-a-time searches, whereas still

allowing for cases wherein 1D differences exist but are small or harder to detect than the joint differences. Despite this indeterminacy, it is a useful guide in identifying pairs that may represent interesting novel biological interactions, as for example, genes that are in the same pathway.

In practice, identification of gene pairs with joint differential expression is challenged by the large number of possible pairs. The usual number p of genes on a chip is in the tens of thousands, so the number of gene pairs, $p(p-1)/2$, is usually in the millions. Challenges increase exponentially as sets of more than two genes are considered. In this chapter, several statistical methods are discussed that can be used to search for joint differential expression: a correlation-based approach ***(5,6)***, the liquid association (LA) ***(7),*** and a generalization ***(8)***, the expected conditional F-statistic (ECF) ***(9)*** and a novel entropy-based method are examined in some detail and compared in simulations. Some algorithmically and computationally more complex methods of investigating gene coregulation are also mentioned ***(4,10,11)***. Several classification and network analysis approaches search more generally for sets of differentially coexpressing genes. These methods seek larger sets of differentially expressed genes, often without specifying the size of the set in advance. In these cases the search space is too large for an exhaustive canvas, and so results depend on efficient search algorithms. Not surprisingly, methods tend to be complex both algorithmically and computationally. *See* **Note 4.1.** for a brief overview.

2. Materials

1. The CorScor R package, which implements the correlation-based method described in **Subheading 3.1.2.** is downloadable from http://stat.ethz.ch/~dettling/jde.html. The package uses the object definition of bioconductor—an open source and open development software project for the analysis and comprehension of genomic data, available at http://www.bioconductor.org/. Both require the R language, which is available at http://cran.r-project.org/.
2. The statistical tools for implementing the LA and projection-based LA (PLA) methods described in **Subheading 3.1.3.** are available as a downloadable R-package from http://kiefer.stat.ucla.edu/LAP/index.php?tools.
3. The R code for calculating ECF-statistics described in **Subheading 3.1.4.** is available from http://bioinformatics.med.yale.edu/microarray/BioSuppl.html.

3. Methods

3.1. Statistical Approaches

3.1.1. Notation

Most of the discussion is developed for the basic case in which microarray experiments are available for two different phenotypes or classes (say normal, denoted by one; and cancer, by two), and interactions between pairs of genes

are of interest. However, extensions to more than two classes and more than two genes are discussed in the context of specific approaches. The notation (1, 2) is used to represents both classes combined, and the subscript *k* is used to indicate class. In the basic case, *k* can be 1, 2, or (1, 2). A vector of G gene specific expression values is denoted by $X_1 \ldots X_G$ and often by *X*, *Y*, and *Z*, when there are only three genes. Then, for example $X = (x_1, \ldots, x_{nk})$ are the gene-expression levels of gene 1 in samples from class *k*, were n_k is the number of such samples. Uppercase *X*'*s* generally refer to a vector of expression levels, whereas lowercase *x*'s refer to the levels in individual samples.

3.1.2. Correlation-Based Scoring Functions

The correlation between any two genes within class *k* is measured by the class-conditional Pearson correlation coefficient

$$\rho_k = \frac{\sum_i^{n_k} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i^{n_k} (x_i - \bar{x})^2 \sum_i^{n_k} (y_i - \bar{y})^2}} \quad (1)$$

which ranges between –1 and 1. If $\rho_k > 0$, the observed expression levels of the two genes are positively correlated (that is they move in the same direction) in samples from class *k.* High conditional correlation occurs when the points within the same phenotype line up as a straight line going upward. If $\rho_k < 0$, levels are negatively correlated, and if $\rho_k = 0$ they are uncorrelated. For each pair of genes, two class-conditional correlation coefficients, ρ_1 and ρ_2, and one combined correlation coefficient, $\rho = \rho_{(1, 2)}$ can be derived using the available samples.

The difference in the class-conditional correlations

$$S_{cross} = |\rho_1 - \rho_2| \quad (2)$$

captures cross patterns ***(5,6)***. S_{cross} increases with an increasing absolute difference of the conditional correlations. The largest S_{cross} is two and can be achieved by perfect conditional correlation in both classes, with opposite signs, for example, $\rho_1 = 1$, and $\rho_2 = -1$. A S_{cross} of zero occurs when correlations are the same, irrespective of their magnitude.

An important related case is L-shaped joint distributions, formed by a horizontal and a vertical ellipsoid. These can be thought of as a rotation of **Fig. 2**, and occur when the phenotype affects the variance of the two genes so that each gene varies only within one class. The S_{cross} measure will not capture these patterns, and will have difficulties with cross pattern that approach an L-shape. However, 1D comparisons of classes one gene at a time, if appropriately designed, will identify these genes ***(12)***.

A different measure is needed to capture shifts, as in shift patterns one can have joint differential expression even though the two class-conditional correlations are very close. The quantity

$$S_{\text{shift}} = \left|\rho_1 + \rho_2 - \alpha\rho\right| \tag{3}$$

with values of α between one and two, produces good empirical results in detecting shifts. When $\alpha = 2$, it is proportional to the difference between the average of the conditional correlations, and the combined correlation. For illustration, $\alpha = 2$ is used throughout the chapter. In **Fig. 1**, if the grey and black lines overlap, then $S_{\text{shjft}} = 0$. When one of the two parallel lines for the two classes shifts up or down, the combined correlation, which is computed after pooling the two classes, decreases whereas the class-conditional ones remain the same. In this way, S_{shift} will capture shifting patterns.

For example, in **Fig. 1**, the two correlation-based scores are $S_{\text{cross}} = |–0.82 – (–0.83)| = 0.01$ and $S_{\text{shift}} = |–0.82 + (–0.83) – 2 \cdot (–0.68)| = 0.29$ whereas in **Fig. 2** $S_{\text{cross}} = |0.85 – (–0.80)| = 1.64$, and $S_{\text{shift}} = |0.85 + (–0.80) – 2 \cdot (0.08)| = 0.11$.

These two correlation-based scores can be used to capture joint differential expression. They are intuitive and are computationally feasible in large search space. The same ideas can be extended straightforwardly to other kinds of association measures for pairs of genes, such as Spearman's correlation. Generalizations to searching for joint differential expression of groups of more than three genes are not readily available, because of the pair-wise nature of correlations. The implementation of these measures is straightforward in any programming language with matrix manipulation functionality. The CorScor R package provides tools for fast evaluation of S_{cross} and S_{shift}, as well as visualization of interesting pairs and significance analysis. Use of the functions in the CorScor package requires previous installation of R and Bioconductor (*see* **Subheading 2.**), and assumes that appropriate preprocessing of expression data is been carried out before joint differential-expression analysis. Bioconductor packages affy (for Affymetrix chip experiments) and limma (for a variety of other experiments including most two-channel arrays), include state-of-the-art tools and produce normalized data sets that are stored into objects of the class exprSet, that can be used as input to CorScor.

Normalization of expression data can affect gene–gene correlations. Specifically, artifacts in gene–gene correlations can be caused by flooring of expression levels, and by exclusion of low-level readings followed by missing data imputation. The latter combination can result in marking genes with similarly low values as missing, and then replacing missing values with values with little variability but very high correlation across genes. This generates falsely high correlations and can impact the joint differential-expression analysis. The minimal

requirements for a joint differential-expression analysis using CorScor are the following two objects:

1. eset: either a matrix of gene-expression data, wherein columns correspond to samples and rows correspond to genes, or an instance of the Bioconductor exprSet class.
2. Classlabel: either a binary factor or a binary numerical vector, describing the phenotype of interest for each sample. If eset is an instance of the exprSet class, it can also be the name of a binary covariate in the phenoData slot of the exprSet.

The correlation scores can then be evaluated simply by the R command: corscor.output = corscor(eset, classlabel), which will generate a new object named corscor.output, of class corscor, described as follows.

Users can also specify any of the following optional inputs:

1. Annotation: an optional character vector, containing a preferred annotation for the gene names.
2. Scenario: a character string, describing which scenario should be considered in the correlation scoring method. The two implemented options are "gapsubst" for the gap/substitution, or "shift" scenario, and "onoff" for the on/off, or "cross" scenario.
3. Cor.method: a character string, describing the way in which correlations should be computed. The default is "default," which means that Pearson correlation is used in the gap/substitution scenario, and Spearman correlation in the on/off scenario. The choice of either "Pearson" or "Spearman" overrules this default.
4. Dumping: a logical, describing whether gene pairs with 10 or more exactly equal values of gene expression should be ruled out. This is designed to protect from major artifacts from flooring. The default is *true*, but as long as the gene expression data set is free of artifacts, this variable will not have any effect.

In turn the object of class CorScor generated by the command corscor.output = corscor(eset, classlable) will include the following slots:

1. Scores: a symmetrical matrix containing the CorScor values for each gene pair.
   a. x: a matrix containing a copy of the input gene-expression matrix.
   b. y: a numeric vector, which codes for the classlabels by zero and one.
2. Annotation: a character vector, containing the annotation for the genes.
3. Scenario: a character string, saying which scenario was used.
4. Cor.method: a character string, saying which correlation method was used.
5. Dumping: a logical, describing whether dumping was active or not.

The package also provides functionality for follow-up analyses. The functions *print* and *summary* can be used to obtain an overview of the best gene pairs and their CorScor values. The function *bestpairs* yields the (column) indices of these genes. The function *plot* yields scatterplots displaying the gene pairs, and was used to generate **Figs. 1** and **2**. Finally, *hmap* yields a heatmap-like, more general overview of the structure, such as shown in **ref. *6***.

3.1.3. LA and Generalizations

LA, developed by Li *(7)*, is another correlation-based method for identifying dynamically coexpressing genes. Assume X, Y, and Z are the standardized and normally distributed expression intensities of gene 1, 2, and 3 and let $f(z) = E(X,Y|Z = z)$, then

$$\mathrm{LA}(X,Y|Z) = Ef'(Z) = Ef(Z)Z = E(XYZ) .$$

When applied to a set of n samples, distributional assumptions are met by substituting standard normal quantiles for the order statistics of each variable and the LA score is estimated as:

$$\mathrm{LA}(X,Y|Z) = \left(x_1 y_1 z_1 + \ldots + x_n y_n z_n\right)/n . \quad (4)$$

PLA was developed by Li and colleagues *(8)* to extend LA to larger genes sets. As the name suggests, PLA applies the LA method to 2D projections of the gene space, selecting the projection that maximizes the LA score. Consider a candidate set of G genes with expression vectors $X = (X_1, \ldots, X_G)$ and an additional mediator gene Z. A 1D projection of X is a linear combination $a'X$ where a is a projection direction and has norm one. For a projection in a 2D space, the method requires that the two projection directions a and b be orthogonal to each other. After projection, PLA considers the LA between $a'X$ and $b'X$ mediated by Z, and seeks informative projections by maximizing $\left|a'\mathrm{E}\left(ZXX'\right)b\right|$ over all pairs of orthogonal projection directions a and b.

Algorithmically, this maximization can be implemented through eigenvalue decomposition of $E\left(ZXX'\right)$: begin with the matrix Σ formed by every possible liquid correlation with mediator Z, that is $\Sigma_{gg'} = LA(X_g,X_{g'}|Z) = E(X_g X_{g'} Z)$. If $\lambda_1, \ldots, \lambda_G$ be the ordered eigenvalues of Σ and $v_1, \ldots, v_G$ be the associated eigenvectors. Then the maximum absolute LA score is

$$\mathrm{PLA}(X_1 \ldots X_G|Z) = (\lambda_1 - \lambda_G)/2 \quad (5)$$

The optimal projection is to the plane defined by the orthogonal vectors $\pm(v - 1 + v_G)/\sqrt{2}$ and $\pm(v - 1 - v_G)/\sqrt{2}$. To facilitate interpretation, the signs of each vector are determined so that the greatest gene-to-gene variation is captured in the positive component of the vector. If the mediating variable Z is binary, perhaps representing a phenotype rather than gene expression, then the LA score is formally equivalent to the correlation-based measure S_{cross} described above. The simulations described in **Subheading 3.2.** are confined to the two-gene, two-class case, and so the LA score is represented there by S_{cross}. Likewise, it is to be expected that application of PLA when the mediating variable is binary and

maximizing S_{cross} over 2D projections, would be likely to capture planar crosses buried in the higher dimensional space.

3.1.4. The ECFs

The *F*-statistic originates in analysis of variance ***(13,14)*** when testing whether the variation in a response of interest, say gene expression, depends on a class-label. It is proportional to the ratio of the between class variance to the within class variance.

$$\frac{K-1}{n-I}F=\frac{\sum_k n_{\mathrm{k}}(\bar{x}_{\mathrm{k}}-\bar{x})^2}{\sum_{k,j}(x_{\mathrm{kj}}-\bar{x}_{\mathrm{k}})^2}$$

where $x_{\mathrm{k,j}}$ is the gene expression of the *j*-th individuals in class *k*, $k = 1, \ldots, K$; $\bar{x}_{\mathrm{k}}$ is the mean expression of samples in class *k* and *x* is the overall mean expression in the pooled classes. If the between group variance is much larger than the within group variance, it can be inferred that gene expression is related to phenotype.

The ECF extends the *F*-statistic to test for coexpression in pairs of genes ***(9)***. If *X* and *Y* represent the expression intensities for a pair of genes, the conditional *F*-statistic (for outcome *X* conditional on $Y = y$) can be written as:

$$F^*_{\mathrm{X}|\mathrm{Y}=y}=\frac{\sum_k p_{\mathrm{k}}\sigma^2_{\mathrm{X_k}}\left(1-\rho^2_{\mathrm{k}}\right)}{\sum_{k<k'}p_{\mathrm{k}}p_{\mathrm{k'}}\left[(\mu_{\mathrm{X_k}}-\mu_{\mathrm{X_{k'}}})-\left(\frac{\mu_{\mathrm{Y_k}}\rho_{\mathrm{k'}}\sigma_{\mathrm{X_k}}}{\sigma_{\mathrm{Y_k}}}-\frac{\mu_{\mathrm{Y_{k'}}}\rho_{\mathrm{k'}}\sigma_{\mathrm{X_{k'}}}}{\sigma_{\mathrm{Y_{k'}}}}\right)+\left(\frac{\rho_{\mathrm{i}}\sigma_{\mathrm{X_k}}}{\sigma_{\mathrm{Y_k}}}-\frac{\rho_{\mathrm{k'}}\sigma_{\mathrm{X_{k'}}}}{\sigma_{\mathrm{Y_{k'}}}}\right)y\right]^2}$$

where $p_{\mathrm{k}} = n_{\mathrm{k}}/n$ is the proportion of samples in class *k*, ρ_{k} is the class-conditional correlation of *X* and *Y*, $\mu_{\mathrm{X_k}}$ and $\mu_{\mathrm{Y_k}}$ are the class conditional means of *X* and *Y*, and $\sigma_{\mathrm{X_k}}$ and $\sigma_{\mathrm{Y_k}}$ are the class conditional standard deviations. The expression above depends on a specific value $Y = y$. To take the expectation over all possible values of *Y*, the conditional *F*-statistic is then weighted by the probability density of *Y* and integrated as follows:

$$E_{\mathrm{Y}}(F^*_{\mathrm{X}|\mathrm{Y}=y})=\int_Y F^*_{\mathrm{X}|\mathrm{Y}=y}f_{\mathrm{Y}}(y)\mathrm{d}y \tag{6}$$

This leads to the final form of the ECF-statistic:

$$\begin{aligned}E_{\mathrm{Y}}(F^*_{\mathrm{X}|\mathrm{Y}=y})=&\left[\sum_k p_{\mathrm{k}}\sigma^2_{\mathrm{X_k}}(1-\rho^2_{\mathrm{k}})\right]^{-1}\sum_{k<k'}\sum_{k''}p_{\mathrm{k}}p_{\mathrm{k'}}p_{\mathrm{k''}}\{[(\mu_{\mathrm{X_k}}-\mu_{\mathrm{X_{k'}}})\\&-(\mu_{\mathrm{Y_k}}\rho_{\mathrm{k}}\sigma_{\mathrm{X_k}}/\sigma_{\mathrm{Y_k}}-\mu_{\mathrm{Y_{k'}}}\rho_{\mathrm{k'}}\sigma_{\mathrm{X_{k'}}}/\sigma_{\mathrm{Y_{k'}}})\\&+(\rho_{\mathrm{k}}\sigma_{\mathrm{X_k}}/\sigma_{\mathrm{Y_k}}-\rho_{\mathrm{k'}}\sigma_{\mathrm{X_{k'}}}/\sigma_{\mathrm{Y_{k'}}})\mu_{\mathrm{Yk''}}]^2\\&+(\rho_{\mathrm{k}}\sigma_{\mathrm{X_k}}/\sigma_{\mathrm{Y_k}}-\rho_{\mathrm{k'}}\sigma_{\mathrm{X_{k'}}}/\sigma_{\mathrm{Y_{k'}}})^2\sigma^2_{\mathrm{Yk''}}\}.\end{aligned} \tag{7}$$

The ECF-statistic can also be simplified as below when $\sigma_{Xk} = \sigma_X$ and $\sigma_{Yk} = \sigma_Y$

$$E_Y(F^*_{X|Y=y}) = \left[\sum_k p_k(1-\rho_k^2)\right]^{-1} \sum_{k''}\sum_{k<k'} p_k p_{k'} p_{k''}\{[(\mu_{X_k} - \mu_{X_{k'}})/\sigma_X - \rho_k(\mu_{Y_k} - \mu_{Y_{k''}})/\sigma_Y + \rho_{k'}(\mu_{Y_{k'}} - \mu_{Y_{k''}})/\sigma_Y]^2 + (\rho_k - \rho_{k'})^2\}.$$

Full details of the derivation are found in **ref. *9***.

As in LA, the derivation depends on normally distributed data. To coerce each variable into a normal distribution, the order statistics are replaced with quantiles from the desired distribution. A key difference is that in the ECF version, the various phenotypic groups are permitted to have different means and variances. The reference distribution is a mixture of normals having those parameters. Quantiles are calculated by simulation. An advantage of the ECF-statistic is that it can be applied to data with an arbitrary number of phenotypic classes. However, there is no available extension to more than two genes.

3.1.5. An Entropy-Based Measure

In this section, a novel entropy-based approach is introduced. This method was developed to encompass both correlation-based scores and generalize them to larger sets of genes. The Shannon entropy of a variable X is a measure of its randomness ***(15)***. Informally, it might be thought of as the inverse of the accuracy with which one can predict the outcome of a random process. Thus, a fair coin flip has higher entropy than that of a coin weighted to land heads-up most of the time. If X takes on values in the set $\{1, \ldots, m\}$ with $Pr(X = x) = p_x$ then the entropy is calculated as $E_X = -\sum_{x=1}^{m} p_x \log_2(p_x)$. Entropy is always nonnegative. It achieves its greatest value, $\log_2(m)$ when the m possible outcomes of X are equally likely (and this have probability $1/m$), and approaches zero when one of the outcomes becomes virtually certain.

The intuition behind the entropy-based approach is that if a set of G standard normal random variables are well correlated, then the multidimensional scatterplots will describe long, narrow ellipsoids. Equivalently, the eigenvalues of the correlation matrix, which are proportional to the lengths of the various axes of the ellipsoid, will include only one or a few large values among many smaller ones. If those eigenvalues are standardized to sum to one, and treated as probabilities, then the distribution they define has one nearly certain outcome corresponding to the long axis of the scatterplot, and so will have low entropy. As illustrated in **Fig. 3** both of the motivating examples, cross and shift, the scatterplot of the pooled data fills a broad ellipse, so the entropy derived from the pooled correlation matrix will be high, whereas the long and narrow class-specific scatterplots

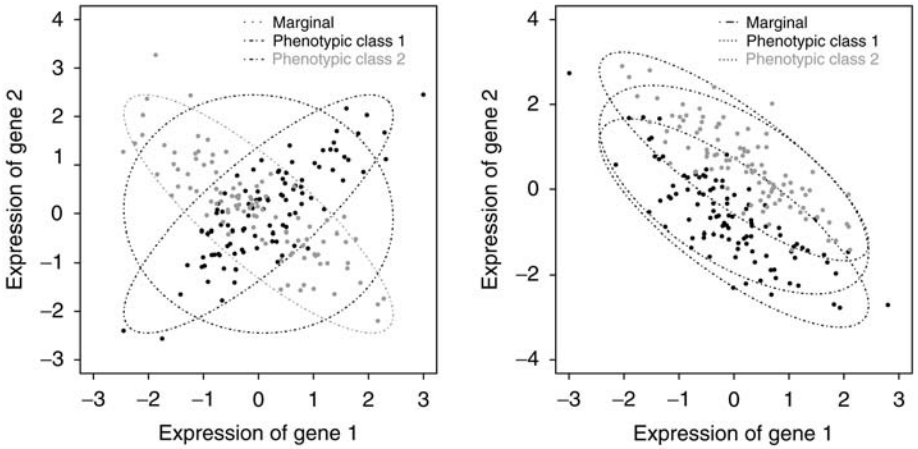


Fig. 3. An illustration of the entropy of standardized eigenvalues with or without the information of phenotypic classes, in both crosses and shift patterns.

are characterized by low entropy. Accordingly, sets of genes will be scored by the difference between pooled and class-specific entropies, using a permutation test to assess significance.

The full entropy-based score is now developed. First calculate the two class-specific correlation matrices, Σ_1 and Σ_2, as well as the correlation for the pooled data Σ. The eigenvalues of each are calculated and normalized to sum to 1, so for class k the standardized eigenvalues are $\Lambda_k = (\lambda_{k1}, \ldots, \lambda_{kG})/G$. The entropy of the standardized eigenvalues is calculated and so that the values do not depend on dimension, is further standardized to the maximum possible entropy value $\log_2(G)$. Thus, the class-specific, standardized entropy can be written as

$$E_k = -\frac{\sum_{g=1}^{G} \lambda_{kg} \log_2(\lambda_{kg})}{G \log_2(G)} \tag{8}$$

If, as above, $X = (X_1, \ldots, X_G)$ is the expression intensity vectors for the candidate set of G genes then the entropy score can be written as

$$\mathrm{ENT}(X) = E - (E_1 + E_2)/2 \tag{9}$$

where E is the entropy obtained from pooling the classes. The ENT score is easily extended to $K > 2$ classes as follows

$$\mathrm{ENT}(X) = E - \frac{1}{K}\sum_k E_k . \tag{10}$$

In the special case of two genes, the class-specific, standardized entropy is a function of the class-specific correlation coefficient ρ_k

$$E_k = -\left[\frac{1+|\rho_k|}{2}\log_2\left(\frac{1+|\rho_k|}{2}\right)+\frac{1-|\rho_k|}{2}\log_2\left(\frac{1-|\rho_k|}{2}\right)\right] \quad (11)$$

Although there is no simple closed form expression for entropy for larger sets of genes, eigenvalues can be calculated efficiently and methods for doing so are implemented in many computational programs.

To assess significance, classlabels are repeatedly permuted and the entropy score is recalculated over all gene sets under consideration. Each permutation gives a distribution of null scores, which are averaged to produce a stable reference distribution. As usual, the *p*-value is the proportion of null scores that exceed the observed value. There is one caveat concerning the calculation of the pooled correlation value. If the class-specific sample sizes are very different, the larger one may dominate the pool. In that case, one might weigh by sample size when calculating the pooled correlation to equalize the influence of the two classes.

3.2. Simulation-Based Evaluation of Methods

To compare methods, data were simulated from each of the archetypical two-class examples, cross and shift, performance of each method was evaluating by calculating power. In the two-class case, LA is equivalent to the S_{cross} and so the two methods coincide in these simulations. The data was simulated from normal distributions, with a sample size of 50 for each of the two classes. Type I error was set to $\alpha = 0.05$ throughout. Null distributions for all methods were obtained by recalculating scores after permuting class labels. The power was computed as the frequency of simulated data sets with a test statistic more than the 95-th quantile of the null distribution.

To simulate shift patterns samples were drawn from class-specific bivariate normal distributions. Class 1 was drawn from a $N(\mu_1 = d, \mu_2 = d, \sigma_1 = 1, \sigma_2 = 1, \rho = \rho_0)$ distribution and class 2 was drawn from $N(\mu_1 = -d, \mu_2 = -d, \sigma_1 = 1, \sigma_2 = 1, \rho = \rho_0)$, where d is allowed to vary. Thus, the expression levels for both genes are increased in one class and decreased in the other, whereas correlation for the two classes remains identical.

Figure 4 demonstrates the power of the three methods to detect shift patterns. Power is shown as a function of the shift d between the distributions of the two classes (in the x-axis) and the correlation of the class-conditional distributions (by panel). The largest ECF-statistic is consistently among the most powerful. S_{shift} matches its power at low correlations, whereas the entropy score matches it at higher correlations. For all methods, power increases with both the shift and the class-conditional correlation, with exception of combinations of low correlation and large shifts, a situation in which increasing the shift will decrease the

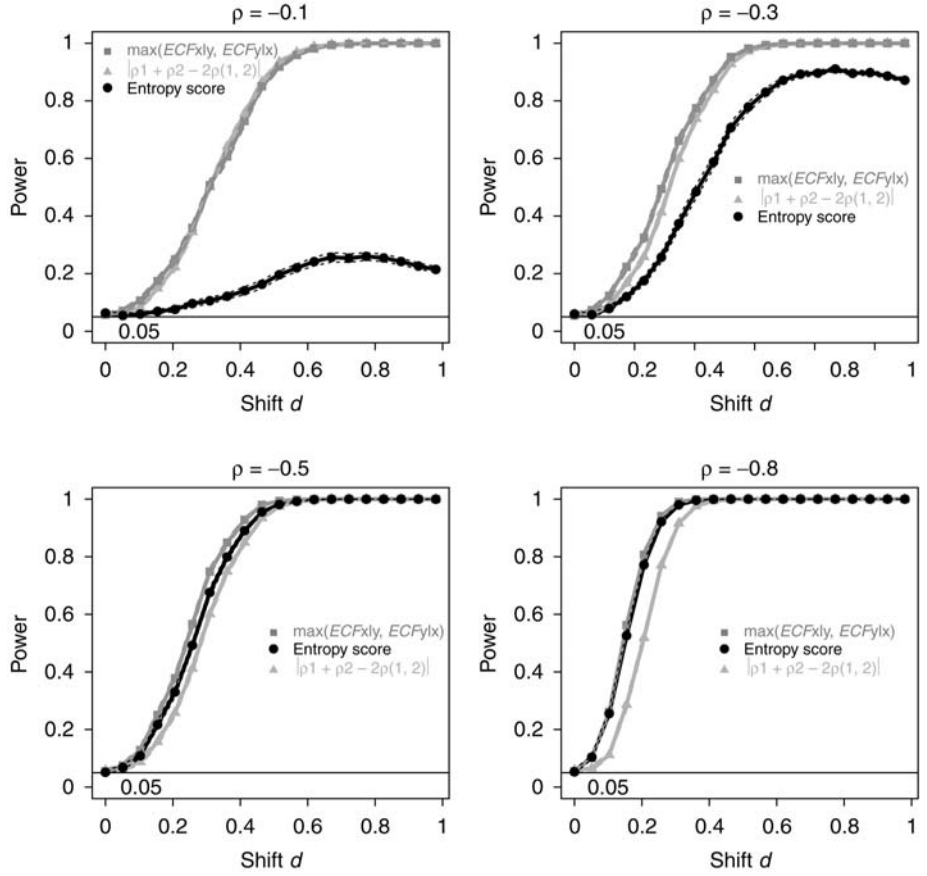


Fig. 4. Comparisons of power for detecting shift patterns as a function of the shift d between the distributions of the two classes. Methods are identified by different shades of gray. Correlations used to simulate the bivariate normal are listed at the top of each panel.

power of the entropy score, because the highly bimodal marginal distribution will display lower entropy than each of the components.

Generally, the power of entropy score increases with increasing shift effects. However, when the marginal separation becomes very large, the entropy score can decrease, as two well-separated subgroups can create a narrow ellipsoid when pooled. This feature of the entropy score has the advantage that gene combinations with large univariate separations are less likely to be captured in the top ranking sets. Because the gene with significant marginal effect can be found more easily by one-gene-at-the-time analyses, one might wish to exclude those when looking for gene combinations.

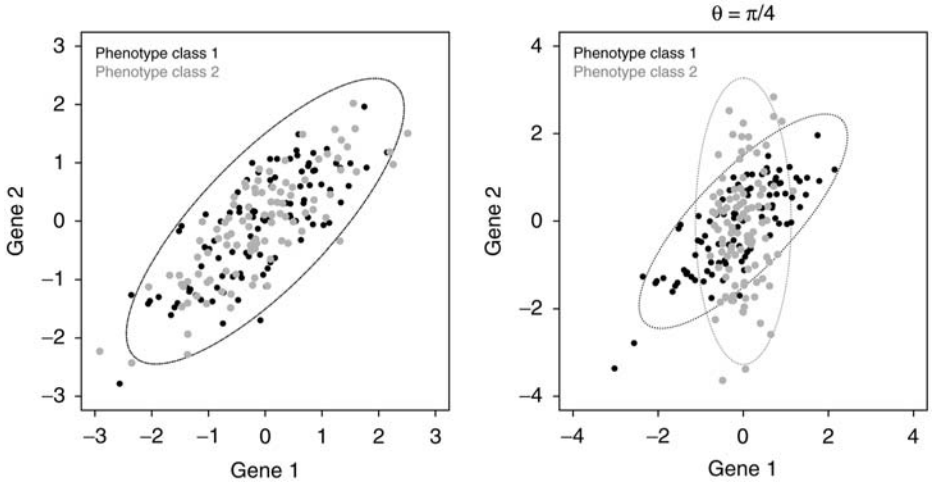


Fig. 5. An illustration of the approach used for simulating cross patterns, when $\theta = \pi/4$.

Cross patterns were simulated by first drawing observations for each of the two classes from a bivariate normal distribution $N(\mu_1 = 0, \mu_2 = 0, \sigma_1, \sigma_2, \rho)$; then the class 2 data is modified by rotating the points for that class by the angle θ, whereas data for class 1 are held fixed. An example, with a rotation of 45°, corresponding to $\theta = \pi/4$ is shown in **Fig. 5**. The difference between classes becomes more pronounced as θ increases until it reaches its maximum at $\theta = \pi/2 = 90°$. When $\theta = \pi = 180°$, the correlations are again equal. In this approach both the combined and the class-conditional gene variances vary with θ, but the standardized eigenvalues, $\lambda_1^* = \frac{1+|\rho|}{2}$, and $\lambda_2^* = \frac{1-|\rho|}{2}$ (the axes of ellipse) were kept fixed.

Figure 6 demonstrates the power of the three compared methods to detect cross patterns of the type simulated. Power is shown now as a function of the angle θ between the ellipsoids of the two classes (on the *x*-axis) and the eigenvalues used to simulate the bivariate normal ellipsoid (by panel). For all methods, power increases with both the angle and the class-conditional correlation. The ECF-statistic depends on which variable is chosen as the conditioning variable. When searching for interesting pairs, it is suggested to use the largest of the two statistics. The largest ECF-statistic is the most powerful, although by a small margin, when the data is simulated to have a very long and narrow shape. Methods are essentially equivalent for low and moderate correlations, but at high correlations the correlation-based approach loses power on small-angle rotations compared with the other two alternatives. This is because the class conditional correlation of the rotated data changes more slowly under these circumstances than other properties of the joint distribution. Because the correlation measure simply compares the two correlations, it lacks power to detect the difference.

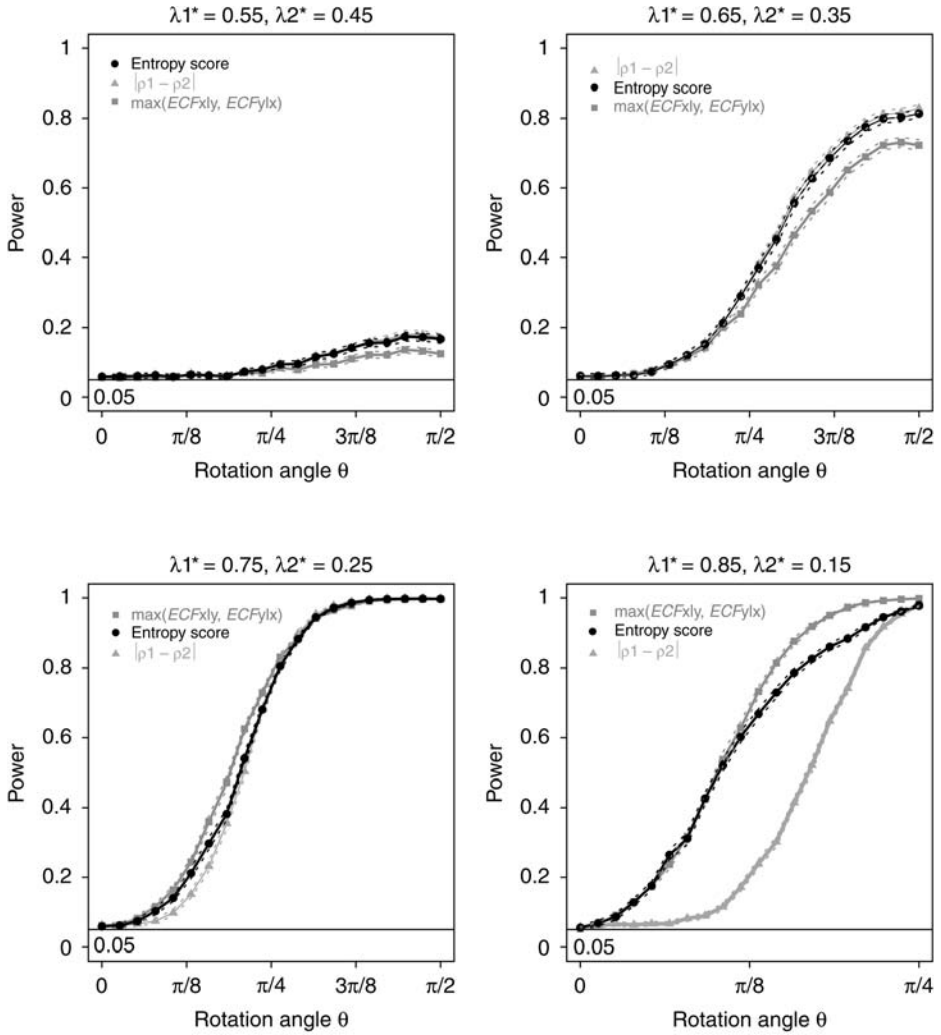


Fig. 6. Comparisons of power for detecting cross patterns as a function of the angle θ between the correlations of the two classes. Methods are identified by different shades of gray. Eigenvalues used to simulate the bivariate normal are listed at the top of each panel.

Now consider separately the two versions of the ECF-statistic. The ECF_{ylx} investigates how well one can predict class using y for a fixed x, whereas the ECF_{xly} investigates how well one can predict class using x for a fixed y. **Figure 7** shows power of both versions in an additional scenario in which the ellipsoids are yet narrower than in **Fig. 6**. The ECF_{xly} and ECF_{ylx} show markedly different behavior, as the power of ECF_{ylx} decreases when the data of phenotypic class 2 is

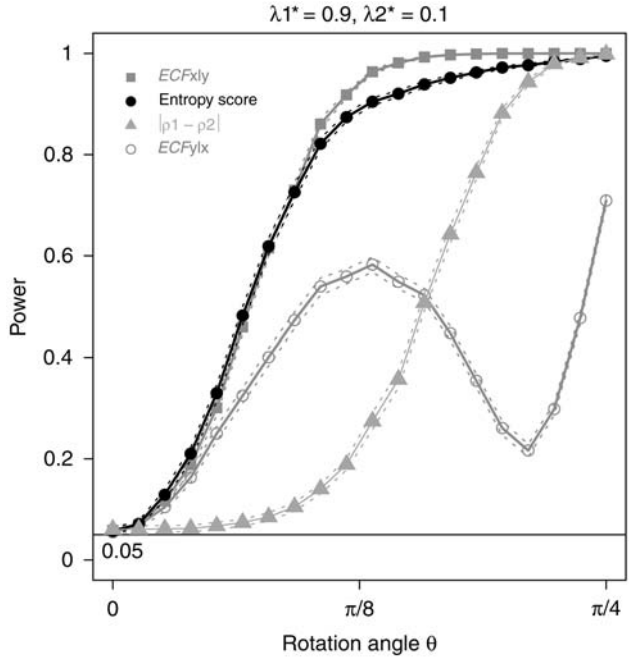


Fig. 7. The power curves of $ECF_{x|y}$, ENT, $|\rho_1 - \rho_2|$, and $ECF_{y|x}$.

vertically aligned, that is when θ is around π/4. To investigate further the reason for this behavior, note that the rotated data points (x', y') of genes 1 and 2 can be expressed in term of the original points (x, y) as:

$$x' = \cos\theta x - \sin\theta y,$$
$$y' = \sin\theta x + \cos\theta y.$$

The resulting variance and correlation of the data in phenotypic class 2 after rotation can be written as:

$$\sigma_{x2}^2 = 1 - 2\cos\theta\sin\theta\rho \tag{12}$$

$$\sigma_{y2}^2 = 1 + 2\cos\theta\sin\theta\rho \tag{13}$$

$$\rho_2 = \frac{(\cos\theta^2 - \sin\theta^2)\rho}{\sqrt{(1 - 2\cos\theta\sin\theta\rho)(1 + 2\cos\theta\sin\theta\rho)}} \tag{14}$$

The ECF-statistics is shown in **Eq. 7**. In the simulation, set $\mu_{x1} = \mu_{x2} = \mu_{y1} = \mu_{y2} = 0$, and $\sigma_{x1} = \sigma_{x2} = 1$, and therefore the ECF-statistic can be expressed as a function of θ and ρ alone. The value of $ECF_{y|x}$ when the class 1 data is fixed and

Table 1
A Synopsis of the Key Features of the Methods

Method	Joint differential expression pattern targeted	Multidimensional extension available	Effect of large increase in marginal separation	More than two phenotypic classes	Continuous mediator variable
S_{cross}	"cross"	No	None	No	No
S_{shift}	"shift"	No	Depends on data	Yes	No
ENT	"cross" and "shift"	Yes	Decreases	Yes	No
LA (PLA)	"cross"	Yes	None	Yes	Yes
ECF	"cross" and "shift"	No	Increase	Yes	No

class 2 data is rotated with a degree θ counterclockwise, is not monotone in θ for a fixed ρ. Rather the value of ECF_{ylx} increases with θ except for a dip when θ is such that the data of class 2 is vertically aligned. This results in the non-monotone pattern of the power function of **Fig. 7**. This is owing to the fact that the ECF_{ylx} calculation is weighted by the marginal distribution of x, which results in narrowing the comparison between classes to a region of relatively small class effect.

3.3. Conclusion

Several statistical approaches are now available for identifying joint differential expression. In this chapter a definition of joint differential expression is proposed, several approaches are reviewed, and a simulation is used to compare the three methods that can at present be used for exhaustive searches of all pairs or all triples of genes in realistically large gene sets. A compendium of properties of different methods is presented in **Table 1**. Overall, no method appears to be uniformly superior. However, in the two-gene analyses investigated in the simulation, the performance of the ECF-statistic is consistently reliable. The CorScor approach maintains an intuitive interpretability and is by far the most attractive computationally. The entropy scoring approach shows promise for comprehensive searches of three-gene sets.

4. Notes

4.1. Other Approaches

In briefly this section, a few other distinctive approaches are described. For full details, the reader is referred to original sources. Methods described in

this section were not included in the simulation-based analysis. For notation in this section, consider a candidate set of G genes and two phenotypic groups with n_1 and n_2 samples, respectively. Then u_i, $i = 1,..., n_1$ and v_i, $i = 1,..., n_2$ are G dimensional vectors representing sample intensities measured over genes in the set.

In a recent article, Xiao and colleagues ***(4)*** considered multivariate searches for differentially expressed gene combinations. Their algorithm is built on a previously proposed multivariate test statistic ***(16)*** and successive selection of differentially expressed sets of genes ***(17)***. Their goal is to uncover subsets of predefined size G such that the multivariate distributions of expression in the two phenotypes differ. To score candidate gene sets users need to choose a kernel function F(u, v) and calculate

$$S = \frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{i'=1}^{n_2} F(u_i, \upsilon_i') - \frac{1}{n_1^2}\sum_{i=1}^{n_1}\sum_{i'=1}^{n_2} F(u_i, u_{i'}) - \frac{1}{n_2^2}\sum_{i=1}^{n_2}\sum_{i'=1}^{n_2} F(\upsilon_i, \upsilon_{i'})$$

wherein the sums are taken over all pairs of samples in each class. Distance functions are classical choices for $F(u, \upsilon)$, the authors use the Euclidean distance function throughout. With that choice, the score S can be described as average between-group distance minus average within-group distance.

The search starts with an arbitrary set of G genes, which are then exchanged, one at a time, at random, with candidate genes from outside the set. Exchanges that do not improve the score are discarded, whereas if an exchange improves the score, the set is modified accordingly and the search continues for a set number of steps, or until predefined criteria is met. Cross-validation is used to stabilize the results of the search procedure. A permutation test is used to evaluate significance, and a multiple-testing procedure is developed to control family-wise error rate when selecting combinations of genes. The approach uncovers sets that potentially consist of combinations of jointly and marginally differentially expressed genes. Kostka and Spang ***(10)*** took a different approach to the basic problem. The goal of their methodology was to identify sets of genes, which are normally tightly coregulated, but which disregulate in a diseased state. The measure of coregulation of a gene set G, within class k, denoted $S(G,k)$, was previously suggested by Cheng and Church ***(18)***. It is calculated as the mean squared residual obtained over values of g in 1, ..., G and i in 1, ..., n_k, after fitting the following model

$$y_{ig} = a_G + b_k + c + \varepsilon_{ig}.$$

Small values of S indicate strong correlation so if the genes in G are tightly correlated in phenotypic group k but not in group k', then the ratio $S(G, k)/S(G, k')$ will be small. The search procedure begins with an arbitrary set of genes,

adding or removing genes to improve the score until further improvement is not possible, or until a predetermined number of iterations are completed. Gene sets are not required to be of a particular size, so they are added or removed individually.

In recent years, several network-based methods for discovering gene-coexpression patterns have been proposed. Bayesian networks are most frequently used in this fashion, although Boolean networks and other approaches have been applied as well ***(19)***. Bayesian networks offer a graphical representation of the dependence structure among a set of variables. In the gene-expression setting, genes are represented in the network by nodes, with edges connecting those nodes when genes strongly coregulate. The parameters of the underlying Bayesian model can be estimated independently of the graphical component of the network model and summarized by nongraphical means. However, the graphical network representation offers additional intuitive, potentially informative, and possibly biologically relevant features with which gene interactions can be characterized. Examples include the degree of connectivity seen in a set of correlated genes and the number of distinct components, or gene sets that can be identified. Candidate network structures can be scored for goodness of fit of the dependence relationships observed in the data. Graphical features that characteristically associate with high-scoring network structures are likely to be interesting.

Every possible network structure corresponds to a Bayesian model, which can be fitted to the data. The score for a network is calculated as the log likelihood of the corresponding model, and so the best-fitting network is one that corresponds to the maximum likelihood model. The space of all possible networks grows exponentially with the number of genes/nodes under consideration and so, as in the methods described earlier in this section, greedy stochastic search algorithms are used to navigate the network space. Edges are added or removed at random to improve the overall fit and the search stops after a predetermined number of steps or when improvement is no longer possible.

Work by Friedman and colleagues ***(11,20)*** is representative of results in this area. The investigators search for the network structure that best fits a set of gene-expression data, identify biologically interesting graphical features of that network, and assign bootstrap-based confidences to the discoveries. The *Markov blanket of a* set of genes/nodes is one such feature. Imagine that a set of nodes *X* is isolated in a corner of the network, relating to the remaining nodes only through the mediation of a small set of neighbors *Y*. Then *Y* is described as the Markov blanket of *X*. A bootstrap procedure is used to assign confidences to discovered features. The data is repeatedly resampled with replacement, each time the search for the best-fitting network structure is performed on the resampled data. The proportion of samples exhibiting the feature under study is taken as the confidence level for the feature.

References

1. Schena, M. (2000) *Microarray Biochip Technology.* BioTechniques Press, Westborough, MA.
2. Pan, W. (2002) A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics* **18,** 546–554.
3. Parmigiani, G., Garrett, E. S., Irizarry, R. A., and Zeger, S. L. (eds.) (2003) *The analysis of gene expression data: an overview of methods and software.* Springer, New York, 1–20.
4. Xiao, Y., Frisina, R., Gordon, A., Klebanov, L., and Yakovlev, A. (2004) Multivariate search for differentially expressed gene combinations. *BMC Bioinformatics* **5,** 164.
5. Shedden, K. and Taylor, J. (2004) Differential correlation detects complex associations between gene expression and clinical outcomes in lung adenocarcinomas. *Methods Microarray Data Anal.* **IV,** 121–132.
6. Dettling, M., Gabrielson, E., and Parmigiani, G. (2005) Searching for differentially expressed gene combinations. *Genome Biol.* **6(10),** R88.
7. Li, K. C. (2002) Genome-wide coexpression dynamics: Theory and application. *Proc. Natl. Acad. Sci.* 16,875–16,880.
8. Li, K. C., Liu, C. T., Sun, W., Yuan, S., and Yu, T. (2004) A system for enhancing genome-wide coexpression dynamics study. *Proc. Natl. Acad. Sci. USA* **101(44),** 15,561–15,566.
9. Lai, Y., Wu, B., Chen, L., and Zhao, H. (2004) A statistical method for identifying differential gene-gene co-expression patterns. *Bioinformatics* **20,** 3146–3155.
10. Kostka, D. and Spang, R. (2004) Finding disease specific alterations in the co-expression of genes. *Bioinformatics* **20(Suppl 1),** i194–i199.
11. Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.* **7(3–4),** 601–620.
12. Tomlins, S. A., Rhodes, D. R., Perner, S., et al. (2005) Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. *Science* **310(5748),** 644–648.
13. Box, G. E. P., Hunter, W. G., and Hunter, J. S. (1978) *Statistics for experimenters: An introduction to design, data analysis, and model building.* Wiley, New York.
14. Kerr, M. K., Martin, M., and Churchill, G. A. (2000) Analysis of variance for gene expression microarray data. *J. Comput. Biol.* **7(6),** 819–837.
15. Shannon, C. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.* **27(3),** 379–423.
16. Szabo, A., Boucher, K., Carroll, W. L., Klebanov, L. B., Tsodikov, A. D., and Yakovlev, A. Y. (2002) Variable selection and pattern recognition with gene expression data generated by the microarray technology. *Math. Biosci.* **176(1),** 71–98.
17. Szabo, A., Boucher, K., Jones, D., Tsodikov, A. D., Klebanov, L. B., and Yakovlev, A. Y. (2003) Multivariate exploratory tools for microarray data analysis. *Biostatistics* **4(4),** 555–567.

18. Cheng, Y. and Church, G. M. Biclustering of expression data. 93–103.
19. Heckerman, D. (1995) A tutorial on learning with bayesian networks. *Tech. rep.*, Microsoft Research, Redmond, Washington. Revised June 96.
20. Friedman, N. (2003) Probabilistic models for identifying regulation networks. *Bioinformatics* **19(Suppl 2),** 1157.

III

Experimental Methods

11

Gene Function Analysis Using the Chicken B-Cell Line DT40

Randolph B Caldwell, Petra Fiedler, Ulrike Schoetz, and Jean-Marie Buerstedde

Summary

Quidquid agis, prudenter agas et respice finem!—*Whatever you do, do it wisely and consider the goal.* In consideration of that sage advice, the chicken B-cell line DT40 is an excellent model cell system to study the function of vertebrate genes. In addition to being highly amenable to gene manipulations, the recent influx of genome and gene/protein resources allows for the straightforward selection, design, and targeting of candidate genes for knockout analysis. This chapter will give a step by step standardized protocol to creating a gene knockout mutant in DT40. With careful consideration, the methods and protocols described herein can be easily modified to allow for further gene manipulations such as creating a knockin or a conditional mutant.

Key Words: Conditional mutant; DT40; gene function analysis; gene manipulation; genotype; homologous recombination; knockin; knockout; phenotype; targeting.

1. Introduction

Advantages of DT40 include the high homologous recombination activity facilitating targeted gene manipulations ***(1)***, the availability of tightly regulated conditional gene expression systems and the ability to study genetic interactions by the stepwise modification of multiple loci using marker recycling ***(2)***. DT40 is particularly suited for genetic analysis if (1) the main phenotype of the gene loss can be studied in cell culture, (2) gene functions are conserved during evolution, and (3) interactions between multiple genes need to be analyzed and the genes of interest are essential for murine embryonic development.

The recent release of the chicken genome sequence has greatly benefited the DT40 research community ***(3)***. For the first time, the genome can be searched

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

for sequences that are conserved during vertebrate evolution between mammals and chicken. This greatly expedites the identification of worthwhile candidate genes and the subsequent analysis and interpretation of mutant phenotypes. Targeting constructs are now easily derived from the genome sequence to delete gene coding regions, modify regulatory sequences, or add gene coding tags for the visualization and purification of protein complexes. Furthermore, enhancing the use of DT40 in gene analysis is (1) the *International Chicken Polymorphism Map Consortium's* release of a comprehensive single nucleotide polymorphism analysis ***(4)***, (2) the Second Report on Chicken Genes and Chromosomes 2005 ***(5)***, and ***(3)*** the first serial analysis of gene expression (SAGE) of the chicken B-cell and DT40 genes by Wahl et al. ***(6)***. In addition, genes expressed in DT40 are often available as full-length cDNA clones from a large bursal cDNA library, thus enhancing DT40's usefulness by further facilitating the complementation of gene disruption phenotypes and the artificial expression of proteins ***(7)***. Thus, DT40 is well suited to study gene function through knockout analysis.

1.1. The Design

1.1.1. Choice of Knockout Candidate Genes

Candidate genes for knockouts in DT40 are usually chosen based on structural homology to genes with known function in other organisms. A thorough check of what is known about the functions of the homologs and whether the suspected phenotype can be measured in cell culture is highly recommended at this stage. Retrieve the nucleotide and amino acid sequences of the nonchicken homologs either from the public databases (http://www.ncbi.nlm.nih.gov or http://www.ebi.ac.uk/embl/) or other sources.

1.1.2. Retrieval of Chicken EST or cDNA

Although an evolutionary conserved cDNA or protein sequence from a nonchicken species may yield a positive result in a basic local alignment search tool (BLAST) search against the chicken genome, only knockouts of genes expressed in DT40 can be expected to give a phenotype. If there is doubt of whether the gene is expressed, searches of the Bursal Transcript Database or the bursal and DT40 SAGE tag databases are recommended (http://pheasant.gsf.de/DEPARTMENT/). Careful analysis of ESTs and the cDNA sequence of a knockout candidate gene are also advantageous to define the exact exon–intron boundaries of the genomic locus. The chicken full-length cDNA may also be needed to complement the mutant phenotype.

To search for ESTs or cDNAs within The Bursal Transcript Database, enter the cDNA nucleotide query sequence of the closest homolog from a nonchicken species or the chicken cDNA deduced from the chicken genome or other

sources as a query sequence. Select the "BLAST search" link from the chicken transcript website. The result lists any homologous ESTs or full-length cDNAs among the sequences from two bursal cDNA libraries. The link may be followed behind the sequence name to obtain the sequences and other information such as whether there are other overlapping sequences or SAGE tags derived from Bursal and DT40 SAGE tag libraries. The sequences in The Bursal Transcript Database are from bursal lymphocytes, but DT40 is derived from these cells and is similar in its transcription profile. Thus, sequences found in The Bursal Transcript Database may be expected to be expressed in DT40 as well. On the other hand, tags from the DT40 SAGE library are direct evidence for expression and their relative frequencies indicate the steady-state level of corresponding transcripts.

Although, the Bursal Transcript Database includes close to 30,000 ESTs from different cDNA clones and more than 2250 unique full-length cDNAs, the absence of sequences matching a candidate gene does not necessarily indicate lack of expression in DT40. The failure may be because of cDNA cloning difficulties or low transcript abundance. Because there are other large-scale chicken EST and cDNA databases in the public domain, they too can be probed for chicken transcript sequences (i.e., http://www.chick.umist.ac.uk/ or http://www.chickest.udel.edu/). If this is successful, the expression of the candidate gene in DT40 can be confirmed by reverse polymerase chain reaction (PCR) of DT40 mRNA.

1.1.3. Finding the Target Locus

The best ways to define the exon–intron structure of the target locus is by running a BLAST search of the full-length chicken cDNA sequence against the chicken genome assembly (http://www.ncbi.nlm.nih.gov/projects/genome/guide/chicken/ or http://www.ensembl.org/Gallus_gallus/index.html) or perform a BLAT search (http://genome.ucsc.edu/). However, if the chicken cDNA is not available, a search of the cDNA or protein sequence of the homolog from another species can be tried. The successful identification of the coding regions depends on the degree of interspecies transcript conservation. If the chicken full-length cDNA is available, a BLAST or BLAT search against the chicken genome should reveal the precise exon–intron structure of the locus. The only possible problem can be errors in the cDNA sequence or gaps or errors in the genome sequence. The chicken genome assembly is estimated to be 90–95% complete.

If only the cDNA sequence of a nonchicken homolog is available, a BLAST/BLAT search against the chicken genome may reveal the conserved coding regions of the chicken locus. Although both the nucleotide and the amino acid sequence can be tried, the amino acid sequence may be the most suitable

query because it is likely to be more conserved than the nucleotide sequence in distantly related species. The obtained information might be enough for the design of the targeting construct, if the deletion of a conserved region is all that is desired. As bioinformatics technology grows, the Internet-based tools available will reflect the enhanced capabilities as either experimentally known or algorithmically predicted sequence structure. The various sites may also show the predicted 5′ and 3′ noncoding sequences of the transcript, but this information is sometimes incomplete or even wrong and it should be considered tentative without the support of an experimentally confirmed full-length cDNA sequence.

1.1.4. Coding Sequence Conservation

The analysis of the primary amino acid sequence conservation is crucial to anticipate the gene knockout phenotype and to plan its analysis. Most of the candidate genes are chosen based on the known function of a homolog in other organisms. The higher the primary amino acid sequence conservation of the candidate gene, the more likely it becomes that its function is likewise conserved. Furthermore, essential structural domains need to be defined for the construction of targeting vectors if a null mutation is desired, but a complete gene deletion is not feasible. Use the amino acid sequence encoded by the chicken candidate gene in a BLAST search of the public protein or EST databases (e.g., http://www.ncbi.nlm.nih.gov/BLAST/) to retrieve the sequences of the likely homologues from other vertebrate species (e.g., human, mouse, rat, and fish). Use the National Center for Biotechnology Information (NCBI) Conserved Domain Database service to search for conserved domains. A BLAST search of the protein database will automatically identify conserved domains and will display a link that can be followed for more information on the identified domains. The usefulness of this and the Gene Ontology (GO, http://www.geneontology.org/) databases should only increase as more is learnt about the relationship that domain structures confer on function.

Align and compare the amino acid sequences of orthologs from different species. One way to do so is to copy the sequences of the homologs in FASTA format to the program "BioEdit" to align and compare the amino acid sequences. BioEdit (http://www.mbio.ncsu.edu/BioEdit/bioedit.html) is a free software program that allows the user to import multiple sequences from the various databases, accepts both text and FASTA formats, and has the ability to toggle between nucleic acid and amino acid views. This program is rather easy for the beginner to use and allows for a detailed visual view that is uncomplicated and straightforward.

1.2. Design of the Targeting Construct

Phenotypes of DT40 mutants are difficult to predict and in most cases a null mutation of the candidate gene is aimed for. Deleting the entire gene coding region

is the safest way to produce such a mutation. Deletions of up to 20 kb have been introduced into DT40 by targeted construct integration, but there is the impression that the efficiency of targeting is less predictable for very large deletions. Therefore, if the target gene locus is large, then alternative strategies need to be considered. A common approach is to introduce an early truncation of the open reading frame in combination with the deletion of a region encoding an indispensable structural domain. The resulting locus most likely encodes only a shortened peptide, and because of the deletion, even aberrant translation or splicing cannot lead to a functional protein. The arms should be amplified by PCR using genomic DT40 as template. This assures that the arm sequences are isogenic to at least one allele of DT40 that may increase the targeting efficiencies.

The plasmid insert of standard DT40 targeting vectors consists of a loxP flanked drug resistance marker cassette flanked 5′ and 3′ by sequences derived from the target locus (*see* **Fig. 1A**,**B**). The 3′ end of the upstream arm and the 5′ end of the downstream arm define the boundaries of the target gene deletion. The plasmid is linearized before transfection using a restriction enzyme (RE) (i.e., *Not*I) whose site is present within the plasmid, but not within the insert.

Many of the rules for the design of DT40 targeting vectors are to a certain degree arbitrary and might be changed if the goal is a single knockout construct for a particular gene. Nevertheless, following these rules for the design of targeting vectors has the advantage that success rates can be measured for each step in the vector construction and the subsequent generation of knockout clones. This will give more predictable results and may lead to further optimization of the methods presented where needed.

1.2.1. The Size and Location of the Target Arms

If the entire gene-coding region is not larger than 5 kb, the targeting vector is made by placing the arms upstream and downstream of the coding region boundaries. This will lead to the deletion of the entire coding region. If it is not possible to delete the whole gene-coding sequence (e.g., because of its large size or owing to long intron sequences), the targeting vector is made by placing the arms upstream and downstream of a coding region that encodes crucial functions of the protein. In addition, the downstream primer of the upstream arm introduces an in-frame stop codon into the gene. The size of the deletion is again limited to 5 kb. Targeted integration of this vector should lead to a null mutation as a result of the partial deletion of the coding region and the introduction of the in-frame stop codon (*see* **Fig. 2**).

The standard sizes of the 5′ and 3′ arms are 3 and 2 kb, respectively. Problems can arise through the presence of restrictions sites that preclude the cloning of the arms, the insertion of the central resistance marker cassette or the linearization of the targeting vector. If these problems are anticipated, it is first

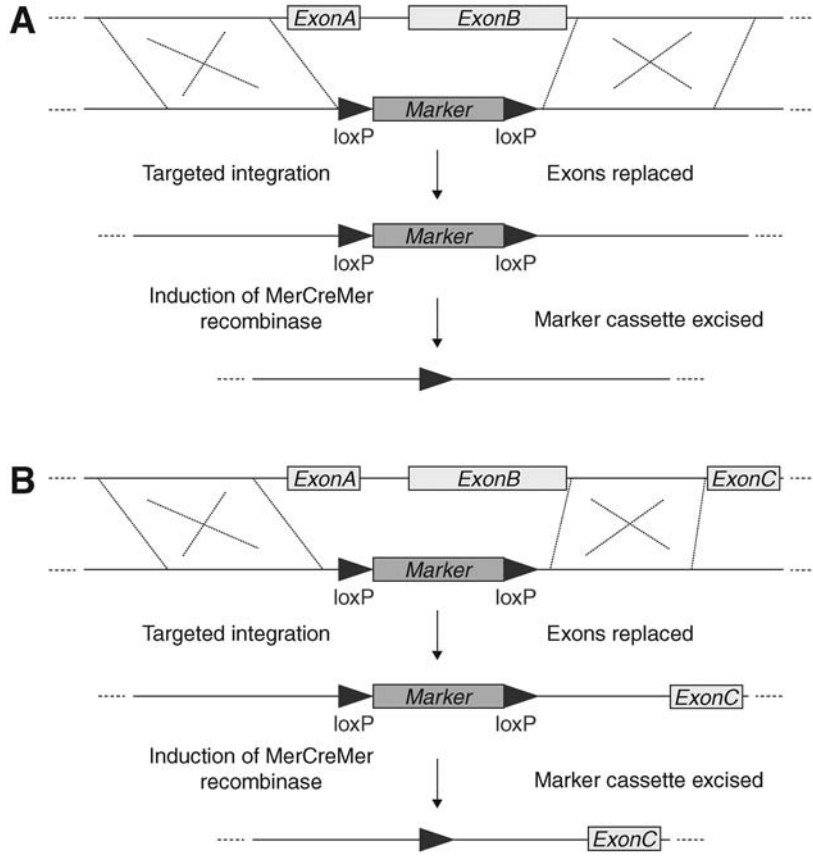


Fig. 1. **(A)** Protein X is made up of two exons. Therefore, the deletion of exons A and B is all that is required to create a knockout. **(B)** Protein X is made up of three exons. Exon C does not contribute significantly to the protein and is made up of almost completely of the 3′UTR. Therefore, the deletion of exons A and B is all that is required to create a knockout.

tried to reverse the sizes of the upstream and downstream arms. If this does not solve the problem, the arms might be shortened down to 1 kb. If this adjustment fails, the positions of the arms within the locus need to be shifted. Other problems are PCR amplification failures or difficulties to clone an arm sequence. If the failure involves an upstream arm, pairs of new upstream primers are designed 500 bp closer to the downstream primers as long as the arm sequence still exceeds 1 kb. In addition, a pair of new downstream primers is made at the next suitable position. If the failure involves a downstream arm, pairs of new downstream primers are stepwise designed 500 bp closer to the upstream

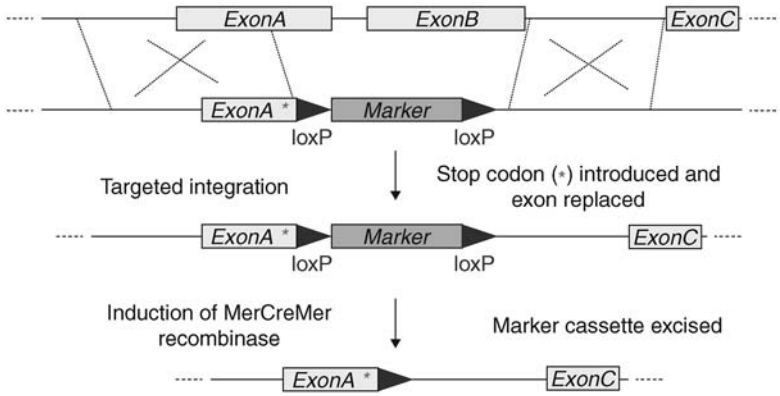


Fig. 2. Protein X's exons are too large for complete deletion targeting. Therefore, a stop codon is introduced into exon A and the critical functional domain encoded by exon A and exon B is targeted for deletion.

primers and a new pair of upstream primers is made. If this fails, the positions of the arms within the locus need to be shifted.

1.2.2. Usage of Restriction Sites

The KS (pBluescript KS+) vector is chosen for the cloning of the targeting constructs. Based on the arrangement of the restriction sites in the KS polylinker, the outside cloning site of one arm is *Xho*I and *Sal*I, and the outside cloning site of the other arm is *Spe*I and *Xba*I. *Not*I can also be used instead of *Xho*I/*Sal*I or *Spe*I/*Xba*I as an outside cloning site in case of problems. In general, both arms are first inserted into the vector, and the resistance marker is then cloned into a central *Bam*HI or *Bgl*II site. The first choice for the marker cloning is *Bam*HI. The site used for the insertion of the resistance marker needs to be unique in the construct. The *Not*I in the KS polylinker is normally used for the linearization of the construct before transfection, but *Spe*I or *Xba*I can serve as alternatives. The site used for linearization may be present more than once in the vector backbone, but it should not be present in the arm sequences or the resistance marker cassettes.

1.2.3. Primer Design

The standard approach is to add *Xho*I-*Sal*I sites to the outside primer of one arm, *Spe*I-*Xba*I sites to the outside primer of the other arm, and (stop)-*Bam*HI-*Bgl*II/*Bgl*II-*Bam*HI sites to the inside primers of each arm. In addition, all primers start with three G nucleotides to facilitate restriction digestion of the added restriction sites. If only a partial deletion of the gene-coding region is possible, the inside primer of the upstream arm adds an in-frame stop codon (*see* **Fig. 3**).

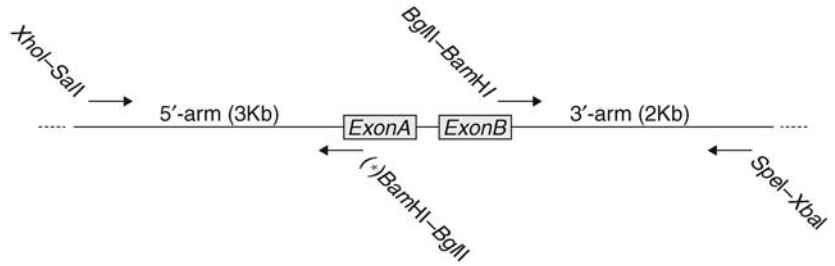


Fig. 3. Primer design.

outside primer of first arm: GGG + *Sal*I/*Xho*I
outside primer of second arm: GGG + *Spe*I/*Xba*I
inside primer of upstream arm: GGG + *Bam*HI/*Bgl*II or
GGG + *Bam*HI/*Bgl*II + in-frame stop
inside primer of downstream arm: GGG + *Bam*HI/*Bgl*II
5′ upstream arm (*Xho*I-*Sal*I)
5′ GGG CTCGAG GTCGAC 3′
3′ upstream arm (STOP-*Bam*HI-*Bgl*II)
5′ GA AGATCT GGATCC CTA 3′
5′ downstream arm (*Bgl*II-*Bam*HI)
5′ GA AGATCT GGATCC 3′
3′ downstream arm (*Spe*I-*Xba*I)
5′ GGG TCTAGA ACTAGT 3′

The 3′ end of each primer contains a 25 nucleotide sequence derived from the knockout gene locus. Only sequences with not less than 40% and not more than 70% GC content are accepted and the primer location is shifted one base at the time until this condition is met. Apart from generally accepted primer design rules known to reduce PCR artifact, try to avoid ending with "T" at the 3′ most position if at all possible.

2. Materials

2.1. Web Resources

1. *DT40 SAGE Tag Database:* http://pheasant.gsf.de/DEPARTMENT/.
2. *Marker Cassette Information:* http://pheasant.gsf.de/DEPARTMENT/dt40.html.
3. *Chicken EST and cDNA Databases:* http://www.chick.umist.ac.uk/, http://www.chickest.udel.edu/.
4. *Chicken Genome Database:* http://www.ncbi.nlm.nih.gov/projects/genome/guide/chicken/ or http://www.ensembl.org/Gallus_gallus/index.html.
5. *BLAT Search:* http://genome.ucsc.edu/.
6. *Gene Ontology Database:* http://www.geneontology.org/.
7. *BioEdit:* http://www.mbio.ncsu.edu/BioEdit/bioedit.html.

2.2. Amplification of the Arms

1. Chicken media (CM): add to 500 mL RPMI-1640 without glutamine (Gibco/BRL), 50 mL (10%) FCS (Biochrom AG), 10 mL (2%) penicillin/streptomycin solution (Gibco/BRL), 5 mL (1%) glutamin solution (Gibco/BRL), 5 mL (1%) chicken serum (Sigma), and 0.05 mL (0.1%) of 1 *M* β-mercaptoethanol solution (Sigma) (*see* **Note 1**).
2. Amplification reaction mixture (target arms PCR): 106.5 µL sterile H_2O, 15 µL cresol red, 15 µL buffer 1 (10X), 3 µL dNTP (10 m*M*), 1.5 µL polymerase mix, 3 µL primer forward (25 pmol/µL), 3 µL primer reverse (25 pmol/µL), and 3 µL genomic DNA (100 ng/µL). The mix is evenly divided among three PCR tubes.
3. Buffer 1 and polymerase mix: Expand Long Template PCR System, (Roche Applied Science).
4. Proteinase K buffer: 100 m*M* NaCl, 10m*M* Tris-HCL (pH 8.0), and 25 m*M* ethylenediamine tetra acetic acid (EDTA).
5. TE (1X): 10 m*M* Tris-HCL (pH 8.0) and 25 m*M* EDTA.
6. Expand Long Template PCR System (Roche Applied Science).
7. Cresol red (Sigma-Aldrich).
8. 10 m*M* dNTP (Fermentas).
9. Standardized insert/vector RE reaction mixture: 30 µL PCR purification product or plasmid (~100 µg/µL), 4 µL appropriate buffer, 1 µL RE, 0.4 µL bovine serum albumin (BSA), and 4.6 µL sterile H_2O.
10. Buffer: use appropriate buffer supplied with the RE(s).
11. Takara DNA Ligation Kit Ver. 2.1: 5 µL solution I, 0.5 µL prepared vector, and 4.5 µL prepared fragment (Takara Bio Inc.).
12. Chemically competent DH5α cells calcium chloride prepared.
13. Amplification reaction mixture (colony PCR): 6.3 µL sterile H_2O, 1 µL dimethyl sulfoxide (DMSO), 1 µL buffer S, 0.2 µL dNTP (10 m*M*), 0.7 µL Taq polymerase, 2 µL primer forward (5 pmol/µL), 2 µL primer reverse (5 pmol/µL), and 10 µL picked colony suspension.
14. Buffer S: 166 m*M* $(NH_4)_2SO_4$, 670 m*M* Tris-HCl (pH8.8), 67 m*M* $MgCl_2$, and 100 m*M* β-mercaptoethanol.
15. Standardized RE analysis reaction mixture: 2.5 µL plasmid prep DNA, 14.3 µL sterile H_2O, 2 µL appropriate buffer, 0.2 µL BSA, 0.5 µL enzyme 1, and 0.5 µL enzyme 2 or sterile H_2O.
16. Buffer: use appropriate buffer supplied with the RE(s).
17. Linearization mixture: 300 µL Maxiprep DNA (1 µg/µL), 51 µL sterile H_2O, 40 µL appropriate buffer, 4 µL BSA, and 5 µL *Not*I (or other enzyme).

2.3. Knockout

1. Blasticidin (stock solution 60 µg/mL) (Invitrogen).
2. Mycophenolic acid (stock solution 2 µg/mL) (Sigma-Aldrich).
3. Puromycin (stock solution 2 µg/mL) (Sigma-Aldrich).
4. K buffer (10 µL): 0.5% Tween-20 and 100 µg/mL proteinase K in 1X PCR buffer (buffer 2 is used from the Expand Long Template PCR System [Roche Applied Science]). Prepare just before use.

5. Targeting screening PCR reaction mixture: 6.3 µL sterile H_2O, 1 µL cresol red, 1 µL buffer 1, 0.2 µL dNTP, 0.07 µL polymerase mix, 0.2 µL primer forward (25 pmol/µL), 0.2 µL primer reverse (25 pmol/µL), and 1 µL DNA from the crude extract.
6. Buffer 1 and polymerase mix expand long template PCR system (Roche Applied Science).

3. Methods

3.1. Amplification of the Arms

Genomic DNA prepared from wild-type or mutant DT40 cells is used as the template to generate isogenic fragments for the targeting constructs. Isogenic arms that are identical to the sequence of the target locus were shown to increase the ratio of targeted-to-random integration after transfection of murine embryonic stem cells ***(8)***. This effect was also seen after transfection of DT40, although it is most likely less pronounced because of increased homologous recombination activity of the cells. Nevertheless, it is advisable to use genomic DNA of DT40 for the amplification of the arm sequences because this may increase the targeting ratio of more difficult genes. As DT40 is derived from an out-bred chicken, there remains the chance that one allele is targeted less efficiently than the other allele because polymorphisms cause differences between the targeting vector and the allelic locus. It should also be noted herein that chromosome 2 of DT40 exists in triplicate.

3.1.1. Genomic DNA Preparation

DT40 cells are grown in a humidified CO_2 (5%) incubator at 41°C in CM.

Day 1:

1. Centrifuge 50 mL of healthy viable DT40 cells at 1500 rpm and 4°C for 5 min. The health and viability of the cells is checked with a microscope. The cells should have a rounded shape and build clusters. This is indicative that they are healthy and are in a growing phase.
2. Wash the pellet with 1–2 mL 1X phosphate buffer solution (PBS) and centrifuge at 1500 rpm and 4°C for 5 min.
3. Resuspend the pellet in 500 µL proteinase K buffer plus 12.5 µL 20% sodium dodecyl sulfate and transfer to a 1.5-mL tube.
4. Incubate tube overnight at 56°C to extract DNA.

Day 2:

1. Add the same amount of phenol (500 µL) to the DNA extract.
2. The mix is rotated gently for 15 min and then centrifuged at 13,000 rpm and 4°C for 5min.
3. 500 µL of the upper phase are transferred to a new tube containing 500 µL phenol/chloroform (1:1).
4. The mix is rotated gently for 15 min and then centrifuged at 13,000 rpm and 4°C for 5 min.

5. 500 µL of the upper phase are transferred to a new tube containing 500 µL chloroform.
6. The solution is mixed by inverting and centrifuged at 13,000 rpm and 4°C for 5min.
7. 500 µL of the upper phase are transferred to a new tube containing 1µL RNAse A.
8. The RNA is digested for 2 h at 37°C.
9. The reaction is stopped by the addition of 50 µL of 0.5 *M* EDTA pH 8.0 and mixing by gently inverting.
10. Perform a quick spin (the DNA can be stored at this point at 4°C until dialysis).
11. The DNA is loaded into a dialysis membrane and dialyzed against ice cold 1X TE at 4°C while mixing. The 1X TE is changed after 2 h and again after 2–4 h followed by overnight dialysis.

Day 3: transfer the DNA from the membrane to a clean tube and measure the OD260 to obtain the concentration.

3.1.2. Amplification of the Target Arm Sequences

The 5′ and 3′ ends flanking the knockout region of the target gene are amplified by PCR. Use the Expand Long Template PCR System (Roche Applied Science) with cresol red, 10 m*M* dNTP, forward primer (25 p*M*), and reverse primer (25 p*M*) in the standardized amplification reaction mixture (target arms PCR). The mixture is evenly divided among three PCR tubes.

PCR program:

93°C	2 min	
93°C	10 s	35 cycles
65°C	30 s	
68°C	5 min*	*time increases: 20 s each cycle.
68°C	7 min	
4°C	∞	

Five microliter of each sample are checked and confirmed through gel electrophoresis. The selected samples are combined for PCR purification through ethanol precipitation or a commercially available kit of one's choice. If at first the PCR fails, repeat once using half the amount of primer and the addition of 1 µL DMSO (*see* **Note 2**).

3.1.3. Preparing Arms and Vector for Cloning

The target arm sequences are now cloned into the vector pKS using chemical or electro competent *Escherichia coli* cells. Sequentially, the 5′ and 3′ arms are cloned into the appropriately restricted vector. In the final step, the marker cassettes are cloned into the targeting vector between the two arms using the *Bam*HI and/or *Bgl*II site(s). The targeting of the two alleles of a gene is accomplished by using two different resistance marker cassettes that

will allow the selection of cells carrying first one and then the other resistance marker (*see* **Notes 3** and **4**).

For cloning, the purified DNA and pKS are digested with the appropriate RE overnight using the standardized insert/vector RE reaction mixture. Vectors are further alkaline phosphatase treated to block religation. The digested fragments are then isolated through agarose gel electrophoresis. The product is cut out of the gel and purified by the gel purification method of one's choice. It is found that ligation efficiency is increased and background colonies decreased when all fragments are gel isolated for cloning. Alternatively, the target arm fragments may be cloned straight from the PCR product using Invitrogen's TOPO TA Cloning kit before subcloning into the pKS targeting vector (*see* **Note 5**).

3.1.4. Ligation and Transformation

For ligation, the Takara DNA Ligation Kit is used as noted in **Subheading 2.** For transformation, use 50 µL of chemically competent DH5α cells. This is found to be quite robust, but the method used at this point is not critical.

3.1.5. Colony Screening

To screen colonies, one can either use plasmid preparation followed by RE analysis or perform colony PCR using a primer specific for the insert and one for the vector. RE analysis should still be performed on PCR-identified clones for unambiguous verification. Colony PCR is performed as defined in Wahl et al. ***(6)***.

1. Pick bacterial colonies and suspend in 50 µL sterile H_2O. After using for colony PCR, this can be stored at 4°C until selected colonies are inoculated into bacterial culture media for further use. Storage time is up to 2 wk without supplementing with bacterial culture media.
2. PCR amplify using the amplification reaction mixture.

PCR program:

95°C	10 min	
95°C	30 s	5 cycles
43°C	30 s	
72°C	3 min	
93°C	30 s	30 cycles
53°C	30 s	
72°C	3 min	
72°C	7 min	
4°C	∞	

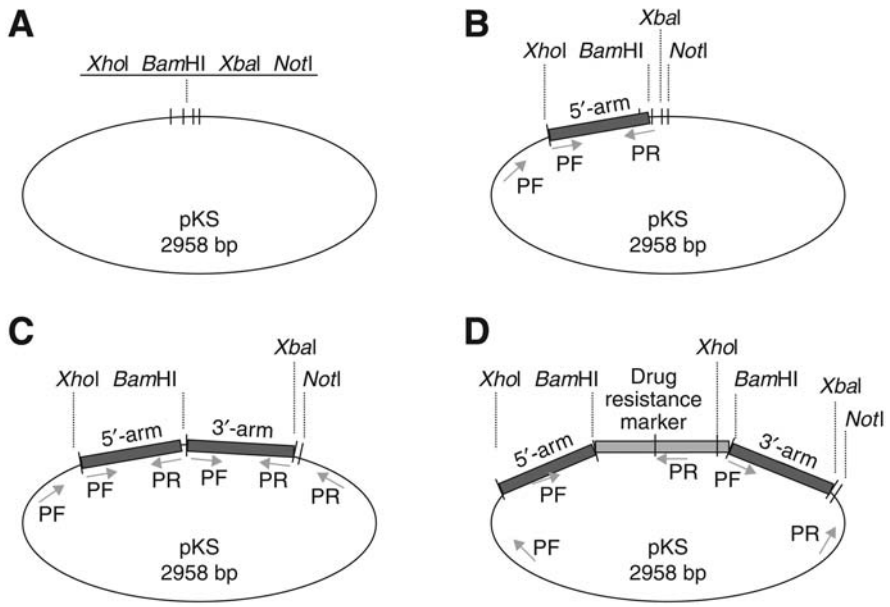


Fig. 4. Cloning steps of the construct: **(A)** The vector used contains the restriction sites *Xho*I, *Xba*I, and *Not*I in its multiple cloning site. **(B)** In the first cloning step, *Xho*I and *Bam*HI were used to insert the 5′-arm. REs or colony PCR (PF, primer forward; PR, primer reverse) can be used to identify the correct clones. **(C)** In the second cloning step, *Bam*HI and *Xba*I were used to insert the 3′-arm. Again, RE or colony PCR is used to identify the correct clones. **(D)** Finally, the drug resistance markers are inserted into the *Bam*HI site between the 5′ and 3′-arms followed by correct clone identification through RE analysis or colony PCR.

RE or primer pairs differ depending on the specific cloning step. **Figure 4** is used to graphically demonstrate the methods presented herein (clones may vary). Digest ≥1 h in the standardized RE analysis reaction mixture, add loading buffer, and run on a 0.8% agarose gel electrophoresis. To check the insertion of the 5′-arm, the plasmid is cut with *Bam*HI and with a mix of *Bam*HI and *Xho*I. After cloning of the 3′-arm, the plasmid is digested with *Bam*HI and a mix of *Bam*HI and *Xba*I. Successful cloning of the drug resistance marker cassette (BSR in this example) is verified with a digest with *Not*I, *Bam*HI, a mix of *Bam*HI and *Xho*I, and with a mix of *Bam*HI and *Xba*I. The marker cassette carries additional restriction sites. This can be used to check the direction of the insert. For more information on marker cassettes (including their sequences and how to request them) *see* http://pheasant.gsf.de/DEPARTMENT/dt40.html.

3.1.6. Plasmid Preparation and RE Analysis

Once the targeting constructs have been successfully built and confirmed, a large-scale plasmid preparation will need to be performed to obtain enough plasmid for transfection into DT40 (40 µg are needed per transfection). Using one's favorite large-scale plasmid preparation method, determine the concentration and dilute to 1µg/µL. This is the final step before transfection and it is extremely important that one be sure of the product in hand. So when in doubt, repeat RE analysis or do partial sequencing to be assured of what one is working with.

3.1.7. Linearization

1. The plasmid is linearized using a single cutter, in this case *Not*I.
2. Digest overnight at 37°C in linearization mixture.

The following day:

1. Add to the overnight digest 1 vol phenol–chloroform (400 µL). Mix well and centrifuge at 13,000 rpm for 5 min at 4°C.
2. Transfer the upper phase to a new tube containing 1 vol chloroform (400 µL). Mix well and centrifuge at 13,000 rpm for 5 min at 4°C.
3. Transfer the upper phase to a new tube containing 1 vol isopropanol (400 µL) and add 0.1 vol 3 *M* sodium acetate (40 µL). Mix well and centrifuge at 13,000 rpm for 30 min at 4°C.
4. Discard the supernatant and wash the pellet with 80% ethanol. Do not mix. Centrifuge at 13,000 rpm for 5 min at 4°C.
5. Discard the supernatant under a laminar flow hood and dry the pellet 10–20 min.
6. Add 300 µL sterile H_2O for an end concentration of 1 µg/µL and store at 4°C for a minimum of 1 h to resolve the pellet, which can be used immediately or stored at –20°C.

3.2. Knockout

3.2.1. Transfection of DT40 Cells

The construct is now ready to be transfected into DT40 to begin the knockout process.

1. Perform a cell viability count on DT40 cells grown in CM. It should be noted that some DT40 mutants have a considerably lower viability than wild-type and one should use the healthiest possible cells regardless.
2. Take 5×10^6 viable cells and centrifuge for 5 min at 1500 rpm and 4°C.
3. While centrifuging the DT40 cells prepare a 1-mL electroporation cuvet; add 40 µL of the linearized construct (concentration 1 µg/mL) and place cuvet on ice.
4. Resolve the DT40 cell pellet in 800 µL sterile 1X PBS or CM, add 800 µL to each cuvet.

5. Electroporate the cells using 25 µF and 700 V (the Gene Pulser I, II, or Xcell from Bio-Rad is recommended).
6. Transfer the electroporated cells to a 50-mL tube containing 15 mL CM.
7. Mix well and transfer 5 mL to second 50-mL tube containing 5 mL CM.
8. Aliquot the two dilutions to two labeled 96-well flat bottom microtiter plates (100 µL/well).
9. Grow overnight at 41°C.
10. Following 12–24 h of incubation time, add 100 µL CM containing the appropriate selective drug (blasticidin [60 µg/mL] or mycophenolic acid [2 µg/mL] or puromycin [2 µg/mL]) to each well to ensure that only cells carrying the resistance marker will grow. End concentration is half that of the stock solution; 30 µg/mL, 1 µg/mL, and 1 µg/mL, respectively.
11. Continue growing until colonies appear (normally 7–14 d). *Note:* some mutants may grow considerably slower and colonies may not appear until after 2 wk (*see* **Note 6**).
12. As soon as colonies are visible, pick single colonies by carefully pipeting 10 µL to a new 96-well flat bottom microtiter plate containing 300 µL CM/well. Continue picking colonies daily until all single colonies possible have been picked. The best way to see the colonies is to hold the plate up to the light and look at the bottom for wells with only single colonies. The majority of wells should have none or single colonies. To pick the colony, stick the pipet tip into the "center" of the colony and withdraw the 10 µL to be transferred to the new plate. In order to assure the picking of a single colony, one may need to perform subcloning through limited dilution on the confirmed positive clones chosen for further use.
13. Grow the picked colonies at 41°C and split and refeed CM as necessary (every 3–4 d) (*see* **Notes 7–9**).

3.2.2. Targeting Screening

To confirm homologous targeting, the grown single colonies are checked through PCR.

Crude DNA Extract:

1. Mix the cells in the picked colony 96-well flat bottom microtiter plate and transfer 200 µL to a 96-well flat bottom microtiter PCR plate. Centrifuge for 5 min at 1500 rpm and 4°C.
2. Discard the supernatant and wash with 200 µL 1X PBS. Centrifuge again for 5 min at 1500 rpm and 4°C.
3. Discard the supernatant and resolve the pellet in 10 µL K buffer. Prepare the K buffer just before use.
4. Spin down cells with a quick spin.
5. Incubate 45 min at 56°C followed by 10 min at 95°C.
6. Store at 4°C. The crude extract is stable for at least 1 wk.

PCR amplify the extract using the targeting screening reaction mixture:

93°C	2 min	
93°C	30 s	35 cycles
65°C	30 s	*time increases: 20 s each cycle
68°C	5 min*	
68°C	7 min	
4°C	∞	

Use various primer pairs to identify the targeted clones (refer to **Fig. 4**):

1. Primer forward upstream of 5′ arm and primer reverse inside the drug resistance marker cassette to amplify the marker and to exclude single crossing over.
2. Primer forward and primer reverse inside target locus to amplify the deleted region.
3. Check PCR screen through agarose gel electrophoresis.
4. Select a number of targeted clones and grow in 24-well flat bottom microtiter plates by adding 100 µL of the correct cell suspension from the 96-well flat bottom microtiter plate and add 2 mL CM (*see* **Notes 8** and **9**).
5. Select an appropriate clone and repeat the transfection procedure using a second targeting construct with a different marker cassette.

At this point, it is highly recommended to perform the phenotype experiments and consider the results in selecting specific heterologous knockout clones for further use.

4. Notes

1. Use only trusted suppliers of cell culture reagents and always test new lot numbers as well as each batch of freshly made CM.
2. The addition of DMSO is known to help with difficult spots and it is used in the sequencing protocol wherein its' inclusion increases both the quality and the length of the sequencing product. Glycerol has sometimes been used for PCR, but in the original tests that were done on it previously, it was found that additives such as Ipegal and DMSO performed better. This effect was seen with multiple polymerases including Taq, TaqGold, Vent, MMLV-RT, and Superscript RT.
3. Marker recycling:
   a. The marker cassette can be removed from the knockout clones to exclude side effects of the drug resistance marker and to reuse the marker.
   b. Cells with viability over 80% are diluted to a concentration of 0.6×10^6 cells in 2 mL.
   c. Add 100 µL 4-tetrahydroxytamoxifen (40 µ*M*) to the 2 mL of cells (final concentration 2 µ*M*) and incubate for 5–12 h at 41°C. The concentration and timing of this step will have to be empirically determined based on the targeted locus.

A thorough reading of Arakawa et al. ***(2)*** is highly recommended before attempting 4-HT induction.

d. Proceed with subcloning.

4. *Drug check after marker recycling or transfection:* drug check is carried out for three purposes: first, to ensure the excision of the marker cassette after marker recycling, and second, to eliminate single targeted cells in a mix of targeted and nontargeted cells after the second transfection, and third, to confirm that the second transfection did not target the already knocked out locus thereby excising the first targeting event.
   a. Pick single colonies from the subcloning and transfer to a 96-well flat bottom microtiter plate containing 300 µL CM/well.
   b. Transfer 75 µL of the cell suspension to serial wells containing 75 µL of the drug(s) to be checked: blasticidin (60 µg/mL) or mycophenolic acid (2 µg/mL) or puromycin (2 µg/mL). End concentration is half that of the stock solution; 30 µg/mL, 1 µg/mL, and 1 µg/mL, respectively.
   c. Let grow for 3–4 d at 41°C.
   d. Choose clones positive/negative for blasticidin, mycophenolic acid, or puromycin as expected.
   e. Cultivate selected clones for further experiments.
5. This may facilitate the cloning of particularly difficult fragments. In using this approach, it is recommended to include the addition of a step to add the 3′ A-overhang, as the long range PCR uses a proofreading enzyme.
   a. Method: 8 µL PCR product, 1 µL 10X PCR buffer, 0.1 µL dATP(10 µ*M*), 0.1 µL Taq polymerase at 72°C for 10 min and then proceed according to the kits instructions. The target arms are then isolated from the prepared TA TOPO kit plasmid through RE as shown above.
6. Drug check following second transfection.
   a. To check transfectants after the second transfection for the two drug-resistance marker cassettes, the cells are cultivated in both drugs simultaneously.
   b. Mix 500 µL of each drug in a 24-well flat bottom microtiter plate and incubate with 100 µL cell suspension.
   c. Let grow at 41°C for 3–5 d.
7. Freezing positive clones:
   a. Positive clones are transferred to flasks and grown in 50 mL CM for 2 d.
   b. The cells are transferred to a 50-mL centrifuge tube and centrifuged for 5 min at 1500 rpm and 4°C.
   c. The supernatant is discarded and the pellet resolved in 10 mL freezing medium containing DMSO.
   d. The cells should now be immediately transferred to 10 labeled cryovials and frozen at –80°C. After 24 h (and up to 1 wk) the cells need to be transferred to a liquid nitrogen tank for long-term storage.
   e. Freezing medium: 70 mL (70%) CM, 20 mL (20%) FCS, and 10 mL (10%) DMSO.

8. Thawing cells:
   a. To thaw the cells, place the vial for 5 min in the 41°C incubator, then spin down the tube for 5 min at 1500 rpm and 4°C.
   b. Remove the freeze medium completely, resolve the pellet in 1 mL CM, and transfer to a flask containing 25 mL CM.
   c. Let grow for 2–3 d at 41°C and check the cells condition every day under the microscope.
9. Subcloning by limited dilution:
   a. Count the viable cells using Trypan blue.
   b. Prepare three tubes containing 10 mL CM each and add 1000, 300, and 100 cells, respectively.
   c. Plate each tube to a 96-well flat bottom microtiter plate by pipeting to each well 100 µL (transfer three plates with 10 cells/well, 3 cells/well, and 1 cell/well). Alternatively, cells can be distributed across two 96-well flat bottom microtiter plates in a 300, 100, 30, 10, 3, and 1 cells/well configuration using a third of the plate for each dilution.
   d. Incubate the plates for 8 d without changing medium.
   e. Subclones should be visible by then as round colonies. Pick single colonies into 300 µL CM.

References

1. Buerstedde, J. M. and Takeda, S. (1991) Increased ratio of targeted to random integration after transfection of chicken B cell lines. *Cell* **67,** 179–188.
2. Arakawa, H., Lodygin, D., and Buerstedde, J. M. (2001) Mutant loxP vectors for selectable marker recycle and conditional knockouts. *BMC Biotechnol.* **1,** 7.
3. International Chicken Genome Sequencing Consortium (2004) Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature* **432,** 695–716.
4. International Chicken Polymorphism Map Consortium (2004) A genetic variation map for chicken with 2.8 million single-nucleotide polymorphisms. *Nature* **432(7018),** 717–722.
5. Schmid, M., Nanda, I., Hoehn, H., et al. (2005) Second report on chicken genes and chromosomes 2005. *Cytogenet. Genome Res.* **109(4),** 415–479.
6. Wahl, M. B., Caldwell, R. B., Kierzek, A. M., et al. (2004) Evaluation of the chicken transcriptome by SAGE of B cells and the DT40 cell line. *BMC Genomics* **5,** 98.
7. Caldwell, R. B., Kierzek, A. M., Arakawa, H., et al. (2005) Full-length cDNAs from chicken bursal lymphocytes to facilitate gene function analysis. *Genome Biol.* **6,** R6.
8. Deng, C. and Capecchi, M. R. (1992) Reexamination of gene targeting frequency as a function of the extent of homology between the targeting vector and the target locus. *Mol. Cell Biol.* **12(8),** 3365–3371.

12

Design and Application of a *shRNA*-Based Gene Replacement Retrovirus

Rugang Zhang, Peter D. Adams, and Xiaofen Ye

Summary

To perform structure/function analyses of a protein in vivo, ideally one should be able to simultaneously abolish expression of the endogenous wild-type protein, substitute it with a form of the protein containing a targeted mutation, and analyze the functional consequences. Until recently, this was a highly challenging and/or laborious approach in mammalian systems, requiring a targeted gene knockin in a human cell line or mouse. Herein is described a RNA interference (RNAi)-based approach to achieve this much more simply in mammalian cells. A single retrovirus has been constructed, which directs expression of a short hairpin RNA (shRNA) to knockdown expression of the endogenous protein of interest; a cDNA coding for a wild-type or mutant version of the same protein that also contains "silent mutations" that do not affect the protein sequence, but do make the mRNA resistant to the shRNA; and a puromycin-resistance gene to allow rapid drug selection of the virus-infected cells. Using this virus, expression of the endogenous Anti-Silencing Function 1a (ASF1a) histone chaperone has been efficiently replaced in primary human cells, by an ectopically expressed epitope-tagged version. Moreover, the virus is designed so that other shRNA and shRNA-resistant cDNA cassettes can easily be substituted, making the approach readily applicable to other protein targets.

Key Words: Gene replacement; retrovirus; shRNA; silent mutations; shRNA-resistant cDNA; U6 promoter.

1. Introduction

RNAi-based technologies have revolutionized the molecular and cellular approaches taken to understand biological processes in mammalian cells. For example, when studying gene function at the cellular level, researchers can use RNAi technology to quickly generate cells lacking the gene of interest and examine its "loss-of-function" phenotype ***(1)***. Furthermore, it is becoming feasible to perform RNAi-based "genetic screens" in mammalian cells for gene products whose inactivation confers a specific cellular phenotype ***(2–6)***.

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

Going one step beyond simple knockdown of gene expression, one can use RNAi technology to knockdown expression of an endogenous protein and simultaneously replace it with a mutant version of the same protein. This is analogous to a gene "knockin" experiment, and has great potential for defining functional domains of proteins using physiological assays, without interference from the wild-type endogenous protein. Herein, a single retrovirus has been described that encodes a RNAi to knockdown the endogenous protein and a RNAi-resistant mRNA that, in turn, codes for a wild-type or mutant version of the same protein. The mRNA is made RNAi-resistant by introducing silent mutations into the redundant positions of each codon, so that they do not affect the amino acid sequence. The virus is designed so that the individual shRNA and cDNA-expression cassettes corresponding to any gene of interest can be readily subcloned into the virus.

In this chapter, it is assumed that the reader has identified a gene of interest and the cDNA(s) encoding the mutant or mutants of interest. First, how to generate a functional shRNA that knocks down the target protein is described. For this purpose, the polymerase chain reaction (PCR)-shagging approach of Hannon and coworkers is used *(7)*. However, an entry vector for the shRNA, called pPUR V2 has been custom designed. Once the shRNA-expression cassette is subcloned into this vector, it can be readily transfected into a transfectable human cell line, the transfected cells selected in puromycin can then be assayed for knockdown of the target protein. Functional shRNAs are then easily subcloned from pPUR V2 into the retrovirus plasmid. The retrovirus is modified from the pQCXI series of vectors (Clontech: Mountain View, CA). The vector has been designed with the capacity to direct expression of the shRNA subcloned from pPUR V2, a cDNA coding for a shRNA-resistant mRNA, and a gene-encoding resistance to puromycin. The retrovirus plasmid is packaged into infectious retrovirus using Phoenix cells and then delivered to the target cells by a standard virus infection.

2. Materials

1. pPUR V2—this cloning vector was modified from pPUR (Clontech). An approx 750-bp *Bam*HI/*Eco*RI fragment was excised from pPUR and replaced by a synthetic oligonucleotide linker containing the multiple cloning site (MCS) in **Fig. 1**. The linker was synthesized with *Bgl*II and *Mfe*I sticky ends, which are compatible with *Bam*HI and *Eco*RI, respectively, but meaning that the original *Bam*HI and *Eco*RI sites are destroyed by the ligation. This plasmid is available on request.
2. pGEM1-U6 plasmid (a gift of Greg Hannon *[7]*). This plasmid contains the human U6 promoter, a promoter that directs expression by RNA polymerase III.
3. 5 U/μL Taq polymerase (Invitrogen).

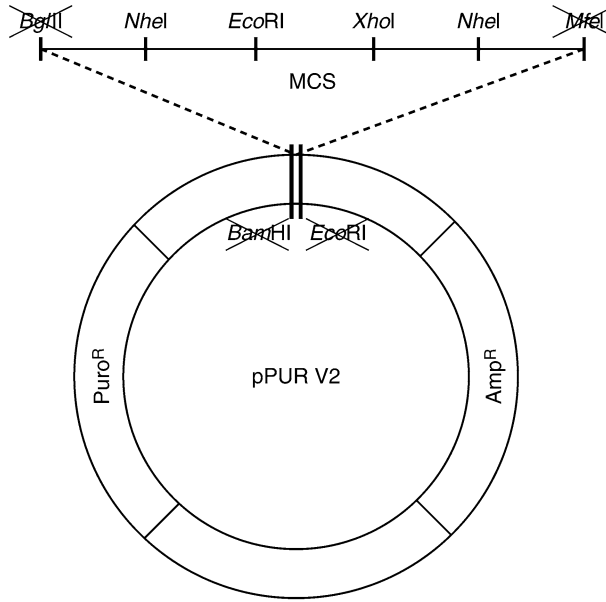


Fig. 1. Restriction map of pPUR V2 vector. The original *Bam*HI and *Eco*RI sites in pPUR have been destroyed. The U6-shRNA cassette, generated by PCR, is subcloned into the *Eco*RI and *Xho*I sites of pPUR V2.

4. 2 U/μL Deep vent DNA polymerase (New England Biolabs [NEB: Ipswich, MA]).
5. Qiaquick PCR purification kit (Qiagen).
6. Restriction enzymes and buffers: *Eco*RI and *Xho*I (NEB).
7. 10X Tris-Acetate-EDTA (TAE) agarose gel-loading dye (150 m*M* ethylenediaminetetraacetic acid, 30% glycerol, 0.25% [w/v] bromophenol blue [Sigma, St. Louis, MA], and 0.25% [w/v] xylene cyanol FF [Sigma]).
8. 1% UltraPure agarose (Invitrogen).
9. Low melting temperature agarose (SeaPlaque Agarose, Cambrex: Walkersville, MD).
10. T4 DNA ligase (Roche: Basel, Switzerland).
11. DH5α competent cells (Invitrogen).
12. Qiafilter plasmid Maxi kit (Qiagen: Valencia, CA).
13. Laemmli sample buffer (50 m*M* Tris-HCl, 2% [w/v] sodium dodecyl sulfate [SDS], 100 m*M* dithiothreitol, 10% [v/v] glycerol, and 0.05% [w/v] bromophenol blue, pH 6.8).
14. Equipment and reagents for SDS-polyacrylamide gel electrophoresis (PAGE).
15. Bradford reagent: Bio-rad, Hercules, CA and 1 mg/mL bovine serum albumin as standard.
16. Polyvinylidene (PVDF) protein transfer membrane (Bio-Rad: Hercules, CA).
17. Towbin transfer buffer (170 m*M* glycine, 22 m*M* Tris-HCl, and 0.01% [w/v] SDS, pH 8.3).

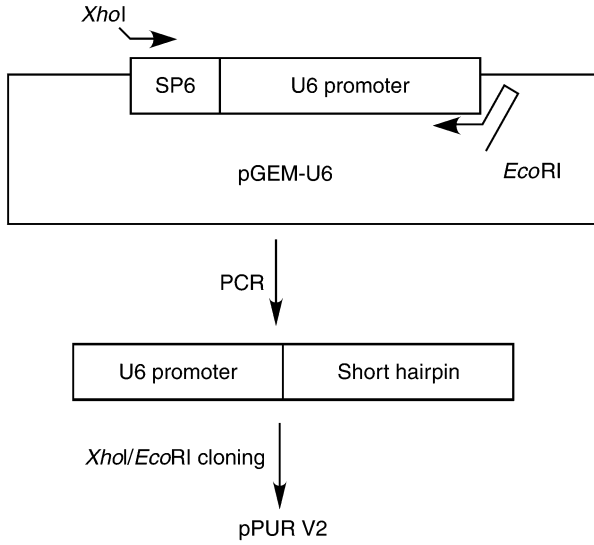


Fig. 2. Generation of U6-shRNA-expression cassette by PCR. The U6-shRNA-expression cassette is generated by PCR using plasmid pGEM-U6 as a template; a forward primer that contains a *Xho*I site and anneals to an SP6 site in the plasmid; a reverse primer that contains an *Eco*RI site, the shRNA sequence, and anneals to the 3′-end of the U6 promoter. The PCR product is digested with *Xho*I and *Eco*RI and subcloned into pPUR V2 cut with the same two enzymes.

18. Sterile-filtered, 1 mg/mL puromycin in phosphate-buffered saline, pH 7.3 (Clontech).
19. U2OS cells grown in Dulbecco's Modified Eagles (DME) supplemented with 10% (v/v) fetal bovine serum (FBS) in a humidified 37°C, 10% (v/v) CO_2 incubator.
20. Luria Bertani media + 100 µg/mL ampicilin (Sigma).
21. Phoenix cells were a gift by Gary Nolan, and WI38 primary human fibroblast cells were purchased from ATCC: Manassas, VA. Phoenix cells should be grown in DME supplemented with 10% (v/v) FBS in a humidified 37°C, 5% (v/v) CO_2 incubator. WI38 cells should be grown in Dulbecco's modified Eagle's medium supplemented with 20% (v/v) FBS, essential and nonessential amino acids, and vitamins (Cellgro: Herndon, VA) in a humidified 37°C, 5% (v/v) CO_2 incubator.
22. 8 mg/mL Polybrene in ddH_2O (Sigma).
23. 0.45-µm Filter (Fisher: Pittsburgh, PA).
24. ddH_2O.

3. Methods

3.1. Generation and Validation of Functional shRNAs in pPUR V2

The shRNAs are generated by PCR, as fusions to the human U6 RNA polymerase III promoter **(Fig. 2)**. It is recommended to first construct 3–6 shRNAs per gene and identify those that efficiently knockdown the target in a

simple transfection-based assay. Typically, it is found that 20–30% of sequence-verified shRNAs efficiently knockdown their target. Therefore, if 3–6 are tested, at least one should knockdown its target.

To generate shRNAs, the U6-shRNA fusion is synthesized by PCR as described in **section 3.1.1.–3.1.7.** and subcloned into pPUR V2 as an *Eco*RI/*Xho*I fragment. This plasmid is transfected into cells that express the target gene (typically U2OS osteosarcoma cells, because they are readily transfectable), the transfected cells are selected for 48 h in 1 µg/mL puromycin and then target knockdown is assayed by protein Western blot and/or reverse transcriptase (RT)-PCR of the mRNA. Plasmid pPUR V2 was designed for this purpose **(Fig. 1)**. In this plasmid, selection of transfected cells with puromycin is rapid and the custom-designed MCS allows functional shRNAs to be easily shuttled from pPUR V2 into a unique *Nhe*I (New England Biolabs, Ipswich, MA) site in the 3′-long terminal repeat (LTR) of the retrovirus (pQCXIP, pQCXIH, or pQCXIN [Clontech]).

1. *Design and order PCR primers:* the reverse primer encoding the shRNA is designed using "RNAi Central" at http://katahdin.cshl.org:9331/RNAi_web/scripts/main2.pl. Click on "shRNA design." Using the radio buttons and drop-down menu, select three "29-mer design sense–antisense" and three "29-mer antisense– sense." Make sure that the accession numbers or sequences that match cDNA or exon sequences are entered. The website generates sequences of oligos encoding the shRNA and an *Eco*RI site for subcloning into pPUR V2. A GCGC sequence should be added to the 5′-end of the oligo to facilitate digestion by *Eco*RI. For each target gene, reverse primers containing the shRNA sequence should be synthesized at 0.05-µmol scale by Sigma-Genosys (The Woodlands, TX) or elsewhere. In addition, the forward PCR primer that is common to all shRNAs and contains a *Xho*I site and a SP6 primer should be synthesized: 5′-GGCCCTCGAGGATTTAGGTGACACTATAG-3′. Additional information on shRNA design is at RNAi Central.
2. *Perform PCR to generate the U6-shRNA cassette:* this is schematized in **Fig. 2**. As a PCR template, the pGEM1-U6 plasmid containing the human U6 RNA polymerase III promoter is used. Set up the PCR reaction, as follows:

pGEM1-U6	1 ng
50 m*M* $MgCl_2$	2.5 µL
2.5 m*M* Deoxynucleotide 5′-triphosphate	4 µL
10X Taq buffer	5 µL
40% Dimethyl sulfoxide	5 µL
50 µ*M* SP6	1 µL
50 µ*M* Hairpin primer	1 µL
ddH_2O	30 µL
Taq DNA polymerase	1 µL
Deep vent DNA polymerase	0.2 µL
Final volume	50 µL

Perform PCR cycles, as follows: 95°C for 3 min; 30 cycles of: 95°C for 30 s, 55°C for 30 s, and 72°C for 1 min; and one cycle of: 72°C for 10 min. Confirm the PCR reaction by 1% TAE agarose gel electrophoresis ***(8)***. A single PCR product of approx 600 bp should be apparent.

3. Purify the PCR product using the Qiaquick PCR purification kit, according to manufacture's instructions. Collect the purified PCR product in 30 µL of ddH_2O.
4. *Digest the PCR product as follows:*

PCR product (3–5 µg DNA)	30 µL
10X NEB buffer 3	4 µL
*Eco*RI	1.5 µL
*Xho*I	1.5 µL
ddH_2O	3 µL, to make a final volume of 40 µL

Incubate for 2 h at 37°C and then add 4 L of 10X TAE-loading buffer. For the vector, digest 3 µg of pPUR V2 in the same way.

5. Purify both DNA fragments by TAE agarose electrophoresis in low-melting point agarose (1% agarose for the PCR product and 0.6% agarose for pPUR V2). Excise the DNA bands. Bands containing DNA can be stored at –20°C at this point.
6. *Perform ligations as follows:* melt the agarose gel slices at 75°C and set up ligation reactions as follows:

Vector DNA (µL)	1	1
Insert DNA (µL)	0	3
ddH_2O (µL)	9	6

Melt the agarose/water mix at 75°C for 4 min. Place at 37°C for 4 min. During this time, make a T4 ligase/buffer master mix. For each ligation reaction: 7 µL of water, 2 µL of 10X T4 DNA ligase buffer (Roche), and 1 µL of T4 DNA ligase (Roche). Add 10 µL of reaction mix to each ligation reaction. Stir gently with pipet tip. Place at room temperature for 2 h to overnight. Melt ligation at 75°C and transform 1 µL into DH5α competent cells.

7. *Restriction digest screening and sequencing of shRNAs:* verify that the ligations worked, based on the number of colonies obtained. Ideally, ligation of cut pPUR V2 alone should generate zero colonies and pPUR V2 with the U6-shRNA insert should generate 10 or 100 s of colonies. Inoculate three to five colonies from each shRNA ligation reaction into 5 mL of Luria Bertani media + ampicilin and grow overnight at 37°C with shaking. Purify plasmid DNA with an Eppendorf perfect-prep mini-prep kit (Eppendorf, Hamburg, Germany), according to the manufacturer's instructions. Verify the clones by restriction digest with *Xho*I and *Eco*RI, which should release an approx 600-bp fragment. Confirm those plasmids with the correct size, insert by direct sequencing using the primer AATTTCTTGGGTAGTTTGCAG, which anneals to the human U6 promoter and directs sequencing into the shRNA.
8. *Transfection of pPUR V2-shRNA into U2OS cells:* transfection quality DNA of each sequence-verified pPUR V2-U6-shRNA plasmid is made using a Qiafilter

Fig. 3. Design of an shRNA-resistant ASF1a. The figure shows the region of the wild-type endogenous ASF1a mRNA that is targeted by the shRNA **(Line 2)**; the corresponding amino acid sequence **(Line 1)**; and the ectopically expressed ASF1a mRNA that does not change the protein sequence, but which is resistant to the shRNA by virtue of base changes in the third position of each codon.

plasmid Maxi kit, according to the manufacturer's instructions. Use each of these purified plasmids to transfect a 60% confluent 10-cm plate of U2OS cells by the calcium phosphate method ***(9)***. In addition, transfect one plate with pPUR V2 vector alone and one plate with no DNA. Add puromycin to a final concentration of 1 µg/mL, 48 h after transfection (*see* **Note 1**). After 72–96 h, scrape the cells into Laemmli sample buffer, determine the protein concentration by Bradford assay and fractionate 50–100 µg of total cellular protein by SDS-PAGE ***(10)*** (*see* **Note 2**).

9. *Western blotting to test for protein knockdown:* transfer the proteins from the gel to a PVDF membrane and immunoblot to detect the protein of interest using a standard protocol ***(10)***. The extract derived from pPUR V2-transfected cells serves as a positive control and the efficiency of knockdown is measured relative to this. If antibodies to the protein of interest are not available, knockdown can be assayed by quantitative real-time RT-PCR.

3.2. Design of shRNA-Resistant cDNA

To design an shRNA-resistant cDNA, silent mutations are introduced in the cDNA in the region targeted by the shRNA **(Fig. 3)**. These mutations change the nucleotide sequence, but do not affect the encoded protein sequence. For most codons, this means changing the third base of the codon. In other cases, more or less flexibility is allowed. Consult the full genetic code for details (e.g., in the New England Biolabs catalog). Mutations are introduced by a standard mutagenesis protocol, for example, two-step mutagenic PCR or with the Strategene Quickchange kit (La Jolla, CA) ***(8)***.

3.3. Subcloning shRNA and cDNA Into Retrovirus

A modified version was constructed of pQCXIN (Clontech, http://orders.clontech.com/clontech/techinfo/vectors/vectorsM-Q/pQCXIN.shtml) **(Fig. 4)**. The final vector encodes the ASF1a shRNA, under control of the U6 promoter; a puromycin-resistance gene, under control of a CMV promoter; and an ASF1a cDNA that is resistant to the ASF1a shRNA, under control of the same CMV promoter and an internal ribosomal entry site.

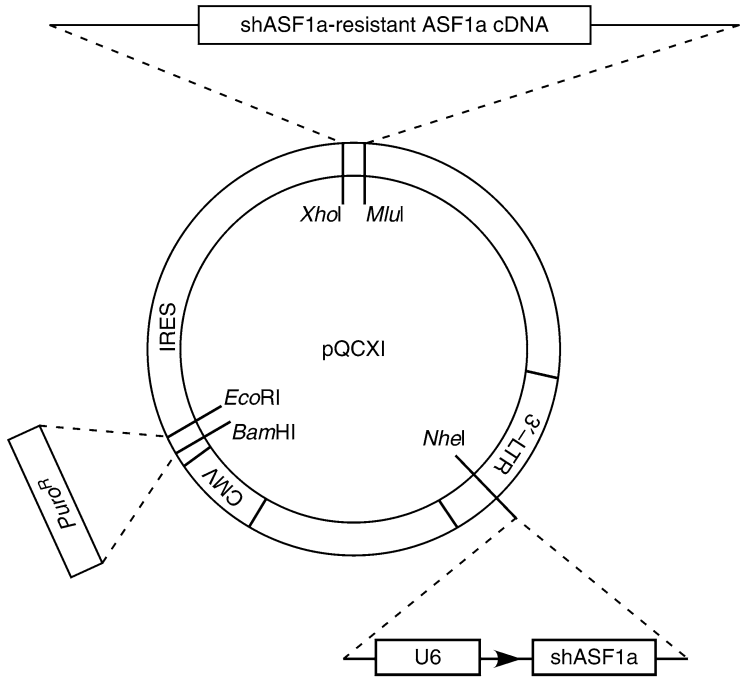


Fig. 4. Schematic map of modified pQCXIP plasmid. This retrovirus plasmid is designed for simultaneous expression of both shASF1a, an shASF1a-resistant wild-type or mutant ASF1a cDNA, and resistance to puromycin under control of the indicated promoters. *See* text for details.

This vector was made as follows. First a puromycin-resistance gene was inserted into the MCS of pQCXIN as a *Bam*HI/*Eco*RI fragment. This fragment was generated by PCR using pQCXIP as a template and PCR primers containing a *Bam*HI site (forward primer, 5′-end of gene) and an *Eco*RI site (reverse primer, 3′-end of gene). Second, using a PCR-based approach, the neomycin-resistance gene downstream of the internal ribosomal entry site in pQCXIN was removed and replaced with a shRNA-resistant ASF1a cDNA flanked by a unique *Xho*I site at the 5′-end and a unique *Mlu*I site at the 3′-end. The cDNA codes for an hemagglutinin (HA)-tagged form of ASF1a. This cDNA can be excised with *Xho*I and *Mlu*I and replaced by another cDNA of choice. Third, the U6-shASF1a cassette was excised from pPUR V2 as a *Nhe*I fragment and inserted into the unique *Nhe*I site in the 3′-LTR of the virus plasmid. This vector was verified by sequencing and is available on request.

3.4. Infection of Cells

The infectious retrovirus is generated by transfecting the plasmid DNA into a packaging cell line, for example, Phoenix cells (http://www.stanford.edu/group/nolan/protocols/pro_helper_dep.html). Transfection quality DNA of the retrovirus plasmid is made using a Qiafilter plasmid Maxi kit, according to the manufacturer's instructions. After transfection, the Phoenix cells reverse transcribe the plasmid DNA into an RNA that is packaged into infectious virus and expelled from the cells. Then, the tissue culture supernatant containing the virus is applied to the cells of interest, to deliver the shRNA and the shRNA-resistant cDNA to the cells in a single virus. The infected cells can be selected in puromycin to enrich for the infected cells. To assess the efficiency of killing of uninfected cells by puromycin, perform a mock virus infection. To assess the efficiency of infection, infect one plate with a virus known to generate good titer (e.g., vector pQCXIP, Clontech).

1. The day before transfection, plate 5×10^6 Phoenix cells in 10 mL of medium in a 10-cm dish. Culture in a 37°C, 5% (v/v) CO_2 incubator overnight.
2. Remove the medium from the 10-cm dish 4 h before transfection, and replace with 9 mL of prewarmed fresh medium.
3. Dilute the required amount of 2.5 *M* $CaCl_2$ to 250 m*M* and aliquot 0.5 mL per transfection to separate sterile 15-mL polystyrene tubes.
4. Add supercoiled plasmid DNA of the intended virus, to a total of 30 µg per tube.
5. Add 0.5 mL of 2X BES Buffered Saline (BBS), by dripping slowly from a 1-mL pipet, vertically down the center of the tube (1–2 drops per second). Do not mix. Wait 15 min. At this time, the precipitate should be barely visible to the naked eye.
6. Use a 1-mL pipet to blow air bubbles through the solution to mix the precipitate. Distribute the mixture drop-wise into the medium, evenly over the plate of Phoenix cells.
7. Rock the plates back and forth very gently to mix the calcium phosphate precipitate and then place in humidified 37°C incubator with 5% (v/v) CO_2 overnight.
8. Remove the medium and replace with 6 mL of fresh medium 24 h after transfection and return to the humidified 37°C incubator with 5% (v/v) CO_2.
9. On the same day, split the target WI38 primary fibroblast cells in 10 mL of medium in a 10-cm plate. The cell density the next day should be about 50% confluent.
10. Harvest the virus containing supernatant 24 h after **step 8**, and then filter through a 0.45-µm filter.
11. Remove 10-mL medium from the target WI38 cell plate, and replace with 5 mL of fresh WI38 cell medium. Add 5 mL of virus containing supernatant from Phoenix cells dropwise into the target WI38 cell plate.
12. Add polybrene to each plate of WI38 cells to a final concentration of 8 µg/mL. Mix polybrene into the medium by gently shaking the plate. Put the cell plate back into a 5% (v/v) CO_2-containing incubator.

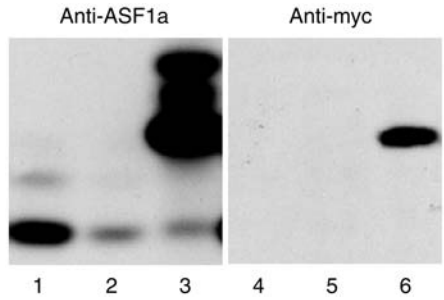


Fig. 5. Replacement of endogenous ASF1a by ectopically expressed HA-tagged ASF1a. W138 cells were infected with mock (1,4) virus encoding shRNA to ASF1a (2,5), or virus encoding shRNA to ASF1a and a *myc*-tagged shRNA-resistant form of ASF1a (3,6). Cell extracts were Western blotted with anti-ASF1a or anti-myc antibodies.

13. Remove the medium from the WI38 cells 24 h postinfection and replace with 10 mL of fresh WI38 medium, containing puromycin at a final concentration of 3 μg/mL.
14. Typically, 3 d after addition of puromycin, all of the noninfected cells should be dead. At this time, there should be no surviving cells left on the mock virus infected plate.
15. Harvest the infected cells for Western blot analysis by scraping directly in 1X Laemmli sample buffer, followed by boiling for 4 min. Samples can be stored frozen at −80°C.

3.5. Testing Protein Knockdown and Ectopic Expression

Cell extracts should be fractionated by SDS-PAGE, transferred to a PVDF membrane and then Western blotted with antibodies to the protein of interest and with anti-HA antibodies (or an appropriate epitope tag) ***(10)***. During SDS-PAGE, it should be possible to resolve the HA-tagged ectopically expressed protein from the untagged endogenous protein, making it possible to detect knockdown of the endogenous protein and its replacement by the ectopically expressed HA-tagged protein of higher molecular weight. This is illustrated for ASF1a in **Fig. 5**.

Once knockdown of the endogenous protein and expression of the ectopic protein are confirmed, the cells can be assayed for the functional consequences. The assays herein will obviously be specific to each protein and researcher. However, regardless of the assay, the following control viruses should also be used: a virus that knocks down the endogenous protein, but does not direct expression of an ectopic protein; a virus that knocks down the endogenous protein, and directs expression of the HA-tagged wild-type protein.

4. Notes

1. About 1 μg/mL Puromycin should kill all of the untransfected cells within 48 h. The efficiency of transfection and cell killing should be determined from the plates

transfected with pPUR V2 only and no DNA, respectively. Approx 20–30% of the cells on the plate should be transfected and the doubling time of U2OS cells is about 24 h. Therefore, after 48 h of drug selection the plate transfected with pPUR V2 alone should be about 60% confluent. There should be no live cells remaining on the plate transfected without DNA.

2. A Bradford assay can be performed on samples in Laemmli sample buffer, provided that not more than 1 µL of Laemmli buffer is added per 700 µL Bradford reaction. Also, add 1 µL of Laemmli sample buffer to the reference Bradford reactions, containing 0, 5, and 10 µg of bovine serum albumin as standards.

Acknowledgments

This work was supported by National Institute of Health R01 GM062281 and the Leukemia and Lymphoma Society, Peter D. Adams (PDA) and the American Federation of Aging Research, Rugang Zhang (RZ).

References

1. Hannon, G. J. and Rossi, J. J. (2004) Unlocking the potential of the human genome with RNA interference. *Nature* **431,** 371–378.
2. Paddison, P. J., Silva, J. M., Conklin, D. S., et al. (2004) A resource for large-scale RNA-interference-based screens in mammals. *Nature* **428,** 427–431.
3. Westbrook, T. F., Martin, E. S., Schlabach, M. R., et al. (2005) A genetic screen for candidate tumor suppressors identifies REST. *Cell* **121,** 837–848.
4. Berns, K., Hijmans, E. M., Mullenders, J., et al. (2004) A large-scale RNAi screen in human cells identifies new components of the p53 pathway. *Nature* **428,** 431–437.
5. Nicke, B., Bastien, J., Khanna, S. J., et al. (2005) Involvement of MINK, a Ste20 family kinase, in Ras oncogene-induced growth arrest in human ovarian surface epithelial cells. *Mol. Cell* **20,** 673–685.
6. Kolfschoten, I. G., van Leeuwen, B., Berns, K., et al. (2005) A genetic screen identifies PITX1 as a suppressor of RAS activity and tumorigenicity. *Cell* **121,** 849–858.
7. Paddison, P. J., Caudy, A. A., Bernstein, E., Hannon, G. J., and Conklin, D. S. (2002) Short hairpin RNAs (shRNAs) induce sequence-specific silencing in mammalian cells. *Genes Dev.* **16,** 948–958.
8. Sambrook, J. and Russell, D. (Ed.) (2001) *Molecular cloning: A laboratory manual. 3rd ed.*, Cold Spring Harbor Laboratory Press, NY.
9. Adams, P. D., Lopez, P., Sellers, W. R., and Kaelin, W. G., Jr. (1997) Fluorescence-activated cell sorting of transfected cells. *Methods Enzymol.* **283,** 59–72.
10. Harlow, E. and Lane, D. (Ed.) (1988) *Antibodies: A laboratory manual.* Cold Spring Harbor Laboratory Press, NY.

13

Construction of Simple and Efficient DNA Vector-Based Short Hairpin RNA Expression Systems for Specific Gene Silencing in Mammalian Cells

Tsung-Lin Cheng and Wen-Tsan Chang

Summary

RNA interference (RNAi) is an evolutionarily conserved mechanism of posttranscriptional gene silencing induced by introducing the double-stranded RNAs (dsRNAs) into cells. Recent progress in RNAi-based gene-silencing techniques has revolutionarily advanced in studies of the functional genomics and molecular therapeutics. Among the widely used dsRNAs including exogenously synthetic and endogenously expressed small interfering RNAs (siRNAs) and short hairpin RNAs (shRNAs), the shRNAs are more efficient than siRNAs on the induction of gene silencing and currently have evolved as an extremely powerful and the most popular gene silencing reagent. The DNA vector-based shRNA-expression systems provide not only a simple and effective way in inhibiting gene activities in either inheritable or inducible manner, but also a cost-effective tool in constructing the expression vectors. To fully explore the DNA vector-based shRNA-expression systems in RNAi-mediated gene-silencing techniques, four distinct RNA polymerase III (Pol III)-controlled type III promoter-based expression vectors are constructed including pHsH1, pHsU6, pMmH1, and pMmU6, which contain either the RNase P RNA H1 (H1) or small nuclear RNA U6 (U6) promoter from human and mouse. Moreover, to improve the constructing and screening efficiency for the shRNA-expression recombinant clones, these four DNA vectors are further reconstructed by inserting a stuffer of puromycin resistance gene ($Puro^R$) between restriction enzyme *Cla*I and *Hin*dIII sites, which makes the preparation of vectors easy and simple for cloning the shRNA-expression sequences. Because of the ease, speed, and cost efficiency, these four improved DNA vector-based shRNA-expression vectors provide a simple, convenient, and efficient gene-silencing system for analyzing specific gene functions in mammalian cells. Herein, the simple and practical procedures for the construction of DNA vector-based expression vectors, potential and rational design rules for the selection of effective RNAi-targeting sequences, efficient and cost-effective cloning strategies for the construction of shRNA-expression cassettes, and effective and functional activity assays for the evaluation of expressed shRNAs are described.

Key Words: DNA vector-based RNAi system; gene silencing; RNA interference (RNAi); RNA polymerase III (Pol III) promoter; RNase P RNA H1 promoter (H1); small nuclear RNA U6 promoter (U6); short hairpin RNA (shRNA); small interfering RNA (siRNA).

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

1. Introduction

RNA interference (RNAi) is a mechanism of posttranscriptional gene silencing in which double-stranded RNAs (dsRNAs) induce sequence-specific cleavage of the homologous RNA transcripts and in turn cause complete degradation of the aberrant RNA fragments, resulting in reduced or loss of activities of the genes ***(1,2)***. During the processes of RNAi-mediated gene silencing, the dsRNAs are first recognized and cleaved into 21–23-nucleotide (nt) small interfering RNAs (siRNA) duplexes, with symmetrical 2-nt 3′-overhangs by dsRNA-specific RNase III-related endonuclease, Dicer ***(3,4)***. The resulting siRNAs are efficiently incorporated into the RNA-induced silencing complex (RISC) to form a ribonucleoprotein complex that first mediates the unwinding of the siRNA duplexes and selectively degrades the sense strand of siRNA. The single-stranded siRNA-coupled (RISC) is in turn guided to catalyze the endonucleolytic cleavage of homologous RNA transcripts at the site where the antisense strand of siRNAs is complementarily bound ***(5,6)***. Subsequently, the resulting disruptive RNA fragments are immediately subjected to exonucleolytic destruction by the action of exoribonuclease.

RNAi is evolutionarily conserved to each of the eukaryotic organisms involved in regulation of the gene activity. The function of RNAi, primarily, appears to be implicated in cellular defense mechanism in antiviral infection and maintaining genomic integrity against transposable element-induced genomic instability ***(7,8)***, as well as in cellular gene regulation and chromosomal epigenetic control ***(9–11)***. Currently, it has emerged as a practically used strategy for reverse functional genomics and in particular as an extremely powerful approach for molecular therapeutics ***(12–15)***. In plants and invertebrates, introduction of the dsRNAs into the cells induces sequence-specific inhibition of homologous gene expression. However, in mammals, the dsRNAs longer than 30 nt in length trigger a strong cytotoxic response through activation of the dsRNA-dependent protein kinase and 2′,5′-oligoadenylate synthetase, resulting in inactivation of the eukaryotic initiation factor-2α and activation of the RNaseL, and in turn causing general inhibition of protein synthesis and nonspecific degradation of single-stranded RNA, respectively ***(16–20)***. However, by using short synthetic 21-nt siRNAs with symmetrical 2-nt 3′-overhangs allow for inducing the sequence-specific gene silencing, yet avoid triggering the nonselective cytotoxic effects by long dsRNAs ***(21,22)***.

In mammals, there are mainly two strategies in producing dsRNAs by exogenous delivery of synthetic siRNAs ***(21,22)*** or short hairpin RNAs (shRNAs) ***(23)*** and endogenous vector-expressed siRNAs ***(24–26)*** or shRNAs ***(27–31)***. The silencing effect induced by synthetic dsRNAs is transient and the target gene is reactive after a few days, as well as the cost of chemical synthesis of RNA

oligonucleotides is expensive ***(21–23)***. In contrast, the inhibition effect triggered by vector-expressed dsRNAs can be easily manipulated in either inheritable or inducible manner, and in particular construction of the expression vectors requires only DNA oligonucleotides that can be easily obtained from local commercial suppliers ***(27,32,33)***. The endogenous dsRNAs including siRNAs and shRNAs can be transcribed from either RNA Pol II-or Pol III-regulated promoters; however, the primary RNA transcripts derived from RNA Pol II promoters are subjected to posttranscriptional processes, including 5′-capping and 3′-polyadenylation. In addition, the RNA Pol II promoters require specific transcription terminator sequences that make it difficult to predefine the size of mature RNA products. Whereas, the RNA Pol III-regulated type III promoters, especially H1 or U6 from human and mouse, have been used most frequently, because they have a well-defined transcription start site and a simple and effective transcription terminator sequence consisting of only five or six consecutive thymidine residues (Ts), and therefore these H1 and U6 promoters are suitable for the synthesis of small RNA transcripts with defined sizes. Moreover, RNA Pol III promoters can efficiently transcribe small RNA transcripts lacking both the 5′-cap and 3′-polyadenosine (poly[A]) tail ***(34–36)***.

In practice, the siRNA-expression vectors utilize dual promoter strategy in which two RNA Pol III promoters align in either tandem or convergent manner (*see* **Fig. 1B**,**C**). The two tandem promoters drive independently the expression of sense and antisense RNAs from two separated transcriptional units ***(24,25)***, whereas the two convergent promoters drive simultaneously the expression of complementary sense and antisense RNAs from a single DNA fragment ***(26)***. In contrast, the shRNA-expression vectors contain a single RNA Pol III promoter followed by the sense, a loop, and the antisense sequences (*see* **Fig. 1A**) ***(27–31)***. In addition, previous studies have reported that both the siRNA-expression systems do not appear to work as efficiently as the shRNA-expression system to inhibit gene expression ***(33,37,38)***. To develop convenient and effective DNA vectors for simple and efficient cloning of small-RNA expression sequences, four distinct expression vectors including pHsH1, pHsU6, pMmH1, and pMmU6, which contain the widely used RNA Pol III promoters H1 and U6 from human and mouse, are constructed ***(38)***. In particular, these four expression cassettes are designed in which the small-RNA expression sequences are cloned between two unique restriction enzyme *Cla*I and *Hin*dIII sites. Moreover, to facilitate the cloning of small-RNA expression sequences into these four expression cassettes, these four expression vectors are further improved on by constructing a stuffer of $Puro^R$ between *Cla*I and *Hin*dIII sites. These improved expression vectors can be used directly for mammalian gene function analysis in vitro cultured cells or in vivo whole

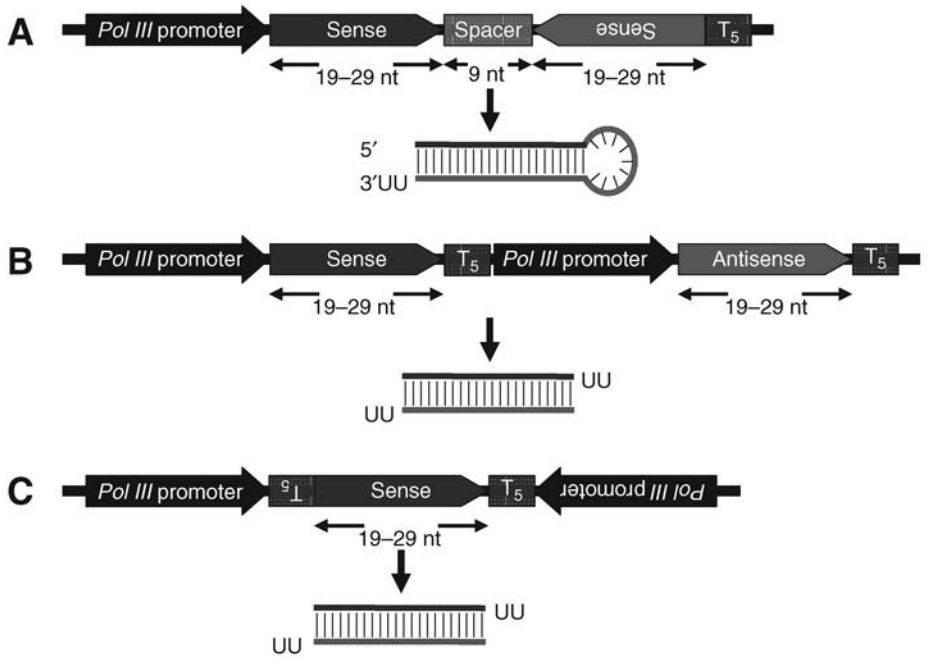


Fig. 1. Structures of shRNA and siRNA-expression systems. **(A)** Construct of RNA Pol III-controlled promoter-based shRNA-expression vector. In this system, a single RNA Pol III promoter drives the expression of a shRNA transcript in which the sense and antisense strands of the shRNA are linked by a 9-nt loop sequence. The connection of five consecutive Ts (T_5) at the 3′-end of the cassette provides not only a transcription termination signal for RNA Pol III but also a 2-nt uridine overhang at the 3′-terminus of shRNA transcript. **(B)** Construct of two tandem array RNA Pol III-controlled promoters-based siRNA-expression vector. In this system, two tandem array RNA Pol III promoter-based expression cassettes are used to drive separately the expression of the sense and antisense RNA transcripts. Both the sense and antisense RNAs then anneal to form a ds-siRNA. **(C)** Construct of two convergent array RNA Pol III-controlled promoters-based siRNA-expression vector. In this system, two convergent array RNA Pol III promoters-based expression unit is used to drive simultaneously the transcription of the sense and antisense strands. Both the complementary sense and antisense strands then anneal to form a ds-siRNA.

animals. In this chapter, the discussion is focused on these improved DNA vector-based shRNA-expression systems used in this laboratory. The protocols described in this chapter provide a comprehensive procedure for constructing the simple and efficient shRNA-expression systems for specific gene silencing in mammalian cells.

2. Materials

2.1. Cell Culture

1. Mammalian cell lines of interest (American Type Culture Collection, Manassas, VA) stored in liquid nitrogen or at −80°C.
2. Cell line-specific growth media (GIBCO-BRL, Rockville, MD) supplemented with or without the heat inactivated various percentages of fetal calf serum (Biological Industries, Ashrat, Israel) and 1% antibiotic/antimycotic solution (GIBCO-BRL), and stored at 4°C.
3. Phosphate-buffered saline (PBS): 2.7 m*M* KCl, 1.8 m*M* KH_2PO_4, 136 m*M* Nacl, 10 m*M* Na_2HPO_4, pH 7.4; stored at room temperature.
4. 0.25% Trypsin solution (GIBCO-BRL) and 1 m*M* ethylenediamine tetraacetic acid (EDTA) (GIBCO-BRL) stored in aliquots at −20°C.
5. Cell scrapers and spatulas (Techno Plastic Products AG, Trasadingen, Switzerland).

2.2. Plasmid Vectors

1. pHsH1, pHsU6, pMmH1, pMmU6, pHsH1puro, pHsU6puro, pMmH1puro, and pMmU6puro expression vectors (*see* **Figs. 2** and **3**) stored in aliquots at −30°C.
2. pGEM-7ZF(+) vector (Promega, Madison, WI) stored at −30°C.
3. pMSCVpuro vector (BD Biosciences Clontech, Palo Alto, CA) stored at −30°C.
4. Competent cells of *Escherichia coli* strain XL 1-blue (Stratagene, La Jolla, CA) stored in aliquots at −80°C.
5. Luria-Bertani (LB) broth stored at room temperature.
6. Ampicilin stock solution (100 mg/mL) stored in aliquots at −30°C.
7. *Cla*I, *Eco*RI, and *Hin*dIII restriction enzymes (Promega) and T4 DNA ligase (Promega) stored at −30°C.
8. Agarose gel (Promega) stored at room temperature.
9. 50X Tris-acetate stock solution stored at room temperature.
10. Gel-loading buffer (6X): 0.25% bromophenol blue, 0.25% xylene cyanol FF, and 15% Ficoll type 400; stored at room temperature.
11. Plasmid mini and maxi purification kits (Viogene, Sunnyvale, CA), as well as gel extraction and polymerase chain reaction (PCR) purification kits (Viogene) stored at room temperature.
12. Pheno/chloroform/isoamyl alcohol (25/24/1) and chloroform/isoamyl alcohol (24/1) stored at 4°C and room temperature, respectively.
13. 3 *M* Sodium acetate, pH 4.8, stored at room temperature.
14. Ethanol (100% and 70% [v/v]) stored at −30°C.
15. Tris-EDT buffer (TE) 10 m*M* Tris-HCl and 1 m*M* EDTA, pH 8.0; stored at room temperature.
16. ABI PRISM® BigDye™ terminator cycle sequencing ready reaction kits with AmpliTaq DNA polymerase (Applied Biosystems, Foster, CA) stored at −30°C.

2.3. Polymerase Chain Reaction

1. Oligonucleotide primers (T7 promoter: 5′-TAATACGACTCACTATAGGG-3′; SP6 promoter: 5′-GATTTAGGTGACACTATAG-3′) stored at −30°C.

Fig. 2. Sequences and structures of RNA Pol III-controlled type III promoter-based shRNA-expression vectors. The human H1 (HsH1) and U6 (HsU6), and mouse H1 (MmH1) and U6 (MmU6) promoters are isolated from human and mouse genomic DNAs by PCR amplification, and cloned into the pGEM-7ZF(+) vector. The resulting DNA constructs are designated as pHsH1 **(A)**, pHsU6 **(B)**, pMmH1 **(C)**, and pMmU6 **(D)** vectors. These four DNA vectors all contain the same unique restriction enzyme *Cla*I and *Hin*dIII sites for cloning the shRNA-coding sequences. In these four DNA vectors, the proximal sequence element is in white and shaded in blue, TATA box is in bold and shaded in green, restriction enzyme sites of *Eco*RI (GAATTC), *Cla*I (ATCGAT), and *Hin*dIII (AAGCTT) are underlined and in purple and bold, and G is the transcription initiation site (+1).

2. 10 m*M* Deoxynucleoside triphosphate mixtures (Promega) stored at −30°C.
3. PCR reagents, including Taq DNA polymerase and 10X reaction buffer with $MgCl_2$ (Promega) stored at −30°C.

2.4. Transfection and Functional Assessments

1. Lipofectamine 2000™ (Invitrogen, Carlsbad, CA) stored at 4°C.
2. TRI Reagent™ (Molecular Research Center, Cincinnati, OH) stored at 4°C.
3. Protein lysis buffer: 50 m*M* NaCl, 50 m*M* Tris-HCl, 2 m*M* EDTA, 0.5% sodium deoxycholate, 1% NP-40 (Roche Molecular Biochemicals, Mannheim, Germany), and 0.1% SDS, pH 7.4, stored at room temperature.
4. Protease inhibitors (Roche) stored in aliquots at −80°C.

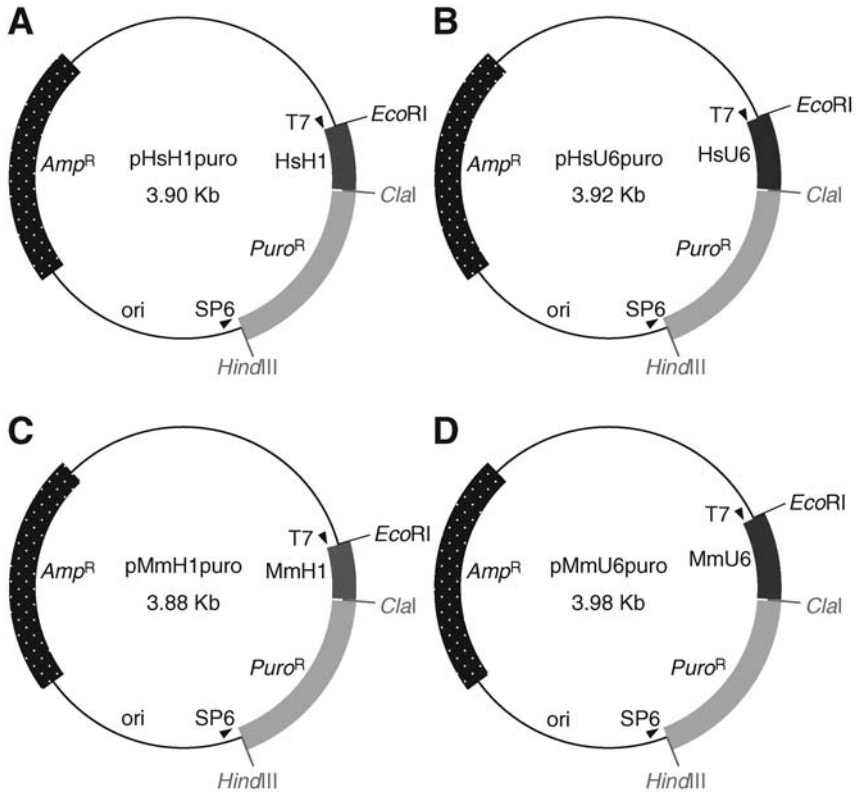


Fig. 3. Constructs of improved DNA vector-based shRNA-expression systems. The *Puro*^{R} is cloned between restriction enzyme *Cla*I and *Hin*dIII sites in the pHsH1, pHsU6, pMmH1, and pMmU6 constructs as a stuffer that is convenient for the cloning of the shRNA-coding sequences. The resulting improved DNA constructs are redesignated as pHsH1puro **(A)**, pHsU6puro **(B)**, pMmH1puro **(C)**, and pMmU6puro **(D)** vectors.

5. 3% Paraformaldehyde (Merck, Darmstadt, Germany).
6. 0.5% Triton X-100 (Merck, Darmstadt, Germany).
7. Bicinchoninic acid assay (Pierce, Rockford, IL) stored at room temperature.
8. Bovine serum albumin (Sigma, St Louis, MO) stored at room temperature.
9. Dual-luciferase reporter assay system (Promega) stored in aliquots at –80°C.
10. Enhanced chemiluminescence Western blotting detection reagents (Amersham Biosciences, Arlington Heights, IL) stored at 4°C.

2.5. Instruments

1. Microcentrifuges (Heraeus Biofuge Pico and Heraeus Biofuge Fresco, Kendro Laboratory Products, Sollentum, Germany).
2. Dri-block heater (Techne DRI-BLOCK DB 20, Techne, Cambridge, UK).

3. Handheld ultraviolet (UV) lamp (VL-4.L, Vilber Lourmat, Marne-la-Vallee, France).
4. UV image system (UV illuminator, Vilber Lourmat, Marne-la-Vallee, France).
5. Spectrophotometer (Beckman DU 640, Beckman Instruments, Fullerton, CA).
6. Microplate reader (Dynatech MR5000, Dynatech Laboratories, Chantilly, VA).
7. Luminometer (MiniLumat LB 9506, EG&G Berthold, Wildbach, Germany).
8. Sodium dodecyl sulfate-polyacrylamide gel electrophoresis apparatus (Mighty Small II 8×7 cm^2, Hoefer Scientific Instruments, San Francisco, CA).
9. Electrophoresis power supply (EPS 1000, Amersham Pharmacia Biotech, Uppsala, Sweden).
10. Semidry transfer apparatus (Semiphor transphor unit, Amersham Pharmacia Biotech).
11. Automated DNA sequencer (ABI PRISM 377 DNA sequencer, Applied Biosystems, Foster, CA).

3. Methods

The methods described in this section outline (1) the structural features and construction of improved shRNA-expression vectors, (2) the molecular characteristics of designed and selected RNAi-targeting sequences, (3) the procedures for cloning shRNA-expression vectors, and (4) the approaches for assessing gene silencing efficiency by shRNA-expression vectors.

3.1. Structural Features and Construction of Improved shRNA-Expression Vectors

3.1.1. Structural Features of Improved shRNA-Expression Vectors

The functional active siRNA, either in vivo identified or in vitro synthesized, is a small 21–23-nt RNA duplex with symmetrical 2-nt 3′-overhangs ***(3,4)***. In addition, the long dsRNAs stimulate a serious cytotoxic response through activation of the dsRNA-dependent protein kinase and 2′,5′-oligoadenylate synthetase in mammalian cells ***(16–20)***. However, this nonselective cytotoxic effect can be overcome by directly applying small dsRNAs with the size smaller than 30-nt in length, including synthetic or expressed siRNAs and shRNAs. The RNA Pol III-regulated type III promoters, especially H1 and U6 from human and mouse, have been used most frequently, because they transcribe the RNA from a defined start site (+1) and terminate at a run of 5–6 Ts. As well as, they can efficiently express small-RNA transcripts without posttranscriptional modification including 5′-capping and 3′-polyadenylation ***(34–36)***. Thus, these promoters are suitable for expression of the defined small-RNA transcripts with the features fulfilling the aforementioned criteria.

In addition, to make the construction of shRNA-expression vectors simple and convenient, all the vectors are constructed to contain the same unique cloning sites, *Cla*I and *Hin*dIII, for cloning the RNAi-targeting sequences (*see* **Note 1**). Because U6 promoter transcribes preferentially from a “G” nucleotide at the +1

position, whereas the H1 promoter is less strict. The shRNA-expression vectors are designed particularly that RNA transcripts start with a nucleotide G in the vectors, where it locates within the restriction enzyme *Cla*I site (*see* **Fig. 2**). Specifically selected RNAi-targeting sequences can be easily cloned into an expression cassette, providing an optimal system for testing endogenous expression and activity of shRNA. However, one big obstacle for DNA vector-based RNAi systems is that it takes much time and effort to clone the DNA constructs. To enhance the convenience of constructing a DNA vector-based RNAi system and facilitate the screening of recombinant clones, all the vectors are further improved by inserting a stuffer of *Puro*^{R} between the unique cloning sites, *Cla*I and *Hin*dIII, which makes the preparation of the DNA vectors simple and easy by only removing the stuffer of *Puro*^{R} DNA fragment with *Cla*I and *Hin*dIII double digestion (*see* **Fig. 3**) **(Note 2)** ***(38)***.

3.1.2. Construction of Improved shRNA-Expression Vectors

The shRNA-expression vectors, including pHsH1, pHsU6, pMmH1, and pMmU6 (*see* **Fig. 2**), are constructed by PCR-based cloning method. The RNA Pol III-regulated type III promoters, including H1 and U6 from human (Hs) and mouse (Mm), are amplified by standard PCR reaction using synthetic oligonucleotides, which are purchased from local commercial suppliers (*see* **Note 3**). The oligonucleotides used for amplification of the HsH1, HsU6, MmH1, and MmU6 are:

HsH1-S: 5′-GGAATTCGAACGCTGACGTCATCAAC-3′ and HsH1-AS: 5′-CCATCGATAAAGAGTGGTCTCATACAG-3′; HsU6-S: 5′-GGAATTCAAGGTCGGGC AGG AAGAGG-3′ and HsU6-AS: 5′-CCCAAGCTTCCATCGATGTTTCGTCCTTTCCACAAGATAT-3′; MmH1-S: 5′-GGAATTCCGCTCTTGAAGGACGACGTCATC-3′ and MmH1-AS: 5′-CCATCGATAGGGTGTAGACCGGCCGCCAC-3′; MmU6-S: 5′-GGAATTCATCCGACGCCGCCATCTCTAGG-3′ and MmU6-AS: 5′-CCATCGATCAAGGCTTTTCTCCAAGGGATA-3′.

To simplify the construction procedures, the amplification product of HsU6 promoter is first treated with *Eco*RI and *Hin*dIII restriction enzymes, then cloned into an *Eco*RI/*Hin*dIII-digested pGEM-7ZF(+) vector (*see* **Note 4**), and the resulting plasmid is designated as pHsU6. Subsequently, the other amplification products including HsH1, MmH1, and MmU6 promoters are treated with *Eco*RI and *Cla*I restriction enzymes, subcloned into an *Eco*RI/*Cla*I-digested pHsU6 vector to substitute the HsU6 promoter, and the resulting plasmids are called as pHsH1, pMmH1, and pMmU6. To construct the improved cloning vectors, including pHsH1puro, pHsU6puro, pMmH1puro, and pMmU6puro (*see* **Fig. 3**), a *Cla*I/*Hin*dIII-treated *Puro*^{R} DNA fragment isolated from

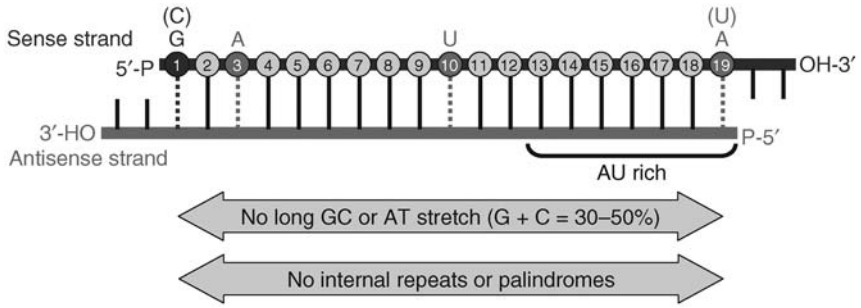


Fig. 4. Sequence-specific features for the rational design of potential siRNAs. The mature siRNA is a 21–23-nt dsRNA that contains a 19-nt duplexed region, symmetrical 2-nt 3′-overhangs, and 5′-phosphate (P) and 3′-hydroxyl (OH) groups. The positions of each nucleotide in the 19-nt duplexed region of the sense strand are numbered. On the basis of recently established design rules, an effective siRNA has high stability at the 5′-terminus of the sense strand, and lower stability at the 5′-antisense terminus and at the cleavage site. In addition, the sequence-specific preferences at the following positions on the sense strand are important including the presence of a G (C) at position 1, an A at position 3, an U at position 10, and an A (U) at position 19.

pMSCVpuro vector is inserted into the *Cla*I/*Hin*dIII-digested pHsH1, pHsU6, pMmH1, and pMmU6 vectors.

3.2. Molecular Characteristics of Designed and Selected RNAi-Targeting Sequences

The efficiency of RNAi-based gene silencing is primarily dependent on the effectiveness and specificity of the RNAi-targeting sequences. To obtain the effective siRNAs, it is necessary to design, synthesize, and screen several different RNAi-targeting sequences from a particular gene. Systematic analyses of the specific features from the effective siRNAs reveal that siRNA might have sequence-specific characteristics associated with its functionality. These molecular characteristics generally include low-to-medium G/C content (30–50%), high internal stability at the sense strand 5′-terminus, low internal stability at the sense strand 3′-terminus, absence of internal repeats or palindromes, and base preferences at the sense strand positions 1, 3, 10, and 19 (*see* **Fig. 4**) ***(39–43)***.

1. Retrieve the nucleotide sequence of any gene from the National Center for Biotechnology Information (NCBI) nucleotide database (GenBank; http://www.ncbi.nlm.nih.gov/).
2. Screen any 19-nt sequence (*see* **Note 5**) within the coding region and 3′-untranslated region that fulfills the aforementioned sequence-specific characteristics and in particular does not contain stretches of four or more consecutive As and Ts.
3. Select any 19-nt sequence containing more than three mismatches to any other gene and also avoid any known single nucleotide polymorphisms by searching the

nonredundant NCBI database (http://www.ncbi.nlm.nih.gov/BLAST/) with the screened sequence.
4. Choose particularly two to four 19-nt sequences with a G/C and an A/T at the sense strand positions 1 and 19, respectively.
5. Design the sense and antisense oligonucleotides: shGene-S: 5′-CGNNNNNNNNNNNNNNNNNttcaagagaNNNNNNNNNNNNNNNNNNCTTTTTGGAAA-3′ and shGene-AS: 5′-AGCTTTTCCAAAAAGNNNNNNNNNNNNNNNNNNtctcttgaaNNNNNNNNNNNNNNNNNN-3′ (*see* **Note 3**).

3.3. Molecular Construction of shRNA-Expression Vectors

This subsection describes the molecular cloning of the shRNA-expression vectors that can efficiently induce inhibition of target-gene expression in a sequence-specific manner. The construction procedures use only standard molecular cloning techniques, which simply involve inserting an annealed oligonucleotide duplex into the *Cla*I/*Hin*dIII restriction enzyme sites in the improved shRNA-expression vectors. The following experimental steps discuss the key components of this procedure, including (1) preparation of the shRNA-expression vectors, (2) preparation of the shRNA-expression templates, (3) cloning of the gene-specific shRNA-expression vectors, (4) screening of the shRNA-expression template positive clones, and (5) sequencing of the shRNA-expression template sequences (*see* **Fig. 5**).

3.3.1. Preparation of the shRNA-Expression Vectors

1. Digest 10 µg of pHsH1puro, pHsU6puro, pMmH1puro, or pMmU6puro in a 1.5-mL Eppendorf tube in a reaction with 5 µL of 10X restriction enzyme buffer, 10 U of *Cla*I and *Hin*dIII, and distilled H_2O to total 50 µL in 37°C water bath for 2 h.
2. Analyze 1 µL of digested DNA mixtures on a 0.8% (w/v) agarose gel with an appropriate molecular weight marker.
3. Inactivate the restriction enzymes by incubation at 70°C heat block for 10 min.
4. Isolate the digested vector by using electrophoresis on a 0.8% (w/v) agarose gel.
5. Recover the DNA fragment from the agarose gel by using the gel extraction kit, and elute the DNA fragment with 50 µL of TE (pH 8.0) (*see* **Fig. 5**).

3.3.2. Preparation of the shRNA-Expression Templates

1. Mix 5 µL of the complementary oligonucleotides (100 µ*M*) in a 1.5-mL Eppendorf tube in a reaction with 2 µL of 10X annealing buffer (T4 DNA ligase ligation buffer) and distilled H_2O to total 20 µL (*see* **Note 6**).
2. Place the Eppendorf tube in a 95°C heat block for 10 min.
3. Remove the Eppendorf tube from the heat block and allow to cool to room temperature on the bench.
4. Centrifuge briefly the Eppendorf tube to recover the reaction solution and store on ice or at 4°C until ready to use (*see* **Note 7**).

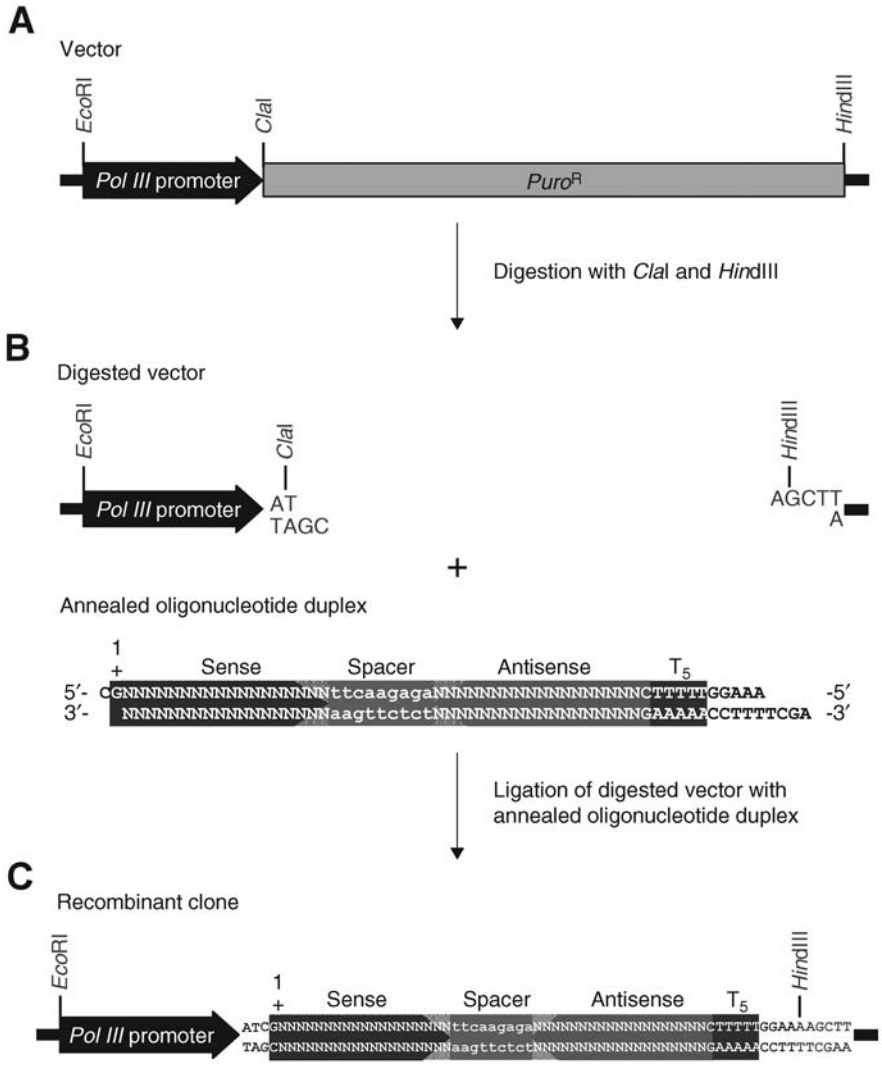


Fig. 5. Experimental procedure for constructing the DNA vector-based shRNA-expression cassette. **(A)** Preparation of DNA vector-based shRNA-expression vector. The improved DNA vector-based shRNA-expression vector is digested with restriction enzymes *Cla*I and *Hin*dIII, simultaneously, to remove the stuffer *Puro*^{R} DNA fragment. **(B)** Cloning of shRNA-expression cassette. The *Cla*I/*Hin*dIII-digested shRNA-expression vector is ligated with an annealed oligonucleotide duplex that contains a specific shRNA-expression sequence with a row of five Ts as a transcription termination signal and two unique restriction enzyme *Cla*I and *Hin*dIII compatible ends. **(C)** Screening of shRNA expressed recombinant DNA clone. The DNA construct containing the shRNA-expression sequence is identified by simply mapping with restriction enzymes *Cla*I and *Hin*dIII, and further confirmed by directly DNA sequencing with oligonucleotide primer against T7 or SP6 promoter. The positive recombinant clones contain the restriction enzyme *Hin*dIII site but usually lose the restriction enzyme *Cla*I site.

3.3.3. Cloning of the Gene-Specific shRNA-Expression Vectors

1. Mix 2 μL of *Cla*I/*Hin*dIII-digested vectors and 8 μL of annealed shRNA-coding DNA templates in a 1.5-mL Eppendorf tube in a reaction with 2 μL of 10X ligation buffer and distilled H_2O to total 19 μL (*see* **Note 8**).
2. Add 1 μL of T4 DNA ligase.
3. Incubate in 16°C water bath overnight.
4. Transform 200 μL of XL 1-blue competent cells with 20 μL of ligated mixtures.
5. Plate on LB agar plates containing 100 μg/mL of ampicillin.
6. Incubate in 37°C incubator overnight.

3.3.4. Screening of the shRNA-Expression Template Positive Clones

1. Inoculate four selected colonies into 3 mL LB broth containing 100 μg/mL of ampicillin (*see* **Note 9**).
2. Incubate in 37°C incubator overnight.
3. Purify plasmid DNAs from 1.5 mL overnight culture by using plasmid mini purification kit, and elute the plasmid DNAs with 50 μL of TE (pH 8.0).
4. Check isolated plasmid DNAs by single digestion with restriction enzyme *Cla*I or *Hin*dIII. Digest 2 μL of purified plasmid DNA in a 1.5-mL Eppendorf tube in a reaction with 2 μL of 10X restriction enzyme buffer, 2 U of *Cla*I or *Hin*dIII, and distilled H_2O to total 20 μL in 37°C water bath for 1 h.
5. Analyze 10 μL of digested DNAs on a 0.8% (w/v) agarose gel with an appropriate molecular weight marker. The positive shRNA-expression clones containing restriction enzyme *Hin*dIII site but usually losing restriction enzyme *Cla*I site are digested only with *Hin*dIII and not digested with *Cla*I. Plasmids showing this restriction enzyme-digestion pattern are presumably correct and should be confirmed by directly sequencing.

3.3.5. Sequencing of the shRNA-Expression Templates

Plasmid DNA is sequenced by using an automated DNA sequencer, which uses the dideoxy sequencing method with fluorescent dyes.

1. Set up cycle sequencing reaction: 500 ng plasmid DNA, 3.2 pmol of T7 or SP6 promoter primer, 8 μL ABI Prism dGTP BigDye terminator, and distilled H_2O to total 20 μL.
2. Perform the PCR reaction by using the following thermocycling parameters:

Step	Time	Temperature (°C)	Cycles
Initial denaturation	2 min	94	1
Denaturation	30 s	96	–
Annealing	15 s	50	25
Extension	4 min	60	–

3. Amplify plasmid DNA containing the correct sequence by using plasmid maxi purification kit, and elute the plasmid DNAs with 500 μL of TE (pH 8.0).

3.4. Functional Assessment of shRNA-Expression Vectors in Mammalian Cells

Tremendous evidence has already shown that not all of the RNAi-targeting sequences selected from a particular gene exhibit the same potencies on inducing gene silencing. Only a limited number of trigger siRNAs are capable of inducing highly efficient target gene silencing in a sequence-specific manner. The silencing efficacy of siRNAs is dependent on the specificity of the target sites within the gene and can only be determined experimentally based on the inhibition of the target-gene expression. Several widely used approaches can be used to analyze the efficiency of gene silencing induced by DNA vector-based shRNA expression, including (1) Northern blot, (2) quantitative reverse transcription (RT)-PCR, (3) Western blot, (4) immunostaining, and (5) functional activity assay (*see* **Fig. 6**). In general, the effect of gene silencing can be detected 24–48 h after transfection, dependent on the abundance and the stability of the proteins encoded by the target genes.

3.4.1. Transfection of shRNA-Expression Vectors

1. Subculture and plate 1×10^5 cells per well in 2 mL growth medium onto a six-well culture plate 24 h before transfection. For immunostaining, cells are plated on a glass cover slip in 2 mL growth medium in a six-well culture plate 24 h before transfection.
2. Transfect 2 μg of shRNA-expression vector, or cotransfect 0.5 μg of RNAi-target gene-expression vector and 1.5 μg of trigger shRNA-expression vector by using Lipofectamine 2000 following the manufacturer's protocol.
3. Incubate the transfected cells at 37°C in a CO_2 incubator for 48 h.

3.4.2. Isolation of Total RNAs for Northern Blot or RT-PCR

1. Remove growth medium and wash the transfected cells three times with PBS.
2. Harvest the transfected cells from the plate by using cell scrapers or spatulas into a 50-mL culture tube.
3. Purify total RNAs from the transfected cells by using TRI reagent following the manufacturer's protocol.
4. Perform Northern blot or RT-PCR analysis with specific probe or primer pair according to standard protocols, respectively.

3.4.3. Preparation of Total Cell Lysates for Western Blot

1. Remove growth medium and wash the transfected cells three times with PBS.
2. Harvest the transfected cells from the plate by using cell scrapers or spatulas into a 50-mL culture tube.
3. Prepare total cell lysates from the transfected cells by using protein lysis buffer containing protease inhibitors.
4. Perform Western blot analysis with specific antibody according to standard protocols.

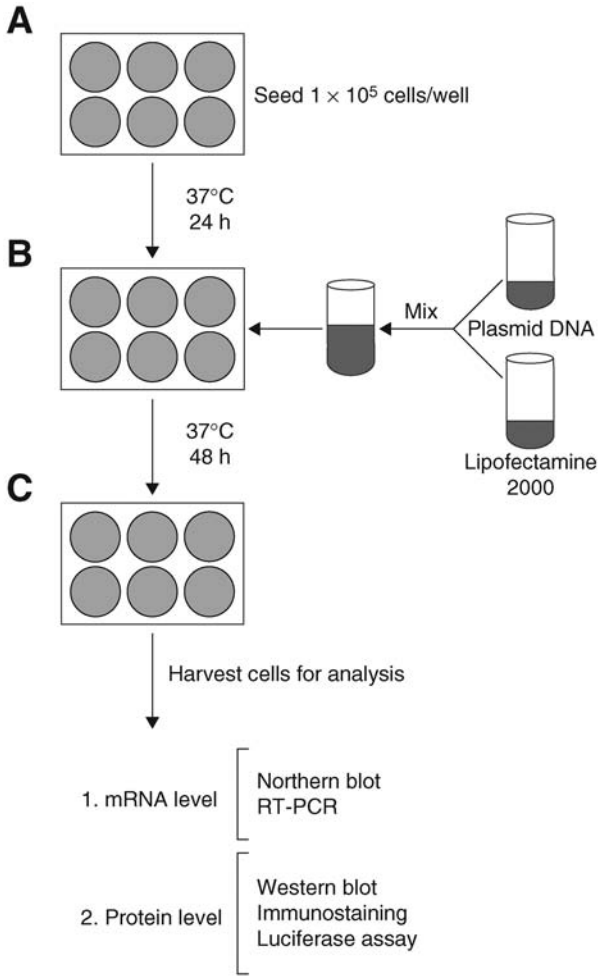


Fig. 6. Experimental procedures for assessing the inhibition efficiency of shRNA-expression constructs. **(A)** Seeding of targeting cell line. The targeting cell line is subcultured 24 h before transfection and plated into six-well culture plate at 1×10^5 cells per well. **(B)** Transfection of shRNA-expression construct. The cultured cells are either transfected with 2 μg of shRNA-expression construct or cotransfected 0.5 μg of RNAi target-gene expression construct and 1.5 μg of trigger shRNA-expression construct by using Lipofectamine 2000 according to the manufacturer's instructions. **(C)** Assessment of inhibition efficiency. After 48 h incubation, the transfected cells are harvested and lysed for either RNA or protein level analysis of target-gene expression by using Northern blot, RT-PCR, Western blot, immunostaining, or functional reporter assay (luciferase activity).

3.4.4. Fixation of Transfected Cells for Immunostaining

1. Remove growth medium and wash the transfected cells three times with PBS.
2. Fix the transfected cells with 3% paraformaldehyde for 1 min and wash the fixed cells three times with PBS.
3. Permeabilize the fixed cells with 0.5% Triton X-100 for 15 min and wash the permeabilized cells three times with PBS.
4. Perform immunostaining with specific antibody according to standard protocols.

4. Notes

1. This approach is cost-effective and convenient, as any annealed oligonucleotide duplexes can be directly cloned into these four different expression vectors at the same time.
2. The main advantage of this approach is that the preparation of the inserting vectors is simple and efficient by only double digestion with restriction enzymes *Cla*I and *Hin*dIII to remove the stuffer of $Puro^R$ gene from pHsH1puro, pHsU6puro, pMmH1puro, and pMmU6puro vectors. This will dramatically increase the cloning efficiency to more than 75%.
3. The oligonucleotides used for constructing the systems can be purchased from any local commercial suppliers without any further modification or treatment.
4. The sequential digestion of pGEM-7ZF(+) by *Eco*RI and *Hin*dIII followed by agarose gel purification is strongly recommended to ensure a complete digestion of vector by both restriction enzymes. This will greatly reduce the self-ligation of vector in the cloning.
5. The length of duplex region for a shRNA is relatively flexible from 19- to 29-nt. Although increasing the length of duplex region for a relatively ineffective 19-nt shRNA can increase its effectiveness, increasing the length of an effective 19-nt shRNA may not further improve the inhibition effect.
6. The annealing of two complementary oligonucleotides can be efficiently carried out in 1X T4 DNA ligase buffer, which can be obtained from any T4 DNA ligase commercial suppliers.
7. The annealed oligonucleotide duplexes do not need to be phosphorylated before the ligation step because it might result in multiple copies of insertion.
8. It is important to construct the second expression cassette with the same orientation as both the human and mouse H1 promoters, because these two promoters could express the protein-coding genes by the activity of Pol II-dependent poly(ADP-ribose) polymerase-2 promoter, efficiently. Otherwise the Pol II-dependent poly(ADP-ribose) polymerase-2 promoter could possibly transcribe the antisense strand of protein-coding genes, resulting in formation of the long dsRNAs that might trigger non-selective cytotoxic effects ***(44)***.
9. By using this protocol for cloning the shRNA-expression cassettes, it is efficient and cost-effective that only four colonies are selected and screened for the positive clones containing the shRNA-expression sequence.

Acknowledgments

This work was supported by grants from the National Science Council of Taiwan, ROC (to Wen-Tsan Chang).

References

1. Fire, A., Xu, S., Montgomery, M. K., Kostas, S. A., Driver, S. E., and Mello, C. C. (1998) Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature* **391,** 806–811.
2. Meister, G. and Tuschl, T. (2004) Mechanisms of gene silencing by double-stranded RNA. *Nature* **431,** 343–349.
3. Bernstein, E., Caudy, A. A., Hammond, S. M., and Hannon, G. J. (2001) Role for a bidentate ribonuclease in the initiation step of RNA interference. *Nature* **409,** 363–366.
4. Carmell, M. A. and Hannon, G. J. (2004) RNase III enzymes and the initiation of gene silencing. *Nat. Struct. Mol. Biol.* **11,** 214–218.
5. Martinez, J., Patkaniowska, A., Urlaub, H., Luhrmann, R., and Tuschl, T. (2002) Single-stranded antisense siRNAs guide target RNA cleavage in RNAi. *Cell* **110,** 563–574.
6. Schwarz, D. S., Hutvagner, G., Haley, B., and Zamore, P. D. (2002) Evidence that siRNAs function as guides, not primers, in the *Drosophila* and human RNAi pathways. *Mol. Cell* **10,** 537–548.
7. Schramke, V. and Allshire, R. (2003) Hairpin RNAs and retrotransposon LTRs effect RNAi and chromatin-based gene silencing. *Science* **301,** 1069–1074.
8. Soifer, H. S., Zaragoza, A., Peyvan, M., Behlke, M. A., and Rossi, J. J. (2005) A potential role for RNA interference in controlling the activity of the human LINE-1 retrotransposon. *Nucleic Acids Res*. **33,** 846–856.
9. Reinhart, B. J., Slack, F. J., Basson, M., et al. (2000) The 21-nucleotide let-7 RNA regulates developmental timing in *Caenorhabditis elegans*. *Nature* **403,** 901–906.
10. Volpe, T. A., Kidner, C., Hall, I. M., Teng, G., Grewal, S. I., and Martienssen, R. A. (2002) Regulation of heterochromatic silencing and histone H3 lysine-9 methylation by RNAi. *Science* **297,** 1833–1837.
11. Matzke, M. A. and Birchler, J. A. (2005) RNAi-mediated pathways in the nucleus. *Nat. Rev. Genet*. **6,** 24–35.
12. Shuey, D. J., McCallus, D. E., and Giordano, T. (2002) RNAi: gene-silencing in therapeutic intervention. *Drug Discov. Today* **7,** 1040–1046.
13. Dorsett, Y. and Tuschl, T. (2004) siRNAs: applications in functional genomics and potential as therapeutics. *Nat. Rev. Drug Discov.* **3,** 318–329.
14. Berns, K., Hijmans, E. M., Mullenders, J., et al. (2004) A large-scale screen in human cells identifies new components of the p53 pathway. *Nature* **428,** 431–437.
15. Paddison, P. J., Silva, J. M. L., Conklin, D. S., et al. (2004) A resource for large-scale RNA-interference-based screens in mammals. *Nature* **428,** 427–431.

16. Player, M. R. and Torrence, P. F. (1998) The 2-5 A system: modulation of viral and cellular processes through acceleration of RNA degradation. *Pharmacol. Ther.* **78,** 55–113.
17. Stark, G. R., Kerr, I. M., Williams, B. R., Silverman, R. H., and Schreiber, R. D. (1998) How cells respond to interferons. *Annu. Rev. Biochem.* **67,** 227–264.
18. Gil, J. and Esteban, M. (2000) Induction of apoptosis by the dsRNA-dependent protein kinase (PKR): mechanism of action. *Apoptosis* **5,** 107–114.
19. Geiss, G., Jin, G., Guo, J., Bumgarner, R., Katze, M. G., and Sen, G. C. (2001) A comprehensive view of regulation of gene expression by double-stranded RNA-mediated cell signaling. *J. Biol. Chem.* **276,** 30,178–30,182.
20. Samuel, C. E. (2001) Antiviral actions of interferons. *Clin. Microbiol. Rev.* **14,** 778–809.
21. Elbashir, S. M., Harborth, J., Lendeckel, W., Yalcin, A., Weber, K., and Tuschl, T. (2001) Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells. *Nature* **411,** 494–498.
22. Elbashir, S. M., Harborth, J., Weber, K., and Tuschl, T. (2002) Analysis of gene function in somatic mammalian cells using small interfering RNAs. *Methods* **26,** 199–213.
23. Siolas, D., Lerner, C., Burchard, J., et al. (2005) Synthetic shRNA as potent RNAi triggers. *Nat. Biotechnol.* **23,** 227–231.
24. Lee, N. S., Dohjima, T., Bauer, G., et al. (2002) Expression of small interfering RNAs targeted against HIV-1 rev transcripts in human cells. *Nat. Biotechnol.* **20,** 500–505.
25. Miyagishi, M. and Taira, K. (2002) U6 promoter-driven siRNAs with four uridine 3′ overhangs efficiently suppress targeted gene expression in mammalian cells. *Nat. Biotechnol.* **20,** 497–500.
26. Zheng, L., Liu, J., Batalov, S., et al. (2004) An approach to genomewide screens of expressed small interfering RNAs in mammalian cells. *Proc. Natl. Acad. Sci. USA* **101,** 135–140.
27. Brummelkamp, T. R., Bernards, R., and Agami, R. (2002) A system for stable expression of short interfering RNAs in mammalian cells. *Science* **296,** 550–553.
28. Sui, G., Soohoo, C., Affar, E. B., et al. (2002) A DNA vector-based RNAi technology to suppress gene expression in mammalian cells. *Proc. Natl. Acad. Sci. USA* **99,** 5515–5520.
29. Paddison, P. J., Caudy, A. A., Bernstein, E., Hannon, G. J., and Conklin, D. S. (2002) Short hairpin RNAs (shRNAs) induce sequence-specific silencing in mammalian cells. *Genes Dev.* **16,** 948–958.
30. Paul, C. P., Good, P. D., Winer, I., and Engelke, D. R. (2002) Effective expression of small interfering RNA in human cells. *Nat. Biotechnol.* **20,** 505–508.
31. Scherer, L. J. and Rossi, J. J. (2003) Approaches for the sequence-specific knockdown of mRNA. *Nat. Biotechnol.* **21,** 1457–1465.
32. Brummelkamp, T. R., Bernards, R., and Agami, R. (2002) Stable suppression of tumorigenicity by virus-mediated RNA interference. *Cancer Cell* **2,** 243–247.

33. Gupta, S., Schoer, R. A., Egan, J. E., Hannon, G. J., and Mittal, V. (2004) Inducible, reversible, and stable RNA interference in mammalian cells. *Proc. Natl. Acad. Sci. USA* **101,** 1927–1932.
34. Baer, M., Nilsen, T. W., Costigan, C., and Altman, S. (1990) Structure and transcription of a human gene for HI RNA, the RNA component of human RNase P. *Nucleic Acids Res.* **18,** 97–103.
35. Paule, M. R. and White, R. J. (2000) Survey and summary: transcription by RNA polymerases I and III. *Nucleic Acids Res.* **28,** 1283–1298.
36. Myslinski, E., Ame, J. C., Krol, A., and Carbon, P. (2001) An unusually compact external promoter for RNA polymerase III transcription of the human HI RNA gene. *Nucleic Acids Res.* **29,** 2502–2509.
37. Yu, J. Y., DeRuiter, S. L., and Turner, D. L. (2002) RNA interference by expression of short-interfering RNAs and hairpin RNAs in mammalian cells. *Proc. Natl. Acad. Sci. USA* **99,** 6047–6052.
38. Wu, M. -T., Wu, R. -H., Hung, C. -F., Cheng, T. -L., Tsai, W. -H., and Chang, W. -T. (2005) Simple and efficient DNA vector-based RNAi systems in mammalian cells. *Biochem. Biophys. Res. Commun.* **330,** 53–59.
39. Khvorova, A., Reynolds, A., and Jayasena, S. D. (2003) Functional siRNAs and miRNAs exhibit strand bias. *Cell* **115,** 209–216.
40. Schwarz, D. S., Hutvagner, G., Du, T., Xu, Z., Aronin, N., and Zamore, P. D. (2003) Asymmetry in the assembly of the RNAi enzyme complex. *Cell* **115,** 199–208.
41. Reynolds, A., Leake, D., Boese, Q., Scaringe, S., Marshall, W. S., and Khvorova, A. (2004) Rational siRNA design for RNA interference. *Nat. Biotechnol.* **22,** 326–330.
42. Ui-Tei, K., Naito, Y., Takahashi, F., et al. (2004) Guidelines for the selection of highly effective siRNA sequences for mammalian and chick RNA interference. *Nucleic Acids Res.* **32,** 936–948.
43. Mittal, V. (2004) Improving the efficiency of RNA interference in mammals. *Nat. Rev. Genet.* **5,** 355–365.
44. Hung, C. -F., Cheng, T. -L., Wu, R. -H., Teng, C. -F., and Chang, W. -T. (2006) A novel bidirectional expression system for simultaneous expression of both the protein-coding genes and short hairpin RNAs in mammalian cells. *Biochem. Biophys. Res. Commun.* **339,** 1035–1042.

14

Selection of Recombinant Antibodies From Antibody Gene Libraries

Michael Hust, Stefan Dübel, and Thomas Schirrmann

Summary

After the sequencing of the human genome is completed, the research focus shifts toward the analysis of gene products. The human genome encodes more than 30,000 genes. Owing to alternative mRNA splicing and posttranslational modifications, for example, glycosylation, phoshorylation, and so on, the number of different proteins of human proteome is supposed to easily exceed 90,000. Antibodies are key detection reagents for the "postgenomic" analysis of these proteins. Any systematic investigation of the human proteome requires high throughput methods for antibody generation. In vitro selection systems utilizing recombinant antibody repertoires offer this capability and capacity. The most commonly used contemporary in vitro selection system is antibody phage display, which has already yielded thousands of useful antibodies for therapy, research, and diagnostics. Herein, methods are described for the selection of recombinant antibody fragments from naive antibody gene libraries.

Key Words: Antibody engineering; panning; phage display; scFv; antibody gene libraries; filamentous phage.

1. Introduction

The production of polyclonal antibodies by immunization of animals is established for more than a century. The first antibody serum was directed against diphtheria and produced in horses ***(1)***. Hybridoma technology was the next milestone, allowing the production of monoclonal antibodies by fusion of an antibody producing spleen B-cell with an immortal myeloma cell ***(2)***. However, hybridoma technology has some limitations like a potential genetic instability of the aneuploid cell lines and most of all its inability to produce antibodies against toxic or highly conserved antigens ***(3)***. When repeatedly administered in therapy, murine hybridoma antibodies induce a human anti-mouse antibody response caused by murine antibodies ***(4)***. This problem can be

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

overcome by two approaches: By humanization of mouse antibodies ***(5)*** or by using repertoires of human antibody genes. The second approach was achieved in two ways. First, human antibody gene repertoires were inserted into the genomes of immunoglobulin (Ig)G-knockout-mice, allowing to generate hybridoma cell lines, which produce human Igs ***(6–8)***. This method yielded a significant number of antibodies that reached clinical studies, but still requires immunization and has the limitations in respect of toxic and conserved antigens.

Alternatively, human antibodies can be generated completely independent from any immune system by an in vitro selection process: "antibody phage display," which utilizes human antibody gene libraries displayed on bacteriophage is the method of choice. The first antibody gene repertoires in phage were generated and screened by using the lytic phage λ ***(9)*** with very limited success. The display method most commonly used today is based on the groundbreaking work of Georg P. Smith ***(10)*** on filamentous phage display. Herein, the genotype and phenotype of peptides were linked by fusing their short gene fragments to the minor coat protein III gene of the filamentous bacteriophage M13. The resulting peptide::pIII fusion protein is expressed on the surface of phage allowing the affinity purification of the desired gene by peptide binding. In the same way, antibody fragments fused to pIII can be presented on the surface of M13 phage ***(11–16)***. Owing to limitations of the *Escherichia coli* folding machinery, complete IgG molecules cannot be displayed on the surface of phage. Therefore, smaller antibody fragments are used for antibody phage display: the Fab fragment or the single chain Fv fragment (scFv). Fab fragments consist of two chains, the variable (V_H) and first constant region of the heavy chain (C_{H1}) and the light chain (LC) of the antibody, both linked by a disulphide bond. In contrast, scFv fragments consist of only one polypeptide chain, made up of the variable region of the heavy chain (V_H) and the variable region of the LC (V_L) fused by a short peptide linker. Two different genetic systems have been developed for the expression of the antibody::pIII fusion proteins. First, the antibody genes can be directly inserted into the phage genome fused to the wild-type *pIII* gene ***(11)***. However, most of the successful systems uncouple antibody expression from phage propagation by providing the genes encoding the antibody::pIII fusion proteins a separate plasmid (phagemid), containing a phage morphogenetic signal for packaging the vector into the assembled phage particles ***(12–16)***. A large variety of phagemids have been constructed for the display of scFvs or Fabs on filamentous phage (for an overview *see* **refs. *17***).

Different types of antibody phage display gene libraries have been created from different genetic sources. First, the variable region genes of Ig-secreting plasma cells from immunized donors or from patients with an antibody titer

against the desired antigen could be isolated to construct "immune" libraries ***(14,18)***. Immune libraries are typically used in medical research to select abundant antibodies against one particular antigen or group of antigens, for example, of infectious pathogens, whereas they are not the source of choice for the isolation of antibodies with other specificities. "Single-pot" or universal libraries are designed to provide antibody fragments binding to every possible antigen. Naive libraries are constructed from rearranged antibody genes from IgM producing B-cells of nonimmunized donors. An example for this library type is the naive human Fab library constructed by de Haardt et al. ***(19)***. "Semisynthetic" libraries are derived from not rearranged V-genes from pre-B-cells (germline cells) or from a single antibody framework with at least one complementary determining region (CDR) genetically randomized, such as the library described by Pini et al. ***(20)***. A combination of naive and synthetic repertoire was used by Hoet et al. ***(21)***. They combined LCs from autoimmune patients with a Fd fragment containing synthetic CDR 1 and CDR2 in one human framework and naive CDR3 regions, origined from autoimmune patients. Fully synthetic libraries have a human framework with randomly integrated CDR cassettes ***(22,23)***. All library types—"immune," "naïve," and "synthetic" and their intermediates—have been proven to be useful sources for the selection of antibodies for diagnostic and therapeutic purposes. To date, "single-pot" antibody libraries with a theoretical diversity of up to 10^{11} independent clones have been generated ***(24)*** to serve as a molecular repertoire for phage display selection procedures. An overview of antibody libraries and the comparison of their construction principles is given by Hust and Dübel ***(25)*** **(Fig. 1)**.

2. Materials

2.1. Coating of Microtiter Wells

1. Maxisorb microtiter plates oder stripes (Nunc, Wiesbaden, Germany).
2. Phosphate-buffered saline (PBS): 8 g/L NaCl, 0.2 g/L KCl, 1.44 g/L $Na_2HPO_4 \cdot 2H_2O$, and 0.24 g/L KH_2PO_4, pH 7.4.
3. Dimethyl sulfoxide.
4. Phosphate-buffered saline Tween (PBST) PBS + 0.1% Tween-20.

2.2. Panning

1. Milk phosphate-buffered saline Tween (MPBST) 2% skim milk in PBST, prepare fresh.
2. Panning block solution: 1% (w/v) skim milk and 1% (w/v) bovine serum albumin (BSA) in PBST, prepare fresh.
3. 10 µg/mL Trypsin in PBS.
4. *E. coli* XL1-blue MRF`(Stratagene), genotype: Δ(*mcrA*)*183* Δ(*mcrCB-hsdSMR-mrr*)*173 endA1 supE44 thi-1 recA1 gyrA96 relA1 lac* (F′ *proAB* $LacI^qZ\Delta M15$ *Tn10* [Tet^R]).

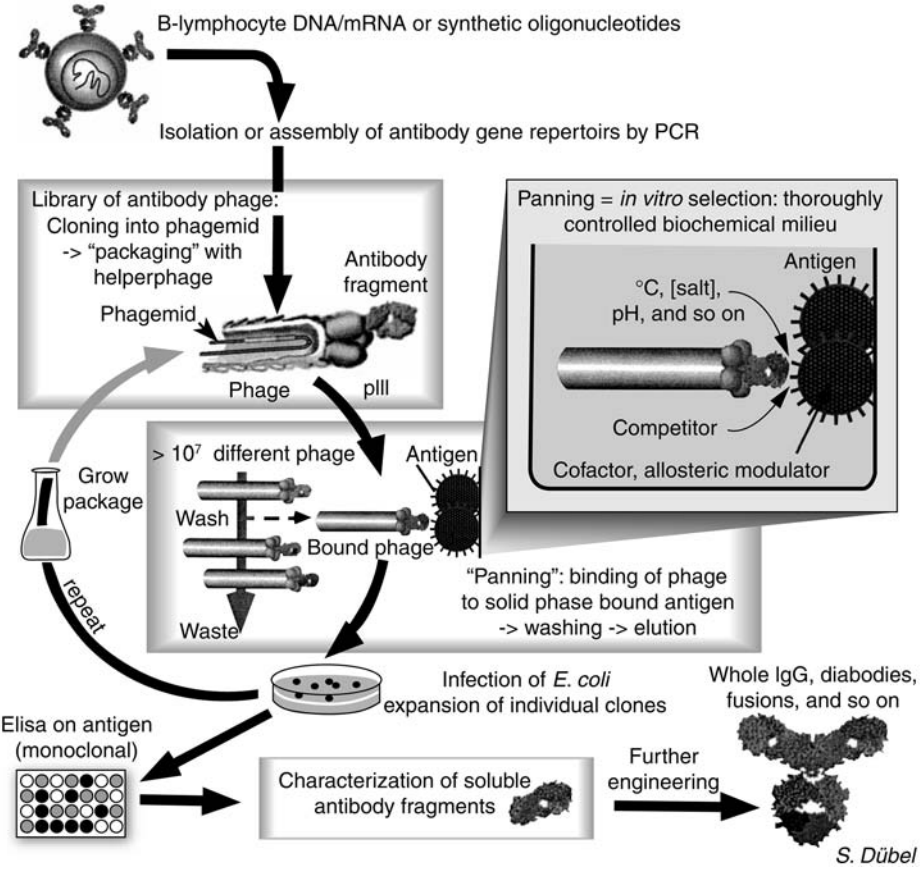


Fig. 1. Schematic description of the selection of antibodies from antibody libraries (panning) by phage display.

5. M13K07 Helperphage (Stratagene, Amsterdam, Netherlands).
6. 2X TY media: 1.6% (w/v) tryptone, 1% (w/v) yeast extract, and 0.5% (w/v) NaCl, pH 7.0.
7. 2X TY-T: 2X TY, containing 50 µg/mL tetracycline.
8. Super Optimal Broth (SOB) media: 2% (w/v) tryptone, 0.5% (w/v) yeast extract, and 0.05% (w/v) NaCl, after autoclavation add sterile 1% (v/v) of the 2 *M* Mg solution, pH 7.0.
9. 2 *M* Mg solution: 1 *M* MgCl + 1 *M* $MgSO_4$.
10. Super Optimal Broth Glucose Ampicilin (SOB-GA) SOB, containing 100 µg/mL ampicilin and 100 m*M* glucose.
11. SOB-GA agar plates: SOB-GA + 1.5% (w/v) agar–agar.
12. 15-cm Petri dishes.
13. 2X TY-GA: 2X TY, containing 100 m*M* glucose and 100 µg/mL ampicilin.
14. Glycerin solution: 87% (v/v).

2.3. Packaging of Phagemids

1. 2X TY-AK: 2X TY, containing 100 µg/mL ampicillin, and 50 µg/mL kanamycin.
2. Polyethylene glycol (PEG) solution: 20% (w/v) PEG 6000: Fluka (part of Sigma-Aldrich Chemie Gmbhl), München, Germany and 2.5 *M* NaCl.
3. Phage dilution buffer: 10 m*M* Tris-HCl, 20 m*M* NaCl, and 2 m*M* ethylenediaminetetraacetic acid, pH 7.5.

2.4. Titering

1. 2X TY-GA agar plates: 2X TY-GA and 1.5% (w/v) agar–agar.

2.5. ELISA of a Polyclonal Antibody Phage Suspension

1. BSA: prepare a 10 mg/mL stock solution in PBS.
2. Anti-M13, horseradish peroxidase (HRP)-conjugated monoclonal antibody (Amersham Bioscience; GE Healthcare, München, Germany).
3. Tetramethylbenzidine (TMB) solution A: 10 g citric acid solved in 100 mL water and add 9.73 g potassium citrate to 1 L water, pH 4.1.
4. TMB solution B: 240 mg tetramethylbenzidine, 10 mL acetone, 90 mL ethanol, and 907 µL 30% H_2O_2.
5. 1 *N* H_2SO_4.

2.6. Production of Soluble Monoclonal Antibody Fragments in Microtiter Plates

1. 96-well U-bottom polypropylene (PP) microtiter plates (Greiner, Germany).
2. AeraSeal breathable sealing film (Excel Scientific: Wrightwood, CA).
3. 2X TY-A containing 50 µ*M* isopropyl-β-D-thiogalactopyranoside (IPTG).

2.7. ELISA of Soluble Monoclonal Antibody Fragments

1. Mouse α-His-tag monoclonal antibodies (α-Penta His, Qiagen, Germany).
2. Mouse α-myc-tag monoclonal antibodies (9E10, Sigma, Germany).
3. Mouse α-pIII monoclonal antibodies (PSKAN3, Mobitec, Germany).
4. Goat α-mouse IgG serum (Fab specific) HRP-conjugated (Sigma, Germany).

3. Methods

The *in vitro* procedure for isolating antibody fragments by their binding activity was called "panning," referring to the gold washers tool ***(26)***. The antigen is immobilized to a solid surface, such as nitrocellulose (e.g., **ref. *27***), magnetic beads (e.g., **ref. *28***), a column matrix (e.g., **ref. *12***) or, most widely used, plastic surfaces as polystyrole tubes (e.g., **ref. *29***) or 96-well microtiter plates (e.g., **ref. *13***). The antibody phage are incubated with the surface-bound antigen, followed by thorough washing to remove the vast excess of nonbinding antibody phage. The bound antibody phage can subsequently be eluted and reamplified by infection of *E. coli*. This amplification allows detection of a single molecular

interaction during panning as after elution a single antibody phage can give rise to a bacterial colony by its resistance marker. The selection cycle can be repeated by infection of the phagemid bearing *E. coli* colonies from the former panning round with a helperphage to produce new antibody phage, which can be used for further panning rounds until a significant enrichment of antigen-specific phage is achieved. The number of antigen-specific antibody phage clones should increase with every panning round. Usually two to six panning rounds are necessary to select specifically binding antibody fragments. High throughput methods using microtiter plates and robotics can facilitate and enhance the panning procedure (for review *see* **ref. *30***).

The first step in the evaluation process of potential binders is mostly done by an enzyme-linked immunosorbent assay (ELISA) with polyclonal phage preparations from each panning round against immobilized, i.e., coated, target antigen and on control protein, for example, BSA. In the next step, antibody clones of panning rounds showing a significant enrichment of specific antigen binding in the polyclonal phage ELISA are produced as soluble monoclonal antibody fragments in microtiter plates followed by an ELISA on coated antigen vs on control protein. The following protocols describe the selection of recombinant antibody fragments from antibody gene libraries by phage display and the initial analysis of the selected antibody fragments.

3.1. Coating of Microtiter Plate Wells

1. (a) *Protein antigen:* for the first panning round, use 2–10 µg protein per panning, for the following rounds use 0.1–1 µg protein for more stringent conditions. Dissolve the antigen in 150 µL PBS, transfer into the microtiter plate well and incubate overnight at 4°C (*see* **Note 1**) and (b) oligopeptide antigen: use 100–500 ng oligopeptide for each panning round. Dissolve the oligopeptide in 150 µL 5% (v/v) dimethyl sulfoxide containing PBS, transfer into the microtiter plate well and incubate overnight at 4°C (*see* **Note 2**).
2. Wash the coated microtiter plate wells three times with PBST using an ELISA washer (*see* **Note 3**).

3.2. Panning

1. (a) Block the antigen-coated wells with MPBST for 2 h at RT. The wells must be completely filled and (b) perform this step only in the first panning round. In parallel, block an additional well (without antigen) per panning with MPBST for 1 h at RT for preincubation of the antibody gene library. The wells must be completely filled. Wash three times with PBST (*see* **Note 3**). Incubate 10^{11}–10^{12} antibody phage from the library in 150 µL panning block for 1 h at RT. This step removes unspecific binders from the antibody gene library.
2. Wash the blocked antigen-coated wells three times with PBST (*see* **Note 3**). Either carry over the preincubated antibody phage library to the blocked wells or

fill 10^{11}–10^{12} amplified phage solved in 150 µL panning block from the former panning round in the blocked wells. Incubate at RT for 2 h for binding of the antibody phage.

3. Remove the unspecifically bound antibody phage by stringent washing. Thereafter, wash the wells 10 times with an ELISA washer in the first panning round. In the following panning rounds increase the washing steps (20 times in the second panning round, 30 times in the third panning round, and so on.) (*see* **Note 3**).
4. Elute with 200 µL trypsin solution for 30 min at 37°C (*see* **Note 4**).
5. Use 10 µL of the eluted phage for titering (*see* titering).
6. Inoculate 50 mL 2X TY-T with an overnight culture of *E. coli* XL1-blue MRF′ (Stratagene, Amsterdam, Netherland) in 100-mL Erlenmeyer flasks and grow at 250 rpm and 37°C.
7. Infect exponentially (OD_{600} ~0.5 nm, after 2–3 h) growing 20 mL XL1-blue MRF′ culture with the remaining 190 µL of the eluted phage. Incubate 30 min at 37°C without shaking and the following 30 min with 250 rpm.
8. Harvest the infected bacteria by centrifugation for 10 min at 3200*g* in 50-mL PP tubes. Resolve the pellet in 250 µL SOB-GA and plate the bacteria suspension on SOB-GA agar plates (15-cm Petri dish). Grow overnight at 37°C (*see* **Note 5**).
9. Harvest the grown colonies by suspending in 2.5 mL 2X TY-GA with a Drigalsky spatula.
10. Use 100 µL of the harvested bacteria for the amplification of the eluted phage (*see* **Subheading 3.3.**).
11. Make a glycerin stock of the panning round by adding 250 µL 87% glycerin to 750 mL of the harvested bacteria. Mix and store at −80°C.

3.3. Packaging of Phagemids

1. For the next panning round the eluted phage must be packaged and reamplified. Inoculate 50 mL 2X TY-GA in a 100-mL Erlenmeyer flask with 100 µL harvested bacteria ($OD_{600} < 0.1$ nm). Grow at 250 rpm at 37°C up to an OD_{600} approx 0.5 nm.
2. Infect 5 mL bacteria culture (~2.5×10^9 cells) with 5×10^{10} PFU (multiplicity of infection = 1:20) of the helperphage M13K07. Incubate for 30 min without shaking and the following 30 min with 250 rpm at 37°C.
3. To remove the glucose, harvest the cells by centrifugation for 10 min at 3200*g* in 50-mL PP tubes.
4. Resuspend the pellet in 30 mL 2X TY-AK in a 100-mL Erlenmeyer flask. Produce the phage for 16 h at 250 rpm and 30°C
5. Pellet the bacteria by centrifugation for 10 min at 3200*g* in 50-mL PP tubes. If the supernatant is not clear, centrifuge again to remove remaining bacteria.
6. Precipitate the phage in the supernatant by adding one-fifth volume PEG solution in 50-mL PP tubes. Incubate for 1 h at 4°C with gentle shaking.
7. Pellet the phage by centrifugation for 1 h at 3200*g* and 4°C. Put the open tubes upside down on tissue paper and let the viscous PEG solution move out completely. Resuspend the phage pellet in 500 µL phage dilution buffer. Titer the phage preparation and use it for the next panning round. Store the remaining phage at 4°C.

3.4. Titering

1. Inoculate 5 mL 2X TY-T in a 100-mL Erlenmeyer flask with XL1-blue MRF′ and grow overnight at 37°C and 250 rpm.
2. Inoculate 50 mL 2X TY-T with 500 µL overnight culture and grow at 250 rpm at 37°C up to OD_{600} approx 0.5 nm (*see* **Note 6**).
3. Make serial dilutions of the phage solution in PBS. The number of eluted phages depends on several parameters (e.g., antigen, library, panning round, washing stringency, and so on). In case of a successful panning, the phage titer usually is 10^3–10^5 phage per well in the first round increasing to 10^6–10^9 phage per well in the succeeding rounds. The phage preparation after reamplification of the eluted phage have a titer of about 10^{12}–10^{13} phage/mL.
4. Infect 50 µL bacteria with 10 µL phage dilution and incubate for 30 min at 37°C (*see* **Note 7**).
5. Titrations can be done in two different ways:
   a. Plate the 60 µL infected bacteria on 2X TY-GA agar plates (9-cm Petri dishes).
   b. Pipet 10 µL (better as triplicate) on 2X TY-GA agar plates. Here, about 20 titering spots can be placed on one 9-cm Petri dish.
6. Incubate the plates overnight at 37°C.
7. Count the colonies and calculate the colony-forming units titer, according to the dilution.

3.5. ELISA of a Polyclonal Antibody Phage Suspension

1. To investigate the enrichment of antigen-specific antibody phage after a panning round, prepare microtiter plates with 100–1000 ng antigen per well for each panning round (for method *see* **Subheading 3.1.**). As a control, prepare wells with 100–1000 ng BSA in 150 µL PBS overnight at 4°C (*see* **Note 8**).
2. Wash the coated microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2.** and **Note 3**).
3. Block the antigen-coated wells with MPBST for 2 h at RT. The wells must be completely filled.
4. Wash the coated microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2.** and **Note 3**).
5. Resuspend 10^{10} antibody phage from each panning round in 150 µL 2%MPST and incubate them for 1.5 h on the antigen and the BSA control, respectively.
6. Wash the microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2.** and **Note 3**).
7. Incubate each well with 100 µL HRP-conjugated anti-M13 antibody 1:5000 diluted in MPST for 1.5 h.
8. Wash the microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2** and **Note 3**).
9. Shortly before use, mix 19 parts TMB solution A and one part TMB solution B. Add 100 µL of the prepared TMB solution to each well and incubate for 1–15 min.

10. Stop the substrate reaction by adding 100 µL 1 *N* sulfuric acid. The color turns from blue to yellow.
11. Measure the extinction at 450 nm in an ELISA reader.

3.6. Production of Soluble Monoclonal Antibody Fragments in Microtiter Plates

1. Fill each well of a 96-well U-bottom PP microtiter plate with 150 µL 2X TY-GA.
2. Pick 96 clones with sterile tips from the desired panning round (*see* **Note 9**) and inoculate each well (*see* **Note 10**). Seal the plate with a breathable sealing film.
3. Incubate overnight in a microtiter plate shaker (e.g., Thermo Shaker PST-60HL-4, Lab4You, Germany) at 37°C and 1200 rpm.
4. (a) Fill a new 96-well polypropyplene microtiter plate with 150 µL 2X TY-GA and add 10 µL of the overnight cultures. Incubate for 2 h at 37°C and 1200 rpm and (b) add 30 µL glycerin solution to the remaining 140 µL overnight cultures. Mix by pipeting and store this masterplate at −80°C.
5. Pellet the bacteria in the microtiter plates by centrifugation for 10 min at 3200*g* and 4°C. Remove 180 µL glucose containing media by carefully pipeting (do not disturb the pellet).
6. Add 180 µL 2X TY-A with 50 µ*M* IPTG and incubate overnight at 30°C and 1200 rpm (*see* **Note 11**).
7. Pellet the bacteria by centrifugation for 10 min at 3200*g* in the microtiter plates. Transfer the antibody fragment containing supernatant to a new PP microtiter plate and store at 4°C.

3.7. ELISA of Soluble Monoclonal Antibody Fragments

1. To analyze the antigen specificity of the monoclonal soluble antibodies, coat 100–1000 ng antigen per well overnight at 4°C. As control coat 100–1000 ng BSA per well (*see* **Subheading 3.1.**, **Notes 8** and **10**).
2. Wash the coated microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2.**, and **Note 3**).
3. Block the antigen-coated wells with MPST for 2 h at RT. The wells must be completely filled.
4. Fill 50 µL MPST in each well and add 50 µL of antibody solution (*see* **Subheading 3.6.**). Incubate for 1.5 h at RT (or overnight at 4°C).
5. Wash the microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2.**, and **Note 3**).
6. Incubate 100 µL α-tag antibody solution for 1.5 h (appropriate dilution in MPBST).
7. Wash the microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2.**, and **Note 3**).
8. Incubate 100 µL goat α-mouse HRP conjugate (1:10,000 in MPBST).
9. Wash the microtiter plate wells three times with PBST (washing procedure *see* **Subheading 3.2.**, and **Note 3**).

10. Shortly before use, mix 19 parts TMB solution A and 1 part TMB solution B. Add 100 µL of the prepared TMB solution into each well and incubate for 1–15 min.
11. Stop the color reaction by adding 100 µL 1 *N* sulfuric acid. The color turns from blue to yellow.
12. Measure the extinction at 450 nm in an ELISA reader.
13. Identify positive candidates with a signal (antigen) 10X over noise (BSA).

4. Notes

1. If the protein is not binding properly to the microtiter plate surface, try bicarbonate buffer (50 m*M* $NaHCO_3$, pH 9.6).
2. If biotinylated oligopeptide is used, dissolve 100 ng streptavidin in 150 µL PBS and coat overnight at 4°C. Coat two wells for each panning, one well is for the panning, the second one for the preincubation of the library to remove streptavidin binders! Some time is necessary to use free streptavidin during panning in competition to remove streptavidin binders. Pour out the wells and wash three times with PBST. Dissolve 100–500 ng biotinylated oligopeptide in PBS and incubate for 1 h at RT. Alternatively, oligopeptides can coupled to BSA and coat overnight at 4°C.
3. The washing should be performed with an ELISA washer (e.g., TECAN Columbus Plus: Crailsheim, Germany) for more stringent and reproducible washing results. To remove antigen or blocking solutions wash three times with PBST ("standard washing protocol" for TECAN washer). If no ELISA washer is available, wash manually three times with PBST. After binding of antibody phage, wash 10 times with PBST ("stringent bottom washing protocol" in case of TECAN washer). If no ELISA washer is available, wash manually 10 times with PBST and 10 times with PBS. For stringent off-rate selection increase the number of washing steps or additionally incubate the microtiter plate in 1 L PBS for some days.
4. Phagemids like pSEX81 ***(18)*** or pHAL1 ***(31)*** or pHAL 14 ***(17)*** have coding sequences for a trypsin-specific cleavage site between the antibody fragment gene and the gIII. Trypsin also cleaves within antibody fragments but does not cleave the phage. The phage protein pIII mediates the binding of the phage to the F pili of *E. coli* required for the infection. It is found that proteolytic cleavage of the antibody fragments from the antibody::pIII fusion by trypsin enhances the infection rate of eluted antibodies, especially when using Hyperphage (Progen, Heidelberg, Germany) as helperphage to obtain polyvalent display ***(32–34)***.
5. The high concentration of glucose is necessary to efficiently repress the lac promoter controlling the antibody::pIII fusion gene on the phagemid. Low glucose leads to an inefficient repression of the lac promoter and background expression of the antibody::pIII fusion protein. The strong selection pressure frequently causes mutations in the phagemid, especially in the promoter region and the antibody::pIII fusion gene. Bacteria with mutated phagemids can proliferate faster than bacteria with nonmutated phagemids. Therefore, the glucose can only be omitted at the phage production step.

6. If the bacteria have reached OD_{600} approx 0.5 nm before they are needed, store the culture at 4°C to maintain the F pili on the *E. coli* cells. If used for titration, a M13K07 positive control is advised.
7. It is advisable to conduct control titerings. To control the PBS, PEG solutions use 10 µL of this solution to "infect" bacteria with this solution and also plate out noninfected XL1-blue MRF′ to control the bacteria. It is recommended to clean the working place each time with virus-inactivating solutions (e.g., Barrycidal 36, BIO-HIT, Germany) and to use filter tips for pipeting.
8. Antibody phage binding unspecifically are usually enriched during panning. These unspecific binding usually results from misfolded or incomplete antibodies. They often bind to BSA, streptavidin, and plastic surfaces.
9. Use the polyclonal antibody phage ELISA to select the suitable panning round for picking.
10. It is recommended to pick only 92 clones. Use the wells H3, H6, H9, and H12 for controls. H3 and H6 are negative controls—these wells will not be inoculated and not used for the following ELISA with soluble antibodies. The wells H9 and H12 are inoculated with a clone containing a phagemid encoding a known antibody fragment. Therefore, the wells H9 and H12 are coated with the corresponding antigen.
11. The appropriate IPTG concentration for induction depends on the vector design. A concentration of 50 µ*M* was well suited for vectors with a Lac promoter like pSEX81 ***(18)***, pIT2 ***(35)***, and pHENIX ***(36)*** and pHAL14 ***(18)***. The method for the production of soluble antibodies works with vectors with (e.g., pHAL 14) and without (e.g., pSEX81) an amber stop codon between antibody fragment and gIII. If the vector has no amber stop codon the antibody::pIII fusion protein will be produced ***(37)***.

Acknowledgments

We gratefully acknowledge the financial support by the German ministry of education and research (Bundesministerium Für Bildung und Forschung [BMBF], Standard Method Protocol [SMP] "Antibody Factory" in the NGFN2 program) and of the German Research Foundation (Deutsche Forschungs geneins-chaf [DFG], SFB 578).

References

1. Von Behring, E. and Kitasato, S. (1890) Über das Zustandekommen der Diphtherie-Immunität und der Tetanus-Immunität bei Thieren. *Deutsche Med. Wochenzeitschr.* **16,** 1113, 1114.
2. Köhler, G. and Milstein, C. (1975) Continous cultures of fused cells secreting antibody of predefined specificity. *Nature* **256,** 495–497.
3. Winter, G. and Milstein, C. (1991) Man-made antibodies. *Nature* **349,** 293–299.
4. Courtenay-Luck, N. S., Epenetos, A. A., Moore, R., et al. (1986) Development of primary and secondary immune responses to mouse monoclonal antibodies used in the diagnosis and therapy of malignant neoplasms. *Cancer Res.* **46,** 6489–6493.
5. Studnicka, G. M., Soares, S., Better, M., Williams, R. E., Nadell, R., and Horwitz, A. H. (1994) Human-engineered monoclonal antibodies retain full specific binding

activity by preserving non-CDR complementarity-modulating residues. *Protein Eng.* **6,** 805–814.
6. Jakobovits, A. (1995) Production of fully human antibodies by transgenic mice. *Curr. Opin. Biotechnol.* **6,** 561–566.
7. Lonberg, N. and Huszar, D. (1995) Human antibodies from transgenic mice. *Int. Rev. Immunol.* **13,** 65–93.
8. Fishwild, D. M., O'Donnel, S. L., Bengoechea, T., et al. (1996) High-avidity human IgG kappa monoclonal antibodies from a novel strain of minilocus transgenic mice. *Nat. Biotechnol.* **14,** 845–851.
9. Huse, W. D., Sastry, L., Iverson, S. A., et al. (1989) Generation of a large combinatorial library of the immunoglobulin repertoire in phage lambda. *Science* **246,** 1275–1281.
10. Smith, G. P. (1985) Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface. *Science* **228,** 1315–1317.
11. McCafferty, J., Griffiths, A. D., Winter, G., and Chiswell, D. J. (1990) Phage antibodies: filamentous phage displaying antibody variable domain. *Nature* **348,** 552–554.
12. Breitling, F., Dübel, S., Seehaus, T., Kleewinghaus, I., and Little, M. (1991) A surface expression vector for antibody screening. *Gene* **104,** 1047–1153.
13. Barbas, C. F., III., Kang, A. S., Lerner, R. A., and Benkovic, S. J. (1991) Assembly of combinatorial antibody libraries on phages surfaces: the gene III site. *Proc. Natl. Acad. Sci. USA* **88,** 7987–7982.
14. Clackson, T., Hoogenboom, H. R., Griffiths, A. D., and Winter, G. (1991) Making antibody fragments using phage display libraries. *Nature* **352,** 624–628.
15. Hoogenboom, H. R., Griffiths, A. D., Johnson, K. S., Chiswell, D. J., Hudson, P., and Winter, G. (1991) Multi-subunit proteins on the surface of filamentous phage: methodologies for displaying antibody (Fab) heavy and light chains. *Nucleic Acids Res.* **19,** 4133–4137.
16. Marks, J. D., Hoogenboom, H. R., Bonnert, T. P., McCafferty, J., Griffiths, A. D., uand Winter, G. (1991) By-passing immunization: human antibodies from V-gene libraries diplayed on phage. *J. Mol. Biol.* **222,** 581–597.
17. Hust, M., Toleikis, L. and Dübel, S. (2007) Antibody phage display. Handbook of therapeutic antibodies, Ed. Dübel, S., Willey-vct 45–68.
18. Welschof, M., Terness, P., Kipriyanov, S., et al. (1997) The antigen binding domain of a human IgG-anti-F(ab′)2 autoantibody. *Proc. Natl. Acad. Sci. USA* **94,** 1902–1907.
19. De Haardt, H. J., van Neer, N., Reurst, A., et al. (1999) A large non-immunized human Fab fragment phage library that permits rapid isolation and kinetic analysis of high affinity antibodies. *J. Biol. Chem.* **274,** 18,218–18,230.
20. Pini, A., Viti, F., Santucci, A., et al. (1998) Design and use of a phage display library. *J. Biol. Chem.* **273,** 21,769–21,776.
21. Hoet, R. M., Cohen, E. H., Kent, R. B., et al. (2005) Generation of high-affinity human antibodies by combining donor-derived and synthetic complementarity-determining-region diversity. *Nat. Biotechnol.* **23,** 344–348.

22. Hayashi, N., Welschoff, M., Zewe, M., et al. (1994) Simultaneous mutagenesis of antibody CDR regions by overlap extension and PCR. *Biotechniques* **17,** 310–316.
23. Knappik, A., Ge, L., Honegger, A., et al. (2000) Fully synthetic human combinatorial antibody libraries (HuCAL) based on modular consensus framework and CDRs randomized with trinucleotides. *J. Mol. Biol.* **296,** 57–86.
24. Sblattero, D. and Bradbury, A. (2000) Exploiting recombination in single bacteria to make large phage antibody libraries. *Nat. Biotechnol.* **18,** 75–80.
25. Hust, M. and Dübel, S. (2004) Mating antibody phage display with proteomics. *Trends Biotechnol*. **22,** 8–14.
26. Parmley, S. F. and Smith, G. P. (1988) Antibody selectable filamentous fd phage vectors: affinity purification of target genes. *Gene* **73,** 305–318.
27. Hawlisch, H., Müller, M., Frank, R., Bautsch, W., Klos, A., and Köhl, J. (2001) Sitespecific anti-C3a receptor single-chain antibodies selected by differential panning on cellulose sheets. *Anal. Biochem.* **293,** 142–145.
28. Moghaddam, A., Borgen, T., Stacy, J., et al. (2003) Identification of scFv antibody fragments that specifically recognise the heroin metabolite 6-monoacetylmorphine but not morphine. *J. Immunol. Methods* **280,** 139–155.
29. Hust, M., Maiss, E., Jacobsen, H. -J., and Reinard, T. (2002) The production of a genus specific recombinant antibody (scFv) using a recombinant Potyvirus protease. *J. Virol. Methods* **106,** 225–233.
30. Konthur, Z., Hust, M., and Dübel, S. (2005) Perspectives for systematic in vitro antibody generation. *Gene* **364,** 19–29.
31. Kirsch, M., Zaman, M., Meier, D., Dübel, S., and Hust, M. (2005) Parameters affecting the display of antibodies on phage. *J. Immunol. Methods* **301,** 173–185.
32. Rondot, S., Koch, J., Breitling, F., and Dübel, S. (2001) A helper phage to improve single-chain antibody presentation in phage display. *Nat. Biotechnol.* **19,** 75–78.
33. Soltes, G., Hust, M., Ng, K.K.Y., et al. (2007) On the influence of vectar design on antibody phage display. *Journal Biotechnology* **127,** 626–637.
34. Hust, M., Meysing, M., Schirrmann, T., et al. (2006) Enrichment of open reading frames presented on bacteriophage M13 using Hyperphage. *Biotechniques* **41,** 335-342.
35. Goletz, A., Cristensen, P. A., Kristensen, P., et al. (2002) Selection of large diversities of antiidiotypic antibody fragments by phage display. *J. Mol. Biol.* **315,** 1087–1097.
36. Finnern, R., Pedrollo, E., Fisch, I., et al. (1997) Human autoimmune anti-proteinase 3 scFv from a phage display library. *Clin. Exp. Immunol.* **107,** 269–281.
37. Mersmann, M., Schmidt, A., Tesar, M., et al. (1998) Monitoring of scFv selected by phage display using detection of scFv-pIII fusion proteins in a microtiter scale assay. *J. Immunol. Methods* **220,** 51–58.

15

A Bacterial/Yeast Merged Two-Hybrid System

Protocol for Yeast Screening With Single or Parallel Baits

Nadezhda Y. Tikhmyanova, Eugene A. Izumchenko, Ilya G. Serebriiskii, and Erica A. Golemis

Summary

The yeast two-hybrid system is a useful tool for identifying new protein–protein interactions, and for the dissection of previously identified interactions. An important issue in protein-interaction studies is frequently that of determining whether a protein associates specifically with one protein or domain of interest, or has a more promiscuous interaction profile. To help address this issue, the authors have created a new two-hybrid system, which can be used either in bacteria or in yeast to counterscreen against "decoy" baits in parallel with a primary screen, hence improving the power and specificity of the method. Protocols of this system for use in yeast are provided; a companion article, Serebriiski et al., describes alternative use of this system in bacteria.

Key Words: Decoy bait; false-positive; protein–protein interaction; yeast two hybrid; bacterial two hybrid system; proteomics, library screening.

1. Introduction

As it was first introduced in 1989 as a means to study protein–protein interactions ***(1)***, the two-hybrid system has evolved into a robust technology. The number of novel interactions detected through use of this system now number in thousands, based on the work of many individual investigators, and increasingly, the output from high-throughput proteomics projects (e.g., **refs. *2–7***). These efforts have begun to yield genome-wide protein interaction linkage maps, and together with other protein interaction detection technologies such as tandem affinity purification-mass spectrometry ***(8,9)*** are key components supporting the nascent field of systems biology (e.g., **refs. *10*** and ***11***). On a smaller scale, the two-hybrid system is a useful way for an individual scientist to gain some insight into the function of a poorly understood protein, by identifying

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

functionally characterized "interactors" for that protein. Numerous technical extensions and derivatives of the two-hybrid system paradigm have been developed, and two-hybrid interaction screening can now be performed not only in yeast, but also in bacteria or other organisms (e.g., **ref. *12***).

Despite these many accomplishments, there remain a number of technical limitations that restrict the utility of two-hybrid-based protein interaction data. One issue is that of false-positives. Estimates of the frequency of nonspecific "positives" obtained for a protein used in a two-hybrid library screen vary, but may be as high as 50% or more, in some particularly bad cases. A second issue is that of false negatives. Meta-analyses comparing the results of large- and small-scale two-hybrid screens, and studies comparing the results of two-hybrid and other protein-interaction techniques, have led to the clear realization that the two-hybrid system probably substantially undersamples the interactor pool for any given sample ***(11,13)***. There are various approaches to addressing these problems. In this and a companion chapter ***(16)***, one systematic approach, which is the development of a two-hybrid system variant with extended screening capacity and greater internal controls will be discussed.

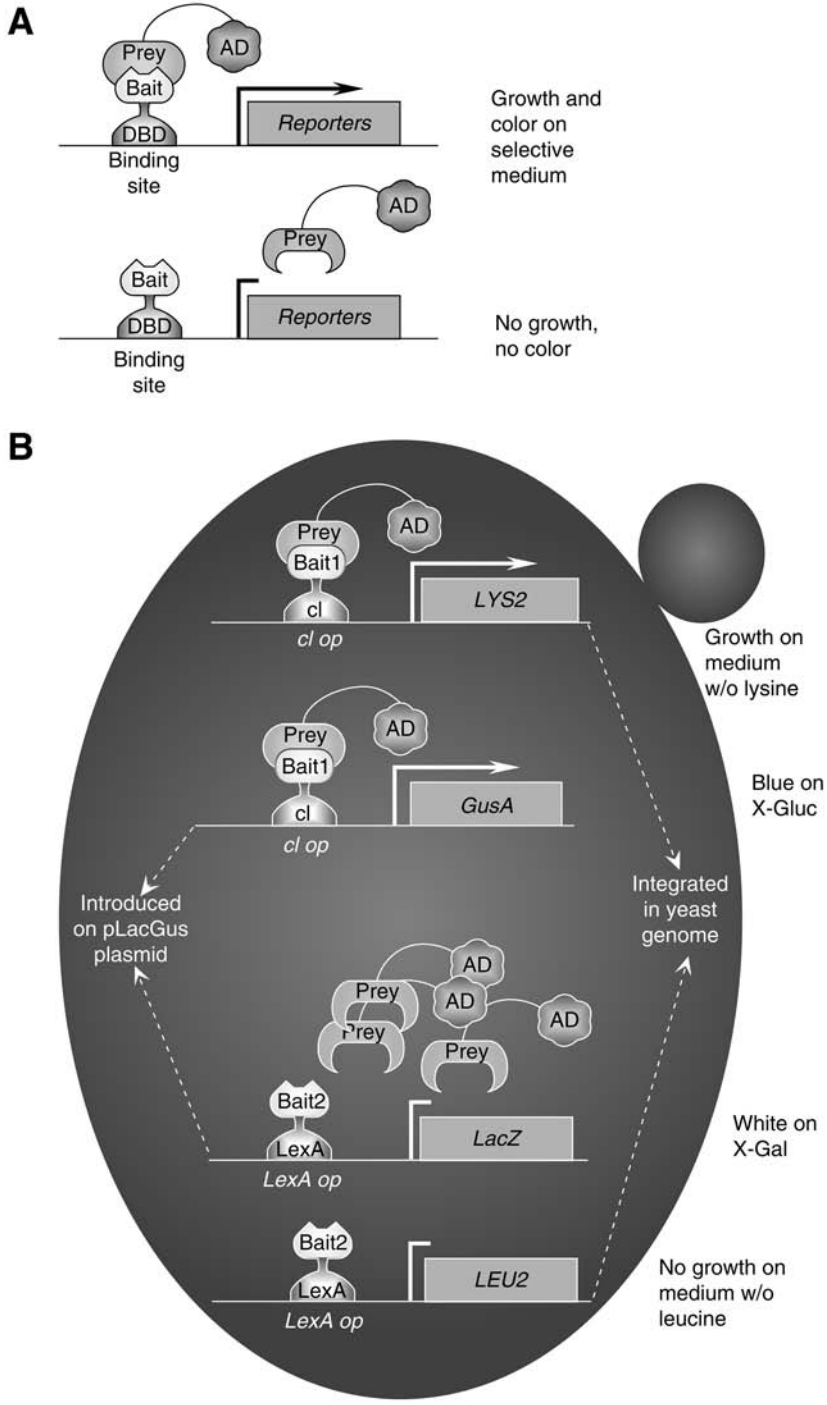
The basic yeast two-hybrid system paradigm is shown in **Fig. 1A**. In library screening with this system, a "bait" protein is made, in which a protein of interest is expressed as a chimera with a DNA-binding domain (DBD) of known sequence-binding specificity. For the "classic" two-hybrid system, this bait protein must be confirmed to lack transcriptional activating sequences; as such autoactivation will make it unusable in a library screen. To measure bait-dependent transcription, the bait is expressed in strains of yeast with reporter genes, wherein a binding site for the bait DBD is located within the promoter region of two reporter genes. These are typically a colorimetric reporter (*LacZ* and *GusA*) and an auxotrophic selection gene (*HIS3*, *LEU2*, and *LYS2*). In library screening, a library of "preys," representing a cDNA library expressed from a vector that fuses them to a transcriptional activation domain (AD), is introduced into yeast containing transcriptionally inactive baits. Interaction of an AD-fused library constituent with the bait turns on the reporter genes, allowing selection of positive clones.

In 1999, the "dual bait" two-hybrid system, which can be used to simultaneously analyze the interaction of two distinct baits with the same interactive

---

Fig. 1. *(Opposite page)* Two-hybrid system and dual bait system. An AD-fused protein (prey) interacts with a LexA-fused protein (bait1) to drive transcription of *LexA* op-responsive *LEU2* and *LacZ* reporters but does not interact with a cI-fused bait, and thus, does not turn on transcription of cI op-responsive *LYS2* and *GusA* reporters.

Note: as drawn herein, cI-fused bait is representing a nonspecific partner; the system can also be configured for the prey to interact with both baits. AD, activation domain.


A
Prey
AD
Bait
DBD
Binding site
Reporters
Growth and color on selective medium
AD
Prey
Bait
DBD
Binding site
Reporters
No growth, no color
B
Prey
AD
Bait1
cl
cl op
LYS2
Growth on medium w/o lysine
Prey
AD
Bait1
cl
cl op
GusA
Blue on X-Gluc
Introduced on pLacGus plasmid
Integrated in yeast genome
AD
Prey
AD
AD
Prey
Prey
Bait2
LexA
LexA op
LacZ
White on X-Gal
Bait2
LexA
LexA op
LEU2
No growth on medium w/o leucine

partner was first described ***(14,15)***. In the dual bait system (schematically shown in **Fig. 1B**), one protein of interest is expressed as a fusion to a DBD provided by λ bacteriophage cI (bait 1), whereas another is expressed as a fusion to a DBD provided by the bacterial protein LexA (bait 2). Four separate reporter genes are used to analyze the interaction between the two baits and preys. *GusA* and *LYS2* are transcriptionally responsive to an operator for cI (*cI op-GusA* and *cI op-LYS2*), whereas *LacZ* and *LEU2* are transcriptionally responsive to an operator for *LexA* (*LexA op-LacZ* and *LexA op-LEU2*). There are many advantages and potential uses for such a system, discussed at length in **refs. *14–17***. For the specific purpose of library screening, a major benefit is that a library can be screened to identify proteins that interact with bait 1, then immediately counterscreened to eliminate "positives" that also interact with bait 2, reducing the false-positive rate. "True-positives" would have a transcriptional activation phenotype such that expression of *LYS2* = *GusA* >> *LacZ* = *LEU2*.

In addressing the second problem, that of false-negatives, the authors and collaborators have exploited the fact that a given bait can identify nonequivalent sets of interactors when used for two-hybrid screening in bacteria vs in yeast ***(18)***. There a set of vectors (represented herein by the prototypic vector pGLS23) suitable for expressing baits in either bacteria or yeast have been described. It is proposed that screening baits constructed in these vectors in both organisms, and/or counterscreening using dual bait capacities in yeast, can significantly improve the power of two-hybrid library interrogation. The companion chapter (Chapter 16) describes the use of the bacterial system in greater detail. This chapter focuses on the use of the yeast system, using a counterscreen approach. It is noted, space limitations do not allow detailed presentation of basic auxiliary protocols related to execution of the technique (e.g., preparation of yeast medium, Western blotting, and so on). These are found in standard reference manuals, including **refs. *19*** and ***20***.

2. Materials

A complete table of reagents compatible with the bacterial/yeast and dual bait two-hybrid systems, together with acknowledgments to the numerous investigators whose work has contributed to the development of these tools, is available at http://www.fccc.edu/research/labs/golemis/interactiontrapinwork.html. Many of these reagents are available commercially, and also can be acquired by request from IG_Serebriiskii@fccc.edu, (215) 728-3885 phone, (215) 728-3616 fax, at Fox Chase Cancer Center (Philadelphia, PA).

2.1. Plasmids

1. pGLS23—the plasmid for making cI fusion protein (bait 1), **Fig. 2A**. Bait expression is from the constitutive alcohol dehydrogenase (ADH1) promoter. The yeast selection marker is *HIS5*, and the bacterial selective marker is Cm^R **(Note 1)**.

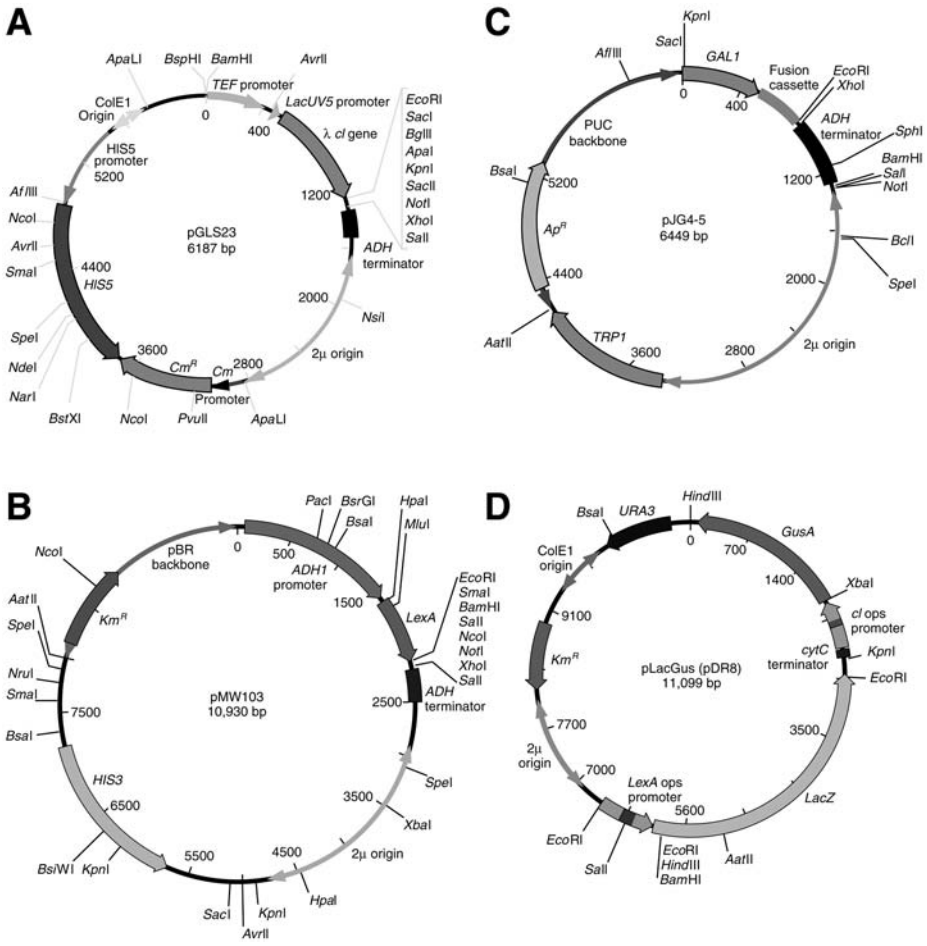


Fig. 2. Plasmid maps. **(A)** pGLS23, used for bait 1 expression. **(B)** pMW103, for bait 2 expression. **(C)** pJG4-5, used for library or defined interactor (prey) expression. **(D)** pLacGus (also known as pDR8), double reporter (cI-responsive *GusA* and *LexA*-responsive *LacZ*).

2. pMW103—plasmid for making LexA-fusion protein (bait 2) **(Fig. 2B)**. Expression is from the constitutive ADH1 promoter. The yeast selection marker is *HIS3* and the bacterial selective marker is Km^R.
3. pJG4-5—the plasmid for making a nuclear localization sequence—AD—hemaglutinin epitope tag fusion to a unique protein or a cDNA library ***(21)***, **Fig. 2C**. AD-fusion expression is from the galactokinase (GAL1) galactose-inducible promoter. The yeast selection marker is *TRP1*, and the bacterial selective marker is Ap^R.
4. pLacGus—the reporter plasmids containing 8 LexA operators upstream of the *LacZ* reporter gene, and 3 cI operators upstream of *GusA* reporter gene **(Fig. 2D)**. The yeast selection marker is *URA3*, and the bacterial selective marker is Km^R.

5. pGLS22-Ras—plasmid encoding cI-Ras, a negative control for activation and positive control for interaction. Selection markers are *HIS5* and Cm^R.
6. pGLS22-EE_{12345}L—a plasmid encoding cI-EE_{12345}L fusion, a strong positive control for activation. Selection markers are *HIS5* and Cm^R.
7. pEG202-Krit—HIS3 plasmid encoding LexA-Krit, a strong positive control for activation. The vector pEG202 is almost identical to pMW103 (yeast selection marker is *HIS3*), but the bacterial selective marker is Ap^R.
8. pEG202-Krev1—a plasmid encoding LexA-Krev1, a negative control for activation and positive control for interaction. Markers are *HIS3* and Ap^R.
9. pJG4-5 (Origene Technologies, Inc., Rockville, MD, as a part of DKT100 DupLEX-A Yeast Two-Hybrid System):Raf—library plasmid encoding a positive control for interaction with Ras. Selection markers are *TRP1* and Ap^R.
10. pJG4-5:Krit1—library plasmid encoding a positive control for interaction with Krev1. Selection markers are *TRP1* and Ap^R.
11. pYesTrp:RalGDS—*TRP1* library plasmid encoding a positive control for interaction with both Ras and Krev1. This plasmid is similar to pJG4-5, but has an extended polylinker and a V5-epitope tag instead of a hemaglutinin tag. Selection markers are *TRP1* and Ap^R.

2.2. Strains

1. Yeast strain PRT50 (*MATα URA3 TRP1 HIS3 2LexA op-LEU2 3cI op-LYS2*).
2. Yeast strain PRT475 (*MATa URA3 TRP1 HIS3 2LexA op-LEU2 3cI op-LYS2*).

2.3. Lithium Acetate Transformation of Yeast

1. 10 m*M* Tris-HCl, 1 m*M* ethylenediaminetetraacetic acid (EDTA), and 0.1 *M* lithium acetate, pH 8.0, sterile filtered.
2. 10 m*M* Tris-HCl, 1 m*M* EDTA, 0.1 *M* lithium acetate, and 40% PEG4000, pH 8.0, sterile filtered.
3. Dimethylsulfoxide (DMSO).
4. 6 mg/mL freshly denatured (i.e., boiled for 5 min and chilled on ice) sheared salmon sperm DNA (sssDNA).

2.4. Minipreps/Polymerase Chain Reaction From Yeast

1. Acid-washed sterile glass beads, 0.15–0.45 mm diameter (e.g., Sigma G-1145).
2. Tris EDTA solution TE:10 m*M* Tris-HCl and 1 m*M* EDTA, pH 8.0.
3. 1:50 β-glucuronidase type HP-2 (crude solution from *Helix pomatia* [Sigma]), 50 m*M* Tris-HCl, 10 m*M* EDTA, and 0.3% (v/v) 2-mercaptoethanol (prepare fresh), pH 7.5.

2.5. XGal/XGluc Overlay Assays

1. 1% Low-melting agarose in 100 m*M* $KHPO_4$, pH 7.0 agarose; add 5-bromo-4-chloro-3-indolyl-beta-D-galactopyranoside (XGal) or 5-bromo-4-chloro-3-indolyl-beta-D-**glucuronide** sodium salt (XGluc) (Diagnostic Chemicals, Oxford, CT) to 0.25 mg/mL when cooled to approx 60°C.
2. Chloroform ($CHCl_3$).

2.6. Media and Plates

1. Plates for growing bacteria (100 mm), Luria Bertani medium containing 50 µg/mL ampicillin.
2. Defined minimal yeast medium: all minimal yeast media, liquid, and plates used in this protocol are based on the following ingredients, which are sterilized by autoclaving for 15–20 min: 6.7 g/L yeast nitrogen base-amino acids (Difco 0919-15), 20 g/L glucose, or 20 g/L galactose plus 10 g/L raffinose, and 2 g/L appropriate nutrient "dropout" mix (*see* in next paragraph). For plates, 20 g Difco spark MDMI bacto agar (Difco 0140-01) are also added.

   A complete minimal nutrient mix includes the following: 2.5 g adenine, 1.2 g L-arginine, 6 g L-aspartic acid, 6 g L-glutamic acid, 1.2 g L-histidine, 1.2 g L-isoleucine, 3.6 g L-leucine, 1.8 g L-lysine, 1.2 g L-methionine, 3 g L-phenylalanine, 22 g L-serine, 12 g L-threonine, 2.4 g L-tryptophan, 1.8 g L-tyrosine, 9 g L-valine, and 1.2 g uracil. To make liquid media or plates for selection of yeast expressing plasmid selection markers or auxotrophic reporters, one or more ingredients are omitted from the complete minimal nutrient mix. Thus, "dropout medium" lacking histidine (denoted His^- in the following recipes) would select for the presence of plasmids with the *HIS3*, *HIS5*, double marker, and so on. *Note:* (1) the quantities of nutrients described above are enough to prepare 40 L of medium, which in most cases is more than will be required and (2) premade dropout mixes are available from some commercial suppliers.
3. Specific yeast liquid media and plates for this protocol (100 mm).
   a. **Yeast** extract, peptone and dextrese medium (YPD) (rich medium): 10 g/L yeast extract, 20 g/L peptone, and 20 g/L glucose, autoclave about 18 min. To make plates: add 20 g Difco bacto agar per liter of (unautoclaved) mix for liquid media, and autoclave about 18 min. 1 L makes approx 40 plates.
   b. Defined minimal dropout plates, with glucose as a carbon source: Trp^-; Ura^- His^-; Ura^- His^- Trp^-; Ura^- His^- Trp^- Leu^-; and Ura^- His^- Trp^- Lys^-.
   c. Defined minimal dropout media, with glucose as a carbon source:Ura^- His^-; and Trp^-.
   d. Defined minimal dropout plates, with galactose and raffinose as a carbon source: Ura^- His^-; Ura^- His^- Trp^- Leu^-; Ura^- His^- Trp^- Lys^-; Ura^- His^- Leu^-; Ura^- His^- Lys^-; and Ura^- His^- Lys^- Leu^- (this last, optional).
   e. Plates for growing yeast library transformations (240×240 mm^2): use minimal dropout plates (Trp^-), with glucose as a carbon source. Pour approx 250 mL medium on each plate.

2.7. Primers

1. *For cI-fusion plasmids:* forward primer, to confirm correct reading frame 5′-ATG ATC CCA TGC AAT GAG AG-3′.
2. *For LexA-fusion plasmids:* forward primer, to confirm correct reading frame 5′-CGT CAG CAG AGC TTC ACC ATT G-3′.
3. *For JG4-5 plasmid:* forward primer, FP1, 5′-CTG AGT GGA GAT GCC TCC-3′. Reverse primer, FP2, 5′-CTG GCA AGG TAG ACA AGC CG-3′.

4. *pYESTrp2 plasmid:* forward primer, FP1 can be used, or 5′-GATGTTAACGATACCAGCC-3′ (Invitrogen: Carlsbad, CA, recommended).
   Reverse primer 5′-GCG TGA ATG TAA GCG TGA C-3′.

2.8. Miscellaneous

1. Sterile glass beads, 3–4 mm in diameter, no. 3000, Thomas Scientific (Waltham, MA) 5663L19 or Fisher (Waltham, MA) no. 11-312A.
2. Sterile glycerol solution for freezing transformants (65% sterile glycerol, 0.1 *M* $MgSO_4$, and 25 m*M* Tris-HCl, pH 8.0).
3. Insert grid from a rack of pipet tips (Rainin RT series [Rainin Instrument, LLC, Oakland, CA], 200 µL capacity).
4. A metal frogger (e.g., Dankar Scientific, Reading, MA, no. MC48).
5. A plastic replicator (Bel-Art Products, Pequannock, NJ, no. 378776-0002, Bel-Blotter, or Fisher no. 1371213).
6. 2X Laemmli sample buffer (0.125 *M* Tris-HCl, 4% [w/v] SDS, 20% [v/v] glycerol, 10% [v/v] 2-mercaptoethanol, and 0.002% [w/v] bromophenol blue, pH 6.8). Add 2-mercaptoethanol shortly before use, store at 4°C, and discard if color becomes orange.
7. Antibody to cI (Invitrogen, Santa-Cruz) and to LexA (Invitrogen).

3. Methods

3.1. Creating and Assessing the Bait

Before beginning to hunt for an interactor, it is necessary to construct plasmids that reliably express the proteins of interest (baits). Baits are expressed as fusions to the λ phage protein cI and/or to the bacterial protein LexA. These plasmids are then transformed ***(22)*** into a yeast reporter strain to assess the suitability of the bait proteins for library screening. Yeast colonies containing baits are then tested to determine whether the baits are appropriately synthesized, and do not exhibit self-activation (trigger the transcription of reporter genes on their own) and are not toxic. For these purposes, they are compared with previously established controls **(Table 1)**. If all these requirements are not met, there are strategies that can be used to modify the bait(s) or screening conditions ***(23)***. Rapid movement through the characterization steps is recommended before starting a library screen, to diminish artifacts and avoid other difficulties. Although plasmids will be retained for extended periods of time in yeast maintained on stock plates, using freshly transformed colonies for all experiments (<10 d to 2 wk, with plates maintained at 4°C) is suggested, as it is much more likely that variable protein expression and anomalous transcriptional activation results will occur if using older transformed stocks.

3.1.1. Constructing and Transforming cI and LexA Bait Proteins

1. Clone (*see* **Note 1**) the DNA encoding the protein of interest into the polylinker of pGLS23 **(Fig. 3A)** to enable synthesis of an in-frame protein fusion to cI (*see* **Notes 1** and **2**).

Table 1
Expected Phenotype for Control Interactions

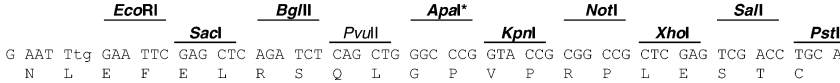
Bait	Prey	*LEU2*	*LacZ*	*LYS2*	*GusA*	Explanation
cI-Ras + LexA – Krev1	Krit	+	+	–	–	Prey interacts with LexA-fused bait only
	RalGDS	+	+	+	+	Prey interacts with both baits
	Raf1	–	–	+	+	Prey interacts with cI-fused bait only
cI-bait1 + LexA – bait2	Random	–	–	–	–	Prey does not interact specifically with bait

Adapted from **ref. *23***.

Each feature marked as positive (+) should be also galactose-dependent, as in **Table 2**, top row. Comparing the phenotype of new baits of interest to this set of controls should help to assess whether an isolated prey interacts with one or both baits.

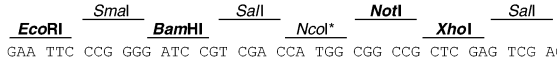
2. Clone the DNA encoding the second protein of interest into the polylinker of pMW103 **(Fig. 3B)** to enable synthesis of an in-frame protein fusion to LexA (*see* **Notes 2** and **3**). If the main research goal is to use the second bait as a library counterselection, rather than to perform a second library screen, a well-characterized LexA-fusion such as pEG202-Krev1 can be used in place of a newly cloned fusion.
3. Select a colony of PRT50 (*see* **Note 4**) and grow a 5 mL culture in liquid YPD medium overnight at 30°C in a shaking incubator.
4. Dilute into 50–60 mL of YPD liquid medium such that the culture has an optical density $(OD)_{600}$ nm of approx 0.15. Continue to incubate at 30°C on an orbital shaker until the culture has reached an OD_{600} nm of 0.5–0.7. This is sufficient yeast for 10 transformations.
5. Transfer culture to a sterile 50-mL Falcon tube, and centrifuge for 5 min at 1000–1500*g* at room temperature. Gently resuspend the pellet in 5 mL of sterile water.
6. Centrifuge the cells for 5 min at 1000–1500*g*. Pour off the water and resuspend the yeast pellet in 0.5 mL of TE/0.1 *M* lithium acetate.
7. Aliquot 1 µg of freshly sheared, denatured salmon sperm DNA to 1.5-mL Eppendorf tubes.
8. Add 50 µL of competent yeast cells from **step 6** to each tube. Add the following combinations of cI-fusion, LexA-fusion, and reporter plasmids (100–500 ng each):
   a. pGLS23-Bait1 + pLacGus + pMW103-Bait2 (test for autoactivation).
   b. pGLS22-Ras + pLacGus + pEG202-Krev1 (negative controls for autoactivation).
   c. pGLS22-EE_{12345}L + pLacGus + pEG202-Krit (strong positive controls for autoactivation).
9. To each tube, add 300 µL of sterile 40% (w/v) PEG 4000/0.1 *M* lithium acetate/TE buffer, pH 7.5. Invert several times to mix (do not vortex). Incubate the tubes at 30°C for 30–60 min.
10. Add 40 µL of dimethyl sulfoxide to each tube, mix by inversion. Heat shock the tubes by incubating at 42°C (in a heat block) for 10 min.

A **Polylinker for pGLS23 *cI*-fusion vector:**



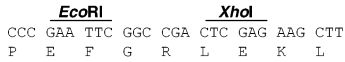
B **Polylinker for pMW103 *LexA*-fusion vector:**

Also applies for related vectors pEG202 (pLexA, displayBait) and pGilda



C **Polylinker for AD-fusion library vectors:**

pJG4-5 (pB42AD, displayTarget)



pYesTrp2

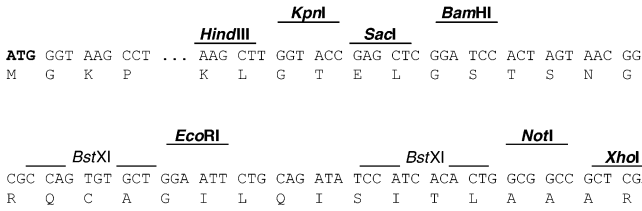


Fig. 3. Polylinkers of basic two-hybrid vectors. **(A)** pGLS23. **(B)** pMW103. **(C)** pJG4-5. Maps and sequences for these and additional vectors are available on the web at http://www.fccc.edu/research/labs/golemis/InteractionTrapInWork.html. Only restriction sites that are available for insertion of coding sequences are shown; those shown in bold type are unique.

11. Microfuge the cells for 20 s at 10,000–15,000*g*. Pour off the supernatant and resuspend the yeast in 0.5 mL of sterile water.
12. Spread each transformation mixture on Glu/CM Ura$^-$ His$^-$ dropout plates, and keep at 30°C for 2 d to select for yeast colonies containing transformed plasmids **(Note 5)**.

3.2. Replica Technique/Gridding Yeast: Assessing Bait Activation of Reporters

For each transformation: pick and analyze at least six independent colonies for their transcriptional activation phenotype, using the auxotrophic and colorimetric

reporters (*see* **Note 6**). Assessment of transcriptional activation requires the transfer of yeast from master plates to a variety of selective (dropout) plates. A sterile toothpick is usually used to move cells from individual patches on the master plate to each of the selective media. In some cases (particularly in genomic-scale applications) a large number of colonies expressing numerous combinations of bait and prey need to be examined. In this case especially, it is useful to use a transfer technique that is made for a high-throughput analysis, such as the one described herein.

1. Add approx 50 µL of sterile water to the wells of 96-well microtiter plate with a syringe-based repeater or multichannel pipet (e.g., use wells A1–C6 for six colonies each of the three transformations described in **Subheading 3.1.1., step 8**). Position a micropipet tip insert grid on the microtiter plate, and attach it with tape: the holes in the insert grid should be placed exactly over the wells of the microtiter plate (this is important for stabilization of the tips in the plate, and will allow simultaneous removal afterward, thereby speeding the replica process).
2. Use sterile plastic micropipet tips (or toothpicks) to pick six yeast colonies (1–2-mm diameter) from each of the transformation plates **a–c** (*see* **Subheading 3.1.1., step 12**). Place the tips in the wells leaving them in a near-vertical position supported by the insert grid until all the colonies have been picked.
3. Swirl the plate gently to mix the yeast into suspension and remove the insert grid, thereby removing all the tips at once.
4. Use a replicator to plate (*see* **Note 7**) yeast suspensions to new plates. Each spoke will leave a drop approximately equal to a 3 µL volume. Use the following plates:
   a. One Glu/CM Ura^- His^- (producing a master plate).
   b. Two Gal-Raff/CM Ura^- His^- plates (for X-Gluc and X-Gal overlay assays, to test for *GusA* and *LacZ* reporter activity, respectively).
   c. One Gal-Raff/CM Ura^- His^- Lys^- (for scoring activation of the *LYS2* reporter).
   d. One Gal-Raff/CM Ura^- His^- Leu^- (for scoring activation of the LEU2 reporter).
   e. One Gal-Raff/CM Ura^- His^- Lys^- Leu^- (optional).
5. Grow the plates at 30°C. After 1–2 d, put the Glu/CM Ura^- His^- master plate at 4°C, and assay the two Gal-Raff/CM Ura^- His^- plates for the activation of *GusA* and *LacZ*. Grow the remaining plates at 30°C until very strong growth is observed on the three *LYS2* and *LEU2* selection plates by the positive controls (for up to 4 d, but typically within 2 d).
6. Activation of the *GusA* and *LacZ* reporters is assessed qualitatively, using the yeast grown for 18–36 h on the two Gal-Raff/CM Ura^- His^- plates. Use one plate for overlay with XGal agarose, and the second for overlay with XGluc agarose, as follows.
   a. Slowly release approx 5 mL chloroform ($CHCl_3$) from a glass pipet held near the inside of the plate (or slowly pour from a small bottle). The objective is not to smear the colonies by too vigorous a release of $CHCl_3$. Do not cover plate with lid. Incubate colonies completely covered in $CHCl_3$ for approx 5 min.
   Caution: $CHCl_3$ is a toxic chemical. Take precautions to avoid inhalation and skin contact. Wear gloves. The procedure must be done in a chemical hood.

To minimize the amount of $CHCl_3$, use just enough to cover the colonies; try to avoid extended contact with the walls of the plate, as $CHCl_3$ dissolves plastic.

b. (Optional) Briefly rinse the plates with another approx 5 mL $CHCl_3$, then drain and let dry, uncovered, for another 5 min at 37°C, or for 10 min in a chemical hood at room temperature.
c. Carefully add about 10 mL of XGal- or X-Gluc agarose on the plate, making sure that all yeast spots are completely covered. Note that it is difficult to spread less than 7 mL of agarose because the plates chill as $CHCl_3$ evaporates.
d. Put plates in 30°C incubator and keep track of color changes. Checking the plates at 20 min, 1 h, and 3 h after agarose addition is recommended. Yeast colonies containing positive control baits (transformation **c**) as well as test baits that strongly activate *GusA* and *LacZ* reporters will become dark blue colonies in 20–60 min, whereas negative controls (transformation **b**) should remain as faint blue or white colonies for several hours. An optimal bait should be either comparable with the negative control or develop faint blue color.

7. Next, at days 3–4 after plating, analyze the transformants for transcriptional activation of reporter genes *LYS2* and *LEU2*, which enable growth of the transformed autotrophic yeast strain on selective media. Yeast containing both cI-and LexA-fused test baits (from transformation **a**) and negative controls (from transformation **b**) should not grow on Gal-Raff/CM Ura^- His^- Leu^-, Ura^- His^- Lys^-, or Ura^- His^- Lys^- Leu^- plates **(Note 8)**. The most important sign that baits may be suitable for screening libraries is the absence of growth similar to negative activation control. If the tested baits grow as well as the positive control for activation (from transformation **c**) they may not be used for library screening, as they are likely to produce high background (*see* **Note 9**).

3.2.1. Detection of Bait Protein Expression

In general, it is recommended to evaluate the expression level and appropriate size of the bait proteins by Western blot analysis, even if the bait is well behaved in the activation assays (*see* **Note 10**).

1. For each bait (test and control), pick at least two primary bait/reporter transformants from the Glu/CM Ura^- His^- master plate (*see* **Subheading 3.2., step 5**), and inoculate them into Glu/CM Ura^- His^- liquid medium. Grow overnight (8–12 h) on an orbital shaker at 30°C. Dilute the saturated cultures into 2 mL of the same medium at a density of approx 0.15 OD_{600}, and grow at 30°C (*see* **Note 11**).
2. After incubating for 4–6 h the OD_{600} of the cultures should reach 0.45–0.7 (measure before harvesting). Centrifuge 1.5 mL of each culture at 13,000*g* for 3–5 min in a microfuge. When each cell pellet is visible (should be approx 2–5 µL of packed cell volume), carefully aspirate the supernatant.
3. Add 50 µL of 2X Laemmli sample buffer to each pellet, and rapidly mix by vortexing to resuspend each pellet. Boil the samples at 100°C for 5 min for immediate assay, or freeze at –70°C (in dry ice) for subsequent use **(Note 12)**.

4. After boiling, chill the samples on ice and centrifuge for 30 s at 13,000*g* to pellet large cell debris. Load 10–25 µL of each sample onto a 0.1% (w/v) sodium dodecyl sulfate-polyacrylamide gel.
5. Prepare a Western blot and use an antibody to cI to analyze cI-fusion (bait 1) expression. Subsequently, strip the blot and probe with antibody to LexA to screen for LexA-fusion (bait 2) expression **(Note 13)**.

3.3. Transforming a Library, and Characterizing Interactors

A partial list of available libraries compatible with the interaction trap is found at http://www.fccc.edu/research/labs/golemis/InteractionTrapInWork.html. Currently, the majority of libraries suitable for the two-hybrid reagents described herein are available commercially, through sources including Origene (Rockville, MD) and Invitrogen (from a noninducible promoter). If one wishes to make one's own library, it should be cloned in a vector such as pJG4-5 or a related vector such as pYESTRP2 (Invitrogen). The polylinker sequence at the site of cDNA insertion for the vector pJG4-5 is shown in **Fig. 3C**.

The protocol outlined next is designed with the goal of performing a screen, which should saturate a cDNA library derived from a genome of mammalian complexity (*see also* **Figs. 4** and **5**, for flow charts). Fewer plates will be required for screens with libraries derived from organisms with less complex genomes, and researchers should scale back accordingly. A protocol is provided for transforming the library into PRT475 yeast, then using mating ***(24)*** to introduce the library against the bait of interest by crossing bait-containing PRT50 with library-containing PRT475 yeast. The main advantage of this approach (as opposed to directly transforming the library into yeast containing the bait), is that if the investigator wishes to use the same library to screen multiple baits, only a single large-scale transformation is required, followed by relatively easy mating steps (*see* **Note 14**).

In order to obtain a clear estimate of the frequency of cDNA-independent false-positives (a frequency that is important to know when deciding how many positives to pick and characterize), it is a good idea to perform a small-scale parallel "test mating" for new bait strains with the PRT475 yeast containing only the library vector. This mating can be performed at the same time as the library mating, and both matings can be treated identically in the next step, selecting interactors.

A positive control is usually quite useful in a subsequent characterization of potential interactors, and can also be performed in parallel with the library transformation. Normally, these positive controls will interact with either or both of the baits expressed in the PRT50 control strain obtained in **Subheading 3.1.1.**, transformation **b**, i.e., (pEG202-Krev1 + pLacGus + pGLS22-Ras). For the experiments described herein, pJG4-5:Raf will interact with pGLS22-Ras, pJG4-5:Krit1 will interact with pEG202-Krev1, and pYesTrp:RalGDS will interact with both.

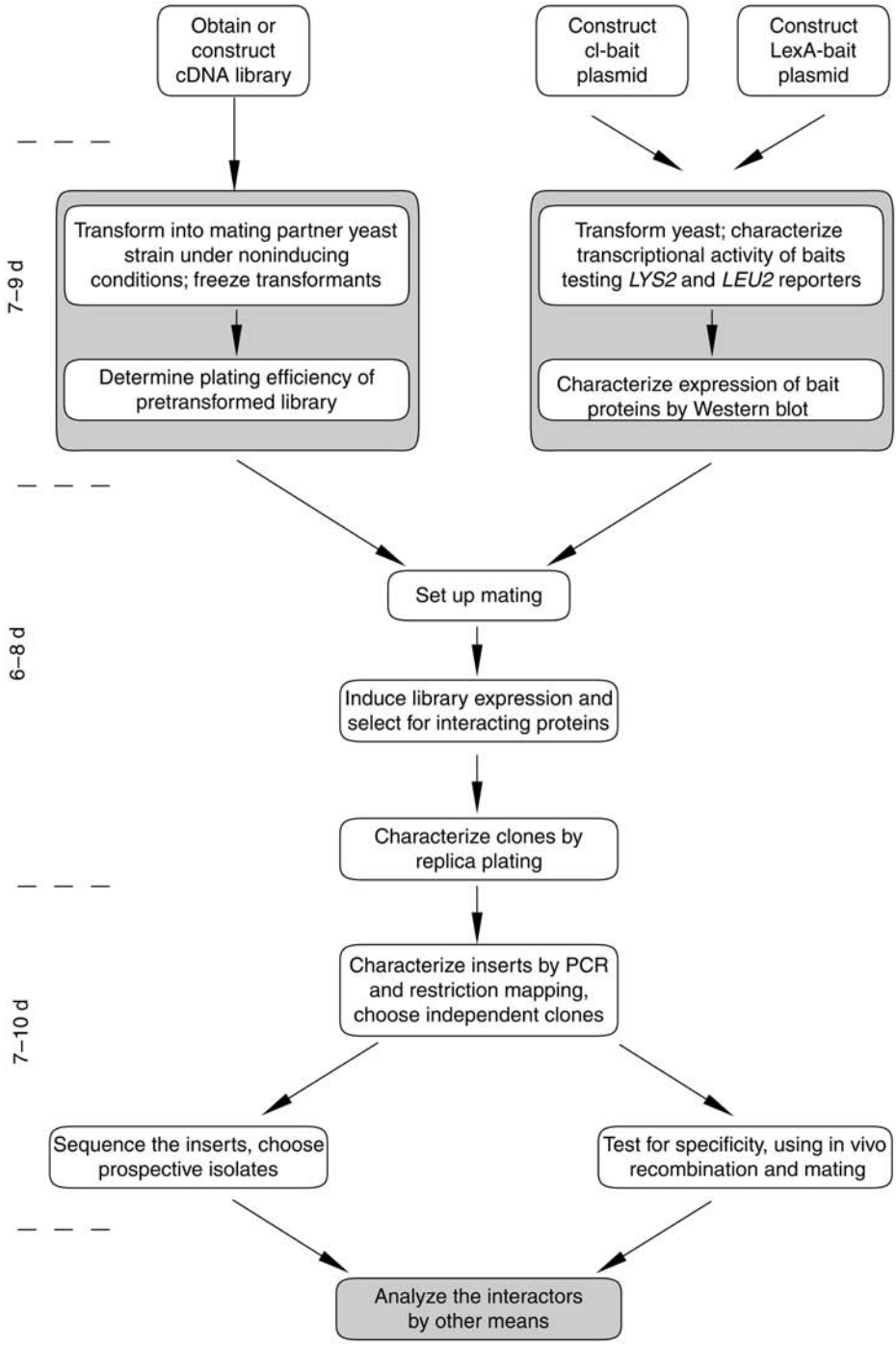


Fig. 4. Flow chart for a two-hybrid screen done by interaction mating. Stage three allows flexibility in step order, *see* text for details.

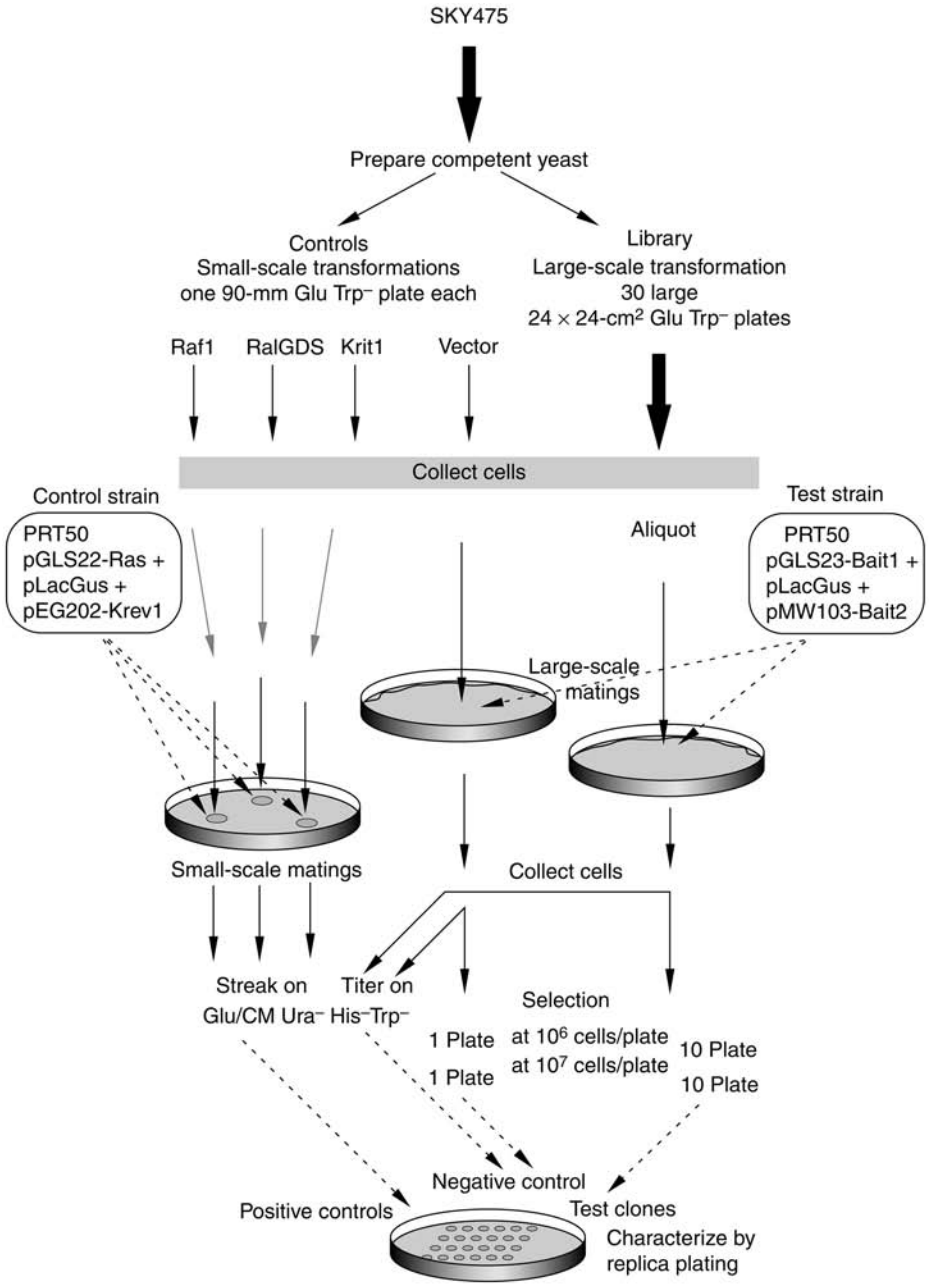


Fig. 5. Detailed library screening flow chart. *See* text for details.

3.3.1. Transforming the Library

1. Use a fresh colony of PRT475 to inoculate a flask containing 20 mL of liquid YPD medium. Grow yeast overnight at 30°C in an orbital shaker (*see* **Note 4**).
2. Dilute the overnight culture into 300 mL of fresh YPD liquid medium in a 1-L flask such that the diluted culture has an OD_{600} nm of 0.15, and grow as before until the culture has reached an OD_{600} nm of 0.5–0.7 (about 4–6 h).
3. Subdivide the 300 mL of culture among six 50-mL sterile disposable tubes, and centrifuge at 1000–1500*g* for 5 min at room temperature. Gently resuspend each pellet in 5 mL of sterile water, and combine all the slurries in a single tube. Add sterile water to the top of the tube and mix.
4. Recentrifuge the cells at 1000–1500*g* for 5 min at room temperature. Pour off the water, add 1.5 mL of TE/0.1 *M* lithium acetate and resuspend the remaining yeast pellet (~150 µL packed volume). Reserve 150 µL for use in **step 10** (parallel control transformations). Total volume should be approx 1.75 mL.
5. Mix 30 µg of library DNA and 1.5 mg of freshly denatured (i.e., boiled for 5 min) sssDNA and mix gently. Add the DNA mix to the yeast. Mix gently and dispense 60-µL aliquots of DNA/yeast suspension into 30 microfuge tubes (*see* **Note 15**).
6. To each tube, add 300 µL of sterile 40% (w/v) polyethylene glycol (PEG) 4000 prepared in 0.1 *M* lithium acetate/TE buffer, pH 7.5. Gently mix by inverting the tubes several times (do not vortex). Incubate the tubes at 30°C for 30–60 min.
7. Add 40 µL of dimethyl sulfoxide to each tube, and mix by inversion. Heat-shock cells by placing the tubes in a heat block set to 42°C for approx 10 min.
8. Pipet the complete contents of each tube onto each of 30 240×240 mm^2 Glu-Trp dropout plate, and spread the cells evenly using 12–24 sterile glass beads. Invert the plates without discarding glass beads (*see* **Note 16**) and incubate at 30°C until all colonies appear (within 3–4 d).
9. Select two representative transformation plates. Draw a 23×23 mm^2 (1% of the plate bottom surface) over an area containing an average density of colonies. Count the colonies in each grid section, and recalculate for the whole transformation. A good transformation performed according to this protocol should yield approx 20,000–40,000 colonies per plate.
10. Use small aliquots (~25 µL) of competent yeast from **step 4** to transform the empty library plasmid (pJG4-5 or pYesTrp2), pJG4-5:Raf1, pJG4-5:Krit1, and pYesTrp2:RalGDS. Plate each on a 100-mm plate, and collect the transformed cells as for the library (protocol outlined next), scaling down accordingly.

3.3.2. Harvesting and Pooling Primary Transformants

In the next step, a homogenized slurry is prepared (*see* **Note 17**) from the pool of approx 3×10^5–10^6 primary transformants, which is then aliquoted and frozen. Each of these aliquots is representative of the complete set of primary transformants, and can be used in subsequent mating.

1. Pour 10 mL of sterile water onto each of five 240×240 mm^2 plates containing transformants. Stack the five plates on top of each other. Holding on tightly, shake

the stack horizontally until all the colonies are in suspension (1–2 min). Using a sterile pipet, collect yeast slurry from each plate (by tilting the plates) and pool the resuspended transformants into a sterile 50-mL conical tube.
2. Repeat for further sets of five plates of transformants, resulting in a total of up to 150 mL of suspension split between three 50-mL tubes (*see* **Note 18**).
3. Fill each tube containing yeast to the top with sterile TE or water, and vortex/invert to suspend the cells. Spin cells down for 5 min at 1000–1500*g* at room temperature, and discard the supernatants. Repeat this step. After the second wash, the cumulative pellet volume should be approx 25 mL of cells derived from up to approx 10^6 transformants.
4. Resuspend each pellet in one volume of glycerol solution. Combine the contents of the three tubes and mix well. Freeze in 0.2–1-mL aliquots at –70°C. (These aliquots are stable for more than 1 yr. Refreezing a thawed aliquot results in the loss of viability, and is not advised.)

3.3.3. Mating the Bait Strain and the Pretransformed Library

Once the bait strain has been made and characterized, and the library strain has been transformed and frozen in aliquots, the next step is an interaction mating between the bait strain and an aliquot of the pretransformed library strain, followed by selection of positive interactors. In parallel, the bait strain is mated with a frozen aliquot of the negative control strain. When mating occurs, individual haploid cells of the bait strain fuse with individual haploid cells of the library strain to form a diploid yeast strain containing bait–prey combinations.

Practically, to mate the two strains, the bait strain and a pretransformed thawed aliquot of library or a control strain are mixed together, and incubated overnight on rich medium. To select for interactors, the diploids, along unmated haploid strains held in reserve as controls, are then recovered from the mating plates, and replated on media on which only diploids can grow (as described in **Subheading 3.3.4.**). In practice, a few aliquots of the diploid/haploid mixture are generally frozen in reserve, to allow tittering of mating efficiency, and repeated platings at various dilutions. Perform the mating with negative control strain (generated in **Subheading 3.3.1.**) at the same time as setting up the library interaction mating. For both matings, use the same techniques, and treat them identically in the next step (*see* **Subheading 3.3.4.**).

1. Start a 30-mL Glu/CM Ura$^-$ His$^-$ liquid culture of the bait strain from the Glu/CM Ura$^-$ His$^-$ master plate prepared in **Subheading 3.2.**, **step 5** (*see* **Note 19**). Grow with shaking at 30°C to mid- to late-log phase (OD_{600} nm = 1–2). *See* **step 6** for parallel bait controls.
2. Collect the cells by centrifuging at 1000*g* for 5 min at room temperature. Resuspend the cell pellet in 1 mL of sterile water and transfer to a sterile 1.5-mL microfuge tube. This will yield a yeast suspension of about 1×10^9 cells/mL.

3. At room temperature, thaw an aliquot of the pretransformed library and negative control library vector strain (*see also* other controls discussed in **step 6**). Mix 200 µL of the bait strain (~2×10^8 cells) with approx 10^8 cells of the pretransformed library (*see* **Subheading 3.3.2.**, **step 4**) or negative control strain on a single 100-mm YPD plate and incubate at 30°C for 12–15 h (overnight).
4. Add 1.5–2 mL of sterile water and 5–10 (3–4 mm) glass beads to the surface of each YPD plate, and suspend the cells as described for library transformation, i.e., by agitating the plate. Transfer the suspension to a sterile tube and vortex gently for 2 min. Collect the cells by centrifugation at 1000*g* for 5 min and resuspend in one volume of sterile glycerol solution. Distribute into 200-µL aliquots, and freeze at –80°C (*see* comment **Subheading 3.3.2.**, **step 4**). However, leave one aliquot unfrozen if one wishes to proceed directly to the next step—plating on selective medium (*see* **Subheading 3.3.4.**).
5. Titer the mated cells by thawing an aliquot (or using the unfrozen aliquot), and plating serial dilutions (made in sterile water) on Glu/CM Trp^- His^- Ura^- plates (this medium will not support the growth of the parental unmated haploids). Incubate plates at 30°C, and count the colonies that grow after 2–3 d on each plate, and determine the plating efficiency/colony forming units (CFUs) of the mated cells (*see* **Note 20**).
6. In parallel with **steps 1–4** above, grow up approx 1.5 mL of control bait strain (pGLS22-Ras + pLacGus + pEG202-Krev1) in Glu/CM Ura^- His^- and approx 1.5 mL cultures of each of the three control prey strains (*see* **Subheading 3.3.1.**, **step 10**) in Glu/CM Trp^-.
7. Take a YPD plate and make three spots of control bait strain by placing a drop (~5 µL) of the liquid culture on its surface. Without waiting for the liquid to soak in, overlay with 5 µL of one of the three control prey strains on each of the spots. Incubate overnight to allow mating, and then streak all three matings onto Glu/CM Ura^- His^- Trp^- plates to select diploids.

3.3.4. Screening for Interacting Proteins

In the next steps, interacting preys are selected by plating the mated cells onto auxotrophic selection plates. It is important to know how many viable diploids were plated onto these selection plates both to gain a sense of how much of the library has been screened (saturation) and to determine the false-positive frequency. This information is provided by the titer (expressed as CFU/mL), which indicates how successful the mating was (*see* **Subheading 3.3.3.**, **step 5**).

Dual bait reagents allow selection for an interaction with the cI-fused bait (on Gal-Raff/CM Ura^- His^- Trp^- Lys^- plates), or the LexA-fused bait (on Gal-Raff/CM Ura^- His^- Trp^- Leu^- plates). It also allows selection for preys that interact with both baits (on Gal-Raff/CM Ura^- His^- Trp^- Leu^- Lys^- plates). Negative selection (one interaction *but not* the second one) is theoretically possible, but impractical in a single step (but very feasible and recommended as a follow-up screen). Hence, one will plate the mated cells (**step 2** below) on

appropriate selective plates based on the purpose of one's screen. If the second bait is to be used mainly to increase the specificity of the primary screen, plate the mating on lysine selection plates only (and hold leucine testing for later). If one wishes to screen two independent baits, plate on two separate sets of lysine and leucine selection plates. Finally, if one is trying to identify proteins that interact with both baits, plating a fraction of the mating on lysine selection plates, and another fraction on double selection (Leu^- Lys^-) plates is suggested.

1. On the day the screen is scheduled, thaw an aliquot of the mated transformants. Dilute 100 µL into 10 mL of Gal-Raff/CM Ura^- His^- Trp^- liquid dropout medium, and incubate with shaking at 30°C for 5 h to allow yeast to begin active growth, and to induce the galactose-dependent expression of library proteins. If the frozen culture was not previously titered, plate serial dilutions onto Glu/CM Ura^- His^- Trp^- plates.
2. After 5–6 h of incubation, measure the OD_{600} of the culture. On the assumption that a culture at OD_{600} nm = 1 contains approx 1×10^7 cells/mL, plate 10^6 cells on five 100-mm plates with the appropriate auxotrophic selection medium. Plate 10^7 cells on each of five additional plates with the same medium. Generally, the plating of 10^6 cells/plate yields the best result. Although, plating cells at 10^7cells/plate density greatly reduces the number of plates that must be processed, allowing one to more thoroughly saturate the library, it may or may not contribute to cross-feeding between yeast, resulting in spurious background growth. Therefore, initially plating mated yeast at two different cell densities and comparing the results is recommended.
3. Place the plates in a 30°C incubator for up to 6 d, inspecting cell growth regularly (*see* **Note 21**). Depending on the individual bait used, good candidates for positive interactors will generally produce *LYS*$^+$ colonies over this time period, with the most common appearance of colonies at days 3–5. *LEU*$^+$ colonies typically form at days 2–4 (*see* **Note 22**).
4. Observe the plates on a daily basis. On the first day that colonies are visible by eye, mark their location on the plate with dots of a given color. Monitor the appearance of the colonies over 5 d. Each day, mark further colonies arising with different colors. At day 4 or 5, streak colonies in a microtiter plate format onto a solid master plate (Glu/CM Ura^- His^- Trp^-), in which colonies are grouped according to the day on which they appeared (*see* **Note 23**). If many apparent positives appear, it might be necessary to pick separate master plates for colonies obtained on day 2, on day 3, and on day 4, respectively: be sure to include controls **(step 5)** on these plates.

   It is important to compare selection plates seeded with lower and higher densities. A "lawn" should not be evident on either class, and the number of colonies should be roughly proportional to the seeding density. If no background growth appears on the more densely seeded plates, this strategy is successful as a means of more efficiently screening more of the library. If one gets disproportionally more colonies on the more densely seeded plates (especially sitting on a thin lawn), this is probably background owing to cross-feeding. In this case, take another aliquot of

the frozen mating, repeat induction and plate at 1×10^6 cells per plate on as many plates as are necessary for full representation of the calculated number of diploids.
5. *Controls:* include the positive control colonies (from mating with the control bait strains) on each of the master plates. Also, it is appropriate at this time to generate additional negative controls for subsequent steps by picking at random a few colonies from the titer plate (**step 1** above) and streaking them in parallel on the master plates to be tested in the next steps. As these contain randomly chosen library plasmids, the transcriptional activation phenotype of these colonies is most likely to be negative: if not, it is necessary to be extremely skeptical of the validity of predicted interactors.
6. Incubate the master plates at 30°C until patches/colonies form (overnight).

3.3.5. First Confirmation of Positive Interactions

The following steps test the specificity of positive interactors, assessing the activation of both the auxotrophic and colorimetric reporters in a galactose-dependent fashion. Simultaneously, galactose-inducible activation of both reporters generally indicate that the transcriptional phenotype is attributable to expression of library-encoded proteins, rather than derived from mutation of the yeast.

1. Invert a replicator, (*see* **Note 7**) on a flat surface, and place a master plate upside down on the spokes, making sure that the spokes and colonies are properly aligned. Remove the plate and insert the replicator into a microtiter plate containing 50 µL of sterile water in each well. Let the plate sit for 5–10 min, shaking from time-to-time to resuspend the cells left on the spokes. When all yeast are resuspended, print on the following plates (*see* **Notes 24** and **25**):
   a. *Master plate:* Glu/CM Ura$^-$ His$^-$ Trp$^-$.
   b. *Test for activation of* LYS2*:* Gal-Raff/CM (Ura$^-$ His$^-$ Trp$^-$) Lys$^-$ and Glu/CM (Ura$^-$ His$^-$ Trp$^-$) Lys$^-$.
   c. *Test for activation of* LEU2*:* Gal-Raff/CM (Ura$^-$ His$^-$ Trp$^-$) Leu$^-$ and Glu/CM (Ura$^-$ His$^-$ Trp$^-$) Leu$^-$.
   d. *Two sets of plates, to be assayed for* LacZ *and* GusA *activation:* Glu/CM Ura$^-$ His$^-$ Trp$^-$ and Gal-Raff/CM Ura$^-$ His$^-$ Trp$^-$.
2. Repeat for each master plate (from **Subheading 3.3.4.**, **step 4**).
3. Incubate the plates at 30°C. After 18–36 h of incubation, take out all Ura$^-$ His$^-$ Trp$^-$ plates. Keep one of the Glu/CM Ura$^-$ His$^-$ Trp$^-$ plates as a fresh master plate. Overlay the remaining two sets with XGal or XGluc agarose, as described in **Subheading 3.2.**, **step 6**. Continue to monitor growth on the Leu$^-$ and Lys$^-$ plates 48–72 h after plating. For comments on interpretation of the results, refer to **Table 2**.

3.3.6. DNA Isolation, and Second Confirmation of Positive Interactions

Execution of the aforementioned protocols for any given bait will result in the isolation of between zero and hundreds of potential "positive" interactors (*see* **Note 25**). These positives must next be evaluated for reproducible phenotype,

Table 2
Interpretation of Primary Isolates' Behavior

Observed phenotype						
Auxotrophic reporter		Colorimetric reporter		Interpretaion		
Glu	Gal	Glu	Gal	*Conservative*	*Optimistic*	Recommendation
–	+	–	+		Very good sign	Work with those clones first
(+)	+	(+)	+	Bait is mutated or its expression is upregulated, causing a high background of transcriptional activation	• *GAL1* promoter is slightly leaky • both bait and prey are very stable • interaction occurs with high affinity	Take a small number of clones (six or less) for confirmation of interaction; store the rest
–	+	–	–	Yeast mutation occurred that favors growth or transcriptional activation on galactose medium	Some bait-interactor combinations are known to preferentially activate one reporter versus another	If all other candidates fail, check these clones (or redo the screen with different bait, library, and so on)
All other phenotypes				Contamination/ plasmid rearrangements/ mutations	Something really new	Trash

From http://www.fccc.edu/research/labs/golemis/interactiontrapinwork.html.

and specific interaction with the bait used to select them, using a strategy as shown in **Fig. 6**. If a large number of positives are obtained, the subsequent characterization steps require prioritization. In this case, select up to about 24–48 independent colonies (preferably, those growing the soonest after plating on selective media) for the first round of assessment, while maintaining master plates of additional positives at 4°C. This first analysis set will be tested for specificity of interaction (i.e., for their ability to bind unrelated baits in addition to the original one) and screened by polymerase chain reaction (PCR)/restriction digest analysis and/or sequencing to establish whether clusters of frequently isolated cDNAs are obtained: such clusters are generally a good indication for a specific interaction.

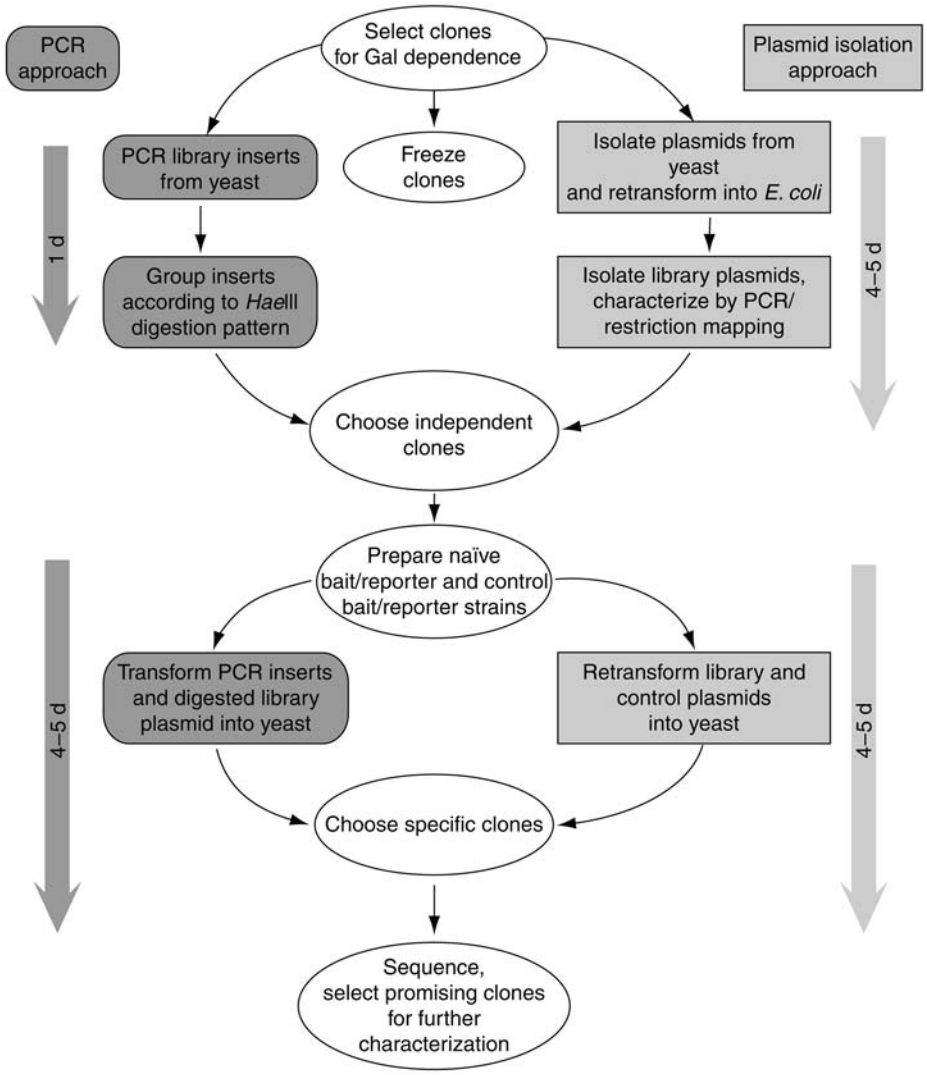


Fig. 6. Detailed flow chart for characterization and second confirmation of primary positives. *See* text for details.

The protocols described next provide two different approaches for analyzing positives: the first is encouraged. These approaches are summarized in the flow chart in **Fig. 6**. Both utilize similar methods, but the order with which techniques are applied differs. The choice between strategies depends on whether the individual investigator would rather spend time and money doing bulk PCR

(option 1) or bulk yeast plasmid recovery (option 2). The first protocol is generally 1–3 d faster, but does not result in bankable plasmids.

3.3.6.1. PCR Approach: Rapid Screen for Interaction Trap Positives

A major strength of the protocol described next is that it will identify redundant clones before plasmid isolation and bacterial transformation, which in some cases greatly reduces the amount of required work. Accurate records should be maintained of how many of each class of cDNA are obtained; and if any ambiguity is present about whether a particular cDNA is part of a set or is unique, investigators should err on the side of caution.

The outlined protocol includes steps of enzymatic treatment to generate crude yeast lysates **(steps 1–3)**, used later as template for the PCR reaction **(step 4)**. PCR product can be obtained directly from the yeast colonies even without β-glucuronidase treatment (e.g., by introducing a 10-min 94°C step at the beginning of the PCR program). However, yeast lysates obtained in this protocol also can be used as a source of plasmid for electroporation into *Escherichia coli*, instead of the more time-consuming plasmid recovery protocol described next.

1. Starting from the Glu/CM Ura$^-$ His$^-$ Trp$^-$ master plate, resuspend yeast in 25 µL of β-glucuronidase solution in a 96-well microtiter plate by using a replicator. Seal the wells using tape, and incubate on a horizontal shaker at 37°C for 1.5–3.5 h (*see* **Note 26**).
2. Remove the tape, and add about 25 µL of glass beads to each well, and seal again. Attach the microtiter plate to a vortex with a flat top surface (e.g., using rubber bands) and mix vigorously for 5 min.
3. To each well, add 100 µL of sterile distilled water. Take 0.8–2 µL as a template for each PCR reaction. Reseal the plate with tape, and keep the remainder frozen at –70°C.
4. PCR amplification:
   a. For PCR amplification use primers specific for the library plasmid used (*see* **Subheading 2.7.**). Perform PCR amplification (in ~30 µL volume) as follows (*see* **Notes 27**):
      i. 2 min at 94°C.
      ii. 45 s at 94°C.
      iii. 45 s at 56°C 31 cycles of **steps ii–iv**.
      iv. 45 s at 72°C.
   b. Simultaneously, perform PCR reactions from the following control templates: Empty library plasmid (diluted to about 0.1 ng); yeast from the positive control colonies (*see* **Subheading 3.3.3.**, **step 7**), treated along with experimental clones as above (*see* **Note 28**); and the same amounts of diluted library plasmid mixed together with the positive control yeast. For analysis of possible outcomes, *see* **Table 3**.
5. Take 10 µL of the PCR product for the *Hae*III digestion (below), and run out the remainder of the PCR reaction (about 20 µL) on a 0.7% agarose gel. Identify

Table 3
Suggested PCR Reactions and Interpretation of Results

Template	Possible outcomes				
Empty prey plasmid[a]	–	+	+	+	+
Yeast containing bait-prey controls[b]	–	–	–	+/–	+
Empty plasmid mixed with yeast bait-prey controls	–	+	–	+/–	+
Clones 1…to *n*	–	–	–	+/–	(+)
Interpretation	Bad mastermix/ bad PCR settings/faulty amplifier	Not enough template from yeast	Lysed yeast inhibited PCR	Too much yeast: uneven template load	As good as it gets; a small fraction of clones always fail
Recommendation	Check or remake solutions/repeat carefully	Add more template/ improve lysis	Add less template	Adjust template load/re-PCR PCR from purified (weak) bands	Analyze other clones until the target number is reached

[a]PCR from the empty library vector yields a product of approx 130 bp for JG4-5 (using the FP1 and FP2 primers); approx 185 bp for YesTrp2 (using the recommended YesTrp2 forward and reverse primers).

[b]From **Subheading 3.3.3.**, **step 7** diploids. Approximate fragment sizes using the recommended primers are for Krit1, approx 1.6 kb; for Raf1, approx 2 kb; and for RalGDS, 0.4 kb.

fragments that appear to be of the same size; *Hae*III digests of these fragments should be run side-by-side. Put gel in a refrigerator until it is ready to isolate fragments (*see* **Note 29**).

6. Perform a restriction digest of 10 µL of the PCR product with *Hae*III in a total volume of 20 µL. Rearrange the loading order according to the results obtained with nondigested PCR, and load the digestion products on a 1.5% agarose gel. Run out the DNA a sufficient distance to get good resolution of DNA products in the 200–1000 bp size range. This will generally yield distinctive and unambiguous groups of inserts, confirming whether multiple isolates of a small number of cDNAs have been obtained.
7. Purify the uncut fragments from the gel by using standard agarose gel-purification techniques. In cases whereby a very large number of isolates representing a small number of cDNA classes have been obtained, the investigator might choose to directly sequence the PCR product (*see* **Note 30**). The purified cDNA can be used directly to reassess the interaction with bait (second confirmation of interaction).
8. The next step is to establish whether isolated cDNAs are reproducible and able to reassess, whether interactors associate specifically with the bait(s) of interest, or are nonspecific ("sticky" proteins), or false-positives that were spuriously isolated owing to mutations in the initial bait strain that lead to nonspecific growth and/or nonspecific transcriptional activation. This can be done using a PCR-recombination approach (derived from **ref. *25***) in a single step, after which confirmed specific positive clones can be worked up through conventional plasmid purification.
9. Perform a restriction digestion of an empty library plasmid with two enzymes producing incompatible ends in the polylinker region (e.g., *Eco*RI and *Xho*I), *see* **Note 31**.
10. Perform PCR from positive control plasmid(s) using the same primers as before, and purify the PCR product.
11. Transform PRT50 containing pGLS22-Ras + pLacGus + pEG202-Krev1 (*see* **Note 19**) with:
    a. Digested library plasmid (50–100 ng).
    b. Digested library plasmid (50–100 ng) and control PCR product (0.5–1 µg).
    c. Uncut library plasmid (50–100 ng).
    Save the extra digested library plasmid and the pPrey-control PCR product for further use in the specificity test **(step 13)**.
12. Plate the transformations on Glu/CM Ura$^-$ His$^-$ Trp$^-$ dropout plates and grow at 30°C for 2 d (until colonies grow). Count colonies (*see* **Notes 32** and **33**). If transformation efficiency in **b** is better than in **a** by 5–20-fold, it is safe to proceed to the next steps. **c** is a positive control for the transformation **(Note 34).**
13. Using same ratios as in **step 11b** above, transform digested library vector in combination with selected PCR products (again, include positive control(s) from **step 9**) to the following:
    a. pGLS23-Bait1 + pLacGus + pMW103-Bait2 (the original naïve bait strain).
    b. pGLS22-Ras + pLacGus + pEG202-Krev1 (the control bait strain).
14. Plate each transformation mix on Glu/CM Ura$^-$ His$^-$ Trp$^-$ dropout plates and incubate at 30°C until colonies grow (2–3 d).

15. Prepare a master plate for each library plasmid being tested. Each plate should contain at least 10 colonies of the transformed PCR-insert/digested plasmid into each of **a** and **b**.
16. Test for coloration and for auxotrophic requirements exactly as described for **Subheading 3.3.5.** above. True-positives should show an interaction phenotype with **a**, but not with **b**. Yeast transformed with control PCR product will provide both positive and negative controls: **a** should be negative whereas **b** should be positive when assayed for both color and growth on the corresponding plates (*see* **Subheading 3.2.**, **step 4**).
17. Proceed with sequencing and biological characterization. Most often, PCR provides ample source of DNA for all subsequent cloning. If needed, transform selected positives into *E. coli* by electroporation ***(20)*** using 1–2 μL of the β-glucuronidase-treated frozen yeast **(step 3)**, and isolate plasmid DNA from Ap^R colonies.

3.3.6.2. Plasmid Isolation Approach: Isolation of Plasmids, Transfer to Bacteria

This protocol provides an alternative option to the basic protocol in case PCR technology is not readily available for use, or in case of failure to obtain a specific PCR product using the library vector primers. This protocol is based on lysing the cells with glass beads after the β-glucuronidase treatment, followed by plasmid transformation in *E. coli* and plasmid isolation, and plasmid retransformation into yeast. A number of kits for yeast minipreps are commercially available, for example, from Clontech (Clontech Laboratories Inc., www.clontech.com, Mountain View, CA) and others.

1. Take 1–2 μL from the β-glucuronidase-treated yeast suspension (*see* **Subheading 3.3.6.1.**, **step 3**) and transform the DNA by electroporation ***(20)*** into any standard *E. coli* strain (e.g., DH5α) selecting a medium containing ampicillin, because only bacteria that have taken up a library plasmid will grow.
2. Select at least two bacterial clones for each yeast clone, and prepare a small quantity of plasmid DNA ***(12)*** from each bacterial clone.
3. Follow **Subheading 3.3.6.1.** from **step 11** to the end essentially as described, except transform with purified library plasmids, instead of mixture of PCR product and digested library vector.

3.4. Follow-up for Library Screening

Following completion of the aforementioned specificity tests, the next step is to leave work with the two-hybrid system, and proceed to biological characterization of the interaction in the appropriate organism for the bait. Such characterization will be necessarily bait specific, and should serve to further eliminate interactions of dubious physiological relevance. Of note, a database of common false-positives, along with discussion of issues related to false-positives, is found at: http://www.fccc.edu/research/labs/golemis/interactiontrapinwork.html.

4. Notes

1. Some of the control plasmids described in this protocol have inserts cloned in a pGLS22 background: pGLS22 is identical to pGLS23 but lacks an *Eco*RI site in the polylinker. The pGLS series of plasmids uses the *HIS5* gene in the histidine synthesis pathway: the PRT50 yeast strains used for selection are doubly mutated in *HIS5* and a second gene in this pathway, *HIS3*, which is used to select the second bait (producing the LexA-fusion). It is thus important to simultaneously introduce both baits into the PRT50 yeast (otherwise no yeast will grow on His^- dropout plates): practically, this saves about a week by skipping transformation step used to combine two different baits.
2. Standard molecular biology techniques (restriction digests) or alternative cloning strategies (i.e., in vivo recombination of PCR products **ref. *26*** or GATEWAY cloning **ref. *27***) can be used. Whatever approach is used, it is a good idea to include a translational stop sequence at the carboxy-terminal end of the bait sequence. It is also important to keep in mind that the assay depends on the ability of the bait to enter the nucleus, and requires the bait to be a transcriptional nonactivator. Hence, obvious membrane localization motifs, or known transcriptional ADs should be removed in the cloning process. Using two-hybrid systems to find associating partners for proteins that are normally extracellular, even though such strategies have apparently worked in a limited number of cases, should be regarded as extremely high risk.
3. A number of modified versions of the plasmid exists, which contains additional restriction sites, altered reading frames, and alternative antibiotic resistance markers (*see* http://www.fccc.edu/research/labs/golemis/interactiontrapinwork.html for details).
4. It is important to use a fresh (thawed from −70°C and streaked to single colony less than ~7 d previously) colony and maintain sterile conditions throughout all subsequent procedures.
5. An efficient transformation yields approx 10^3 transformants/µg of DNA (when three plasmids are being simultaneously transformed). Therefore, this experiment also provides a good chance to assess transformation efficiency, which will be of much higher importance by the time of library transformation. If only a very small number of colonies are obtained, or colonies are not apparent within 3–4 d, this implies that transformation is for some reason very inefficient, and that results obtained in characterization experiments may not be typical. In this case all solutions/media/conditions must be double-checked or prepared fresh, and transformation be repeated. In library transformations, sssDNA is often used as carrier DNA to boost transformation frequency. sssDNA must be of very high quality, whether obtained from a commercial vendor or homemade ***(22)*** (also *see* http://www.umanitoba.ca/faculties/medicine/biochem/gietz/Solutions.html). As a separate issue, if very few transformants containing the bait plasmid appear (compared with the controls), or yeast expressing the bait protein grow noticeably more poorly than control yeast, or if colony population appears much more heterogeneous than control (e.g., presents a mix of large and small colonies), this would suggest that the bait protein is somewhat toxic to the yeast.

6. Assay of multiple colonies is important, because for some baits, protein expression level is heterogeneous between independent colonies, with accompanying heterogeneity of apparent ability to activate transcription of the two reporters. For further discussion, *see* Chapter 16 **Fig. 4**.
7. A replicator for the transfer of multiple colonies can be purchased or easily homemade; it is important that all of the spokes have a flat surface, and that spoke ends are level. A metal frogger can be sterilized by autoclaving or by alcohol/flaming; a plastic replicator must be cut in half to fit to a standard 90-mm Petri plate; it can be sterilized by autoclaving or rinsing with alcohol. The replicator should have 48 spokes in a 6×8 configuration. When making prints on a plate, dip the replicator in the wells of the microtiter plate, then put it on the surface of the solidified medium. Tilt slightly in circular movement, then lift replicator and put it in the microtiter plate (keep the correct orientation). Make sure all the drops left on the surface are of approximately the same size. If only one or two drops are missing, it is easy to correct this by dropping approx 3 µL of yeast suspension on the missing spots from the corresponding wells. If many drops are missing, make sure that all the spokes of the replicator are in good contact with liquid in the microtiter plate (it may be necessary to cut off the side protrusions on the edge spokes of the plastic replicator) and redo the whole plate. Continue replicating by shuttling back and forth between microtiter and media plates. Let the liquid absorb to the agar before putting the plates upside down in the incubator. For alternative techniques to assess LacZ reporters, including growing yeast directly on Xgal- or Xgluc-containing plates, *see* http://www.fccc.edu/research/labs/golemis/interactiontrapinwork.html. The technique described herein is much more sensitive than a standard XGluc plate assay, and can be done within 24 h of plating on appropriate medium, and is generally preferred in high throughput analysis.
8. Transcriptional activation phenotype of bait 1 on the auxotrophic reporters is the most important consideration for library screening, because this is used for direct selection for interaction phenotype. Therefore, if no activation is detected on Lys^- plates, one should proceed further; if bait causes considerable growth on Lys^- plates, it must be modified (e.g., by truncation). There is normally a good correlation between activation of the two reporters, so it is unlikely that a bait, which does not significantly activate *LYS2* will significantly activate *GusA*. For the screening strategy described herein, the behavior of the cI bait is most important; as the LexA bait is primarily being used as a counterselection, weak activation is tolerated.
9. In an optimal result, all six colonies assayed for a given transformation would have essentially the same phenotype. For a small number of baits, this is not the case. The most typical deviation is that of six colonies assayed for a new bait, some fraction appears to be inactive (white in colorimetric assay, and not growing on auxotrophic selection medium), whereas the remaining fraction display some degree of blueness and growth. Do *not* automatically select the white, nongrowing colonies as starting point in a library screen; generally, these colonies possess the phenotypes they do because they are synthesizing little or no bait protein (as can

be assayed by Western blot, *see* **Subheading 3.2.1.**). The reasons for this are not clear; however, it appears to be a bait-specific phenomenon, and may be linked to some degree of toxicity associated with continued expression of particular proteins in yeast. It is usually necessary to modify such bait(s), as use of the unmodified baits is associated with high backgrounds of false-positives and artifactual results. In case of problematic (toxic, autoactivating nonnuclear) baits, a number of bait modification strategies have been described for LexA-based (but not for cI-based) fusions ***(23)***, which may provide useful models for subsequent steps.

10. In addition to the simplified technique described in this chapter, a number of more elaborate (and time-consuming) protocols exist (e.g., *see* Clontech's Yeast Protocols Handbook, available at http://www.clontech.com.
11. Many fusion proteins exhibit sharp decreases in detectable levels of protein with the onset of stationary phase. Therefore, use of the saturated cultures is not recommended for this assay.
12. Frozen samples will be stable for at least 4–6 mo, and will need to be boiled for 5 min at 100°C before loading on an sodium dodecyl sulfate-polyacrylamide gel.
13. The lysates prepared from yeast cultures containing pGLS22-Ras and pEG202-Krev1 allows comparison of expression levels of new baits with two well-expressed bait proteins that have worked well under library screening conditions. Some proteins may be synthesized at lower levels, or be posttranslationally cleaved by proteases (resulting in anomalously small baits). This can be because of the size of the fused domain derived from the protein of interest (proteins of 60–80 kDa and larger often have problems). In case of proteolytically clipped proteins, screens might inadvertently be performed with DBD fused only to the amino(N)-terminal end of the larger intended bait. It is also possible for the proteins expressed at low levels, and seemingly inactive in transcriptional activation assays to be upregulated to much higher levels under the auxotrophic selection, and suddenly demonstrate a high background of transcriptional activation. Hence, it is often a good idea to remake baits showing these properties as smaller derivatives of the proteins of interest. A high percentage of the colonies not appropriately expressing the bait protein, although containing the bait plasmid, may be indicative that the bait is toxic in the yeast. Finally, the best way to find out if a bait protein is correctly expressed is to coexpress it in yeast with a known interaction partner as a prey (i.e., expressed as an AD-fusion), and scoring for transcriptional activation on appropriate dropout media.
14. A "traditional" but more laborious alternative to the mating, directly transforming the library into yeast containing the bait ***(21)***, is not really practical for dual bait system. Such direct transformation in the bait strain requires media not only selective for library plasmid, but also maintaining selective pressure to keep both baits and reporter.
15. A good library transformation efficiency should yield approx 10^5 transformants/µg of library DNA (for transformation with a single plasmid). Transformation of yeast in multiple small aliquots in parallel helps reduce the likelihood of contamination; furthermore, it frequently results in significantly better transformation efficiency than that obtained by using larger volumes of yeast in a smaller number

of tubes. Finally, do not use excess transforming library DNA per aliquot of competent yeast, as competent cells may then take up multiple library plasmids, complicating subsequent analysis. Under the conditions described herein, less than 10% of yeast will contain two or more library plasmids.

16. Although it is possible to throw away the beads after spreading, it is acceptable and efficient to keep the glass beads on the lids while incubating the plates; glass beads will be needed to harvest the library transformants (*see* **Subheading 3.3.2.**). Contamination is much less likely to occur on the glassbeads than on the plates themselves.
17. Thoroughly inspect the plates visually before making a slurry to collect transformants. If visible molds or other contaminants are observed on the plates, carefully excise them and a region around them using a sterile razor blade before adding liquid.
18. This technique also minimizes the time the plates are open, and thus reduces contamination from airborne molds and bacteria. It is more important to ensure the same wash-off rate for all plates, than to collect as many yeast as possible (about one-third of the yeast slurry will be left on the plates). Furthermore, a significant amount of the water added will soak into the plates, so although 10 mL is added, 5 mL is commonly recovered. A second wash can greatly improve the homogeneity of the yield. If one wishes to do a second wash after the first wash, add 10 mL of water, shake again, and transfer the slurry to the next unwashed plate; at the end, make sure all yeast are pooled in a common tube. Optionally, the 24×24 cm^2 plates can be reused many times after removing the remaining agar, washing, and alcohol/ultraviolet sterilization. As these plates are quite expensive, this is a useful point of economy.
19. The bait and reporter plasmids should have been transformed into the yeast less than about 7–10 d before mating with pretransformed library. If it is older, repeat the transformation and Western blot.
20. Titering can be also be done later, in parallel with selection (*see* **Subheading 3.3.4**).
21. Compare selection plates seeded with lower and higher densities. The number of colonies should be roughly proportional to the seeding density, and there should be no background growth. If disproportionally more colonies (or a lawn) appear on the more densely seeded plates, this is background resulting from cross-feeding. In this case, a higher number of plates seeded at lower density should be used. Calculate, how many plates at acceptable cell density are necessary for full representation of the desired number of diploids, and if needed, repeat induction and plating from another frozen mating aliquot.
22. If colonies do not arise within the first week after plating, colonies appearing at later time-points are not likely to represent bona fide positives. True interactors tend to come up in a window of time specific for a given bait, with false-positives clustering at a different time-point: hence, pregrouping by date of growth facilitates the decision of which clones to analyze first.
23. The number of candidate colonies to pick and characterize should be based on the number of cDNA-independent false-positives that arise on the same selection plates for the control mating. The higher the frequency of false-positives, the more

colonies should be picked to find rare true-positives. As the frequency of true-positives will be unknown at this step, the goal will be to pick through all of the false-positives that are expected in the number of library transformants being screened. For example, if the number of library transformants was 10^6, the goal will be to pick through the number of false-positives expected in 10^6 diploids. If the cDNA-independent false-positive frequency is one LYS^+ colony in 10^4 CFU plated, it will be necessary to pick at least 100 LYS^+ colonies to find a true-positive that exists at a frequency of one in 10^6.

24. In general, test plates for auxotrophic reporter characterization lacking only leucine or lysine would automatically keep selective pressure for the presence of the prey and the corresponding bait plasmids. Using plates with fewer dropped-out components would slightly accelerate the growth, and the potential loss of other plasmids would not influence the results of the assay on these plates. However, at the investigator's discretion, Leu^- and Lys^- plates can be substituted for Ura^- His^- Trp^- Leu^- or Ura^- His^- Trp^- Lys^-.
25. In some cases no positives are obtained from library screens. Reasons for this might include inappropriate library source; an inadequate number of screened colonies (<500,000); a bait that in spite of production at high levels is nevertheless incorrectly folded or posttranslationally modified; or alternatively, a bait that does not interact with its partners with a sufficiently high affinity to be detected. Be as well aware of such simple explanations as a wrong batch of plates. In such cases, it may be worth trying screens again with a different variant of bait, screening strain, and/or library, although success is not guaranteed. It is rarely if ever profitable to continue to rescreen the same bait/strain/library combination through >3–5,000,000 primary transformants.
26. Transfer approximately the volume of one middle-sized yeast colony (2–3 µL packed pellet); do *not* take more, or quality of isolated DNA will suffer. The master plate does not need to be absolutely fresh: plates that have been stored for 5 d at 4°C have been successfully used.
27. Modified versions of this protocol with extended elongation times were also found to work; the variant given in this chapter has amplified fragments of as much as 1.8 kb in pretty fair quantity.
28. If the library being screened is based on pJG4-5 plasmid (and primers specific for this plasmid are used in PCR mastermix), only clones containing Raf1 and Krit-1 plasmids would produce products; for a pYesTrp2-based library, take RalGDS clone as positive control.
29. Sometimes a single yeast cell will contain two or more different library plasmids. If this happens, it will be immediately revealed by PCR. In this case two bands can be separately isolated from the gel, and reamplified for subsequent characterization by retransformation against naïve bait. Also, after bacterial transformation an increased number of clones should be checked to avoid the loss of the "real" interactor.
30. *Note*: only the forward primer, FP1, works well in sequencing of PCR fragments; the reverse primer will only work in sequencing from purified plasmid. In general, the TA-rich nature of the ADH terminator sequences downstream of

the polylinker in the pJG4-5 vector makes it difficult to design high-quality primers in this region.

31. Gel analysis produces little information on the completeness of digestion, because it is not possible to distinguish between plasmid species cut by one and two enzymes. Purification of the digested plasmid is not necessary.
32. This control experiment is an indicator of the degree of digestion of the library plasmid. The background level of colonies transformed with digested empty library plasmid **(a)** (*see* **Subheading 3.3.6.1.**, **step 11**) should be minimal. In case the background is high, make sure that the digestion of the empty library plasmid is full and not partial by increasing the digestion incubation time or the restriction enzyme concentration.
33. The fraction of the correct clones can be assessed by replica-plating 12–24 clones to check their phenotype (as in **Subheading 3.3.5.**). Normally, it should be between 85 and 95%.
34. When transformed together, the PCR-amplified cDNA fragment from pPrey-control PCR product and the digested library plasmid will undergo homologous recombination in vivo in up to 97% of the transformants that acquired both vector and insert ***(25,26)***. This is owing to the identity between the cDNA PCR fragment and the plasmid at the priming sites. The background level of colonies transformed with digested empty library plasmid **(a)** should be minimal.

References

1. Fields, S. and Song, O. (1989) A novel genetic system to detect protein-protein interaction. *Nature* **340,** 245, 246.
2. Rual, J. F., Venkatesan, K., Hao, T., et al. (2005) Towards a proteome-scale map of the human protein-protein interaction network. *Nature* **437,** 1173–1178.
3. Stelzl, U., Worm, U., Lalowski, M., et al. (2005) A human protein-protein interaction network: a resource for annotating the proteome. *Cell* **122,** 957–968.
4. Giot, L., Bader, J. S., Brouwer, C., et al. (2003) A protein interaction map of *Drosophila melanogaster*. *Science* **302,** 1727–1736.
5. Stanyon, C. A., Liu, G., Mangiola, B. A., et al. (2004) A Drosophila protein-interaction map centered on cell-cycle regulators. *Genome Biol.* **5,** R96.
6. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA* **98,** 4569–4574.
7. Schwikowski, B., Uetz, P., and Fields, S. (2000) A network of protein-protein interactions in yeast. *Nat. Biotechnol.* **18,** 1257–1261.
8. Gavin, A. C., Bosche, M., Krause, R., et al. (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415,** 141–147.
9. Rigaut, G., Shevchenko, A., Rutz, B., Wilm, M., Mann, M., and Seraphin, B. (1999) A generic protein purification method for protein complex characterization and proteome exploration. *Nat. Biotechnol.* **17,** 1030–1032.

10. Bandyopadhyay, S., Sharan, R., and Ideker, T. (2006) Systematic identification of functional orthologs based on protein network comparison. *Genome Res*. **16(3),** 428–435.
11. Lee, I., Date, S. V., Adai, A. T., and Marcotte, E. M. (2004) A probabilistic functional network of yeast genes. *Science* **306,** 1555–1558.
12. Joung, J. K., Ramm, E. I., and Pabo, C. O. (2000) A bacterial two-hybrid selection system for studying protein-DNA and protein-protein interactions. *Proc. Natl. Acad. Sci. USA* **97,** 7382–7387.
13. Xia, Y., Yu, H., Jansen, R., et al. (2004) Analyzing cellular biochemistry in terms of molecular networks. *Annu. Rev. Biochem.* **73,** 1051–1087.
14. Serebriiskii, I., Khazak, V., and Golemis, E. A. (1999) A two-hybrid dual bait system to discriminate specificity of protein interactions. *J. Biol. Chem.* **274,** 17,080–17,087.
15. Serebriiskii, I. G. and Joung, J. K. (2002) Yeast and bacterial two-hybrid selection systems for studying protein-protein interactions, in *Protein-Protein Interactions: A Molecular Cloning Manual*, (Golemis, E., ed.), Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, pp. 93–142.
16. Serebriiskii, I. G., Mitina, O. V., Chernoff, J., and Golemis, E. A. (2001) Use of a two-hybrid dual bait system to discriminate specificity of protein interactions in small GTPases. *Methods Enzyme* **332,** 277–300.
17. Reeder, M. K., Serebriiskii, I. G., Golemis, E. A., and Chernoff, J. (2001) Analysis of small GTPase signaling pathways using Pak1 mutants that selectively couple to Cdc42. *J. Biol. Chem.* **276,** 40,606–40,613.
18. Serebriiskii, I. G., Fang, R., Latypova, E., et al. (2005) A combined yeast/bacteria two-hybrid system: development and evaluation. *Mol. Cell Proteomics* **4(6),** 819–826, http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15781424.
19. Ausubel, F. M., Brent, R., Kingston, R., et al. (1994-present) Current Protocols in Molecular Biology. 1994-present. John Wiley & Sons, New York, http://www.mrw.interscience.wiley.com/emrw/ 9780471142720/home.
20. Sambrook, J. and Russell, D. (eds.) (2001) *Molecular cloning: a laboratory manual*, *3rd ed.*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
21. Gyuris, J., Golemis, E. A., Chertkov, H., and Brent, R. (1993) Cdi1, a human G1 and S phase protein phosphatase that associates with Cdk2. *Cell* **75,** 791–803.
22. Schiestl, R. H. and Gietz, R. D. (1989) High efficiency transformation of intact yeast cells using single stranded nucleic acids as a carrier. *Curr. Genet.* **16,** 339–346.
23. Serebriiskii, I. G. and Kotova, E. (2004) Analysis of protein-protein interactions utilizing dual bait yeast two-hybrid system. *Methods Mol. Biol.* **261,** 263–296.
24. Finley, R. and Brent, R. (1994) Interaction mating reveals binary and ternary connections between Drosophila cell cycle regulators. *Proc. Natl. Acad. Sci. USA* **91,** 12,980–12,984.

25. Petermann, R., Mossier, B. M., Aryee, D. N., and Kovar, H. (1998) A recombination based method to rapidly assess specificity of two-hybrid clones in yeast. *Nucleic Acids Res.* **26,** 2252, 2253.
26. Stanyon, C. A., Limjindaporn, T., and Finley, R. L., Jr. (2003) Simultaneous cloning of open reading frames into several different expression vectors. *Biotechniques* **35,** 520–522, 524–526.
27. Hartley, J. L., Temple, G. F., and Brasch, M. A. (2000) DNA cloning using in vitro site-specific recombination. *Genome Res.* **10,** 1788–1795.

16

A Bacterial/Yeast Merged Two-Hybrid System

Protocol for Bacterial Screening

Ilya G. Serebriiskii, Nadia Milech, and Erica A. Golemis

Summary

Yeast two-hybrid systems are artificial genetic systems that allow identification and characterization of protein–protein interactions. One common limit to the use of these techniques is when the intrinsic property of "bait" proteins of interest transcriptionally autoactivates reporters, eliminating the basis for interaction detection. To circumvent this problem, autoactivating baits can be alternatively used in bacteria wherein such activation does not occur. A single-vector system has been developed, which can be used either in yeast or in bacteria, streamlining and expanding capacity for protein–protein interaction screens. A concise proposal is provided for use of this system in bacteria; a companion article, chapter 15, describes use of the system in yeast.

Key Words: Protein–protein interaction; transcriptional activation; two-hybrid; yeast; bacteria; library screen.

1. Introduction

The yeast two-hybrid system is a powerful tool for studying protein–protein interactions. This genetic method, based on the reconstitution of a functional transcriptional activator in yeast ***(1)***, has now been used extensively both to identify novel protein–protein interactions and to analyze known interactions. Many extensions to the original two-hybrid system have greatly expanded its utility, enabling use of a two-hybrid paradigm to selectively study protein interactions with RNA, DNA, peptides, and small molecules ***(2)***. Separately, a bacterial genetic selection system analogous to the yeast two-hybrid has been described ***(3–7)***. This bacteria-based system offers two significant advantages over its yeast counterpart: (1) it permits the analysis of very large libraries (>10^8 in size) and (2) it provides an alternative approach to identify interacting partners

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

for eukaryotic proteins that are not amenable to analysis in the yeast-based system (e.g., DNA-binding domain [DBD]-fused proteins that autoactivate transcription, proteins toxic to yeast, or proteins that have undesired interactions with endogenous yeast proteins). In contrast, some advantages are specific to the yeast-based system, including the fact that proteins from eukaryotic organisms are more likely to be properly folded and posttranslationally modified in an eukaryotic milieu.

Together, the yeast and bacterial two-hybrid systems provide powerful methods for analyzing protein–protein interactions. Recently, the creation and optimization of novel vectors was described that could be used to express DBD-fused "baits" that could be used for library screening in either bacterial or yeast environments ***(8)***. A specific advantage of this system is that it reduces bait cloning and characterization work, facilitating screening for interacting proteins in yeast and bacterial systems in parallel, and allowing extremely direct comparison of results obtained in the two systems. As is shown, a single bait used for library screens in yeast and in bacteria could identify very different sets of interacting partners in the two environments (probably because of the considerations discussed earlier) ***(8)***. Hence, use of both systems is more likely to identify a full set of interactive partners for a given bait. This chapter describes the procedures for use of these reagents for screening in bacteria; Chapter 15 describes the use of related reagents (including a common pGLS23 plasmid series) for screening in yeast.

1.1. A Bacterial Two-Hybrid System Based on Transcriptional Activation

The bacterial two-hybrid system described herein is based on the observation that two interacting proteins, "X" and "Y," can trigger transcriptional activation of a weak promoter in *Escherichia coli*. As shown in **Fig. 1**, transcriptional activation is dependent on the expression of two fusion proteins in the cell. One of the fusions (the bait) consists of protein X covalently linked to a DNA-binding protein, which in turn binds to a specific DNA site positioned near the weak promoter. In the system described in this chapter, the DNA-binding protein used for sequence-specific binding comes from bacteriophage λcI repressor. The other fusion (the prey) consists of protein Y (or the proteins encoded by a cDNA library) linked to the α-subunit of the *E. coli* RNA polymerase (RNAP). In this configuration, X is tethered near the weak promoter (through the DNA-binding protein part of the fusion) and, if X interacts with Y, this recruits RNAP to the weak promoter to thereby activate transcription.

In theory, any interacting protein–protein (X–Y) pair should be able to mediate this transcriptional activation, and a number of experiments have demonstrated that this is generally true. Interacting protein–protein pairs from prokaryotes

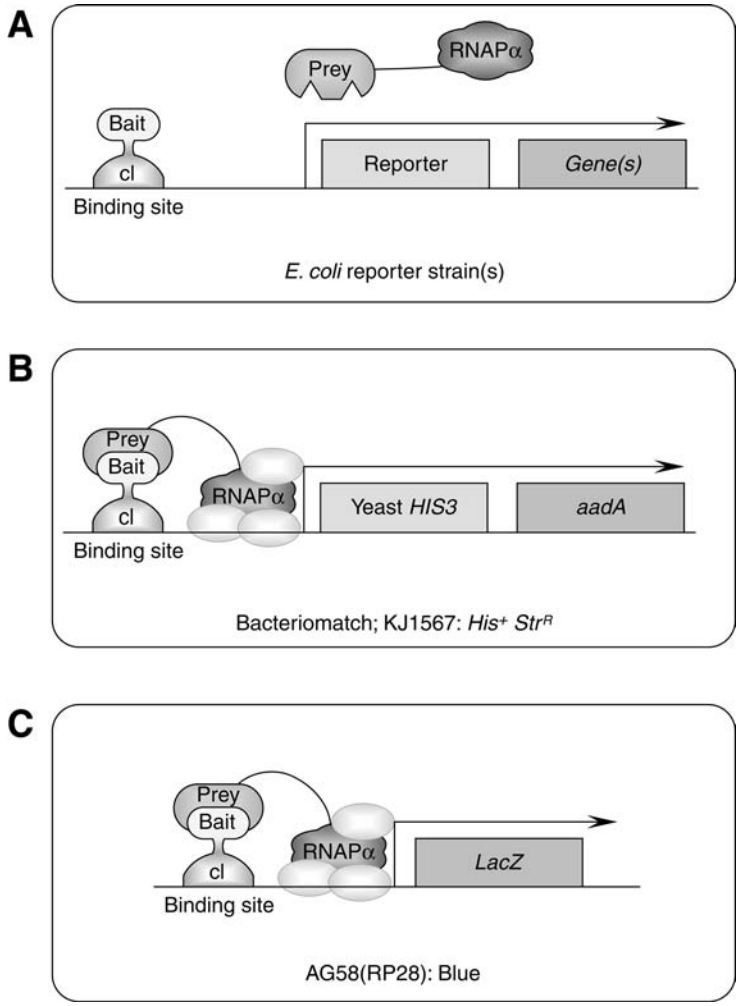


Fig. 1. The bacterial two-hybrid system. **(A)** Schematic showing the components of the bacterial two-hybrid system. A first protein, the bait, is fused to a DBD provided by λcI that in turn binds to a specific DNA-binding site positioned near a weak promoter. A second protein, the prey, is fused to the α-subunit of the *E. coli* RNAP. Interaction between bait and prey mediates the recruitment of RNAP to the weak promoter, thereby activating transcription of a downstream reporter gene(s). **(B)** Expression of the yeast *HIS3* gene following bait–prey interaction can be selected for *E. coli* strains lacking a functional *hisB* gene (KJ1567 or Bacteriomatch). The bacterial *aadA* gene encodes for streptomycin and spectinomycin antibiotic resistance, and provides a second selection. **(C)** The *LacZ* gene in the strain AG58(RP28) encodes β-galactosidase, which can be assayed either qualitatively on a plate-based assay, or quantitatively, as discussed in the text.

(3,9), yeast *(4,5)*, and mammals *(5,8,10)* have all been shown to activate transcription in this system. In addition, the affinity of the interacting partners approximately correlates with the magnitude of transcriptional activation observed *(3,7)*, although it is subject to some transcriptional activation threshold considerations that compress dynamic range *(8)*.

1.2. Selectable Marker Genes Used With the Bacterial Two-Hybrid System

In a "classic" yeast two-hybrid system used for library screening, a yeast strain carries at least one auxotrophic reporter (e.g., *LEU2*, *HIS3*, or *LYS2*) and one colorimetric reporter (e.g., *LacZ* or *GusA*) *(2)*. In contrast, the bacterial system's auxotrophic (*HIS3*) and colorimetric (*LacZ*) reporters are separated in two different strains **(Fig. 1B**,**C)**, whereas the *HIS3* auxotrophic selection is supplemented by a cocistronic antibiotic-resistance selection cassette. For library screening purposes, the *HIS3* selection strain is used as a primary assay system. Positive clones are subsequently assessed by streptomycin/spectinomycin antibiotic resistance using the initial bacteria strain, and by β-galactosidase assay in a second *LacZ* reporter strain.

1.2.1. The HIS3/aadA Selection Cassette

E. coli cells bearing a *hisB* gene deletion do not grow on medium lacking histidine (His-deficient medium). However, expression of the yeast *HIS3* gene in bacteria is sufficient to complement such a *hisB* defect, permitting growth on His-deficient medium *(11,12)*. The stringency of this selection can be raised or lowered by altering the concentration of 3-aminotriazole (3-AT), a competitive inhibitor of the HIS3p enzyme, in the medium *(13)*. In bacteria, the *HIS3* selection marker exhibits a low spontaneous background frequency (~3×10^{-8} breakthrough colonies with 20 m*M* 3-AT *[5]*, and up to ~10^6 candidates can be plated on a single 100-mm agar plate). As shown in **Fig. 1B**, the selection cassette also harbors a secondary selection gene that is expressed cocistronically with the primary *HIS3* selectable marker *(5)*. This secondary reporter, the bacterial *aadA* gene (conferring resistance to the antibiotics streptomycin and spectinomycin *[14]*), can provide a rapid independent means to verify potential positives that come through the initial selection on His-deficient medium (*see* **Subheading 3.**), as a mutation affecting *HIS3* activity will not influence *aadA* expression.

Two *E. coli* strains are currently available for *HIS3/aadA*-based bacterial two-hybrid screening: KJ1567 *(8,15)* and Bacteriomatch II (Stratagene). Both strains grow on the same minimal medium. The most essential difference between them is that the Bacteriomatch II strain is tetracycline (Tc) sensitive, and thus accepts Stratagene libraries constructed in (Tc^R) plasmids such as the

pTRG series, whereas the KJ1567 strain requires a library based on ampicillin-resistant (Ap^R) plasmids. It should be noted that both versions of the system are quite new, reflected by limited publications to date (e.g., **ref. *8***). When choosing which strain to use, one factor to be considered is availability and affordability. The Bacteriomatch II system is predominantly available commercially, whereas a basic set of plasmids described herein is available free of charge from the authors on request. If a library screen is intended, the investigator should check whether or not the appropriate library exists and is affordable, or has to be constructed **(***see* **Note 2)**. Constructing a new library in a pTRG plasmid may be easier using a commercially available kit available from Stratagene, La Jolla, CA. Conversely, pAC-UV5-αLP encodes an *f*1 phage origin, so a library constructed in this vector can easily be converted into a library of infectious transducing phage, and subsequently introduced into selection strain cells by simple phage infection. This may permit the reproducible plating of more than 10^8 library members on a single selection plate ***(15)***.

Besides availability/affordability considerations, both strains should produce comparable results and should be equally well suited to test pair-wise interactions between the targeted proteins. Throughout this protocol, the use of the KJ1567 strain is described; at the appropriate steps, notes indicate the minimal changes in the protocol to be made to instead use Bacteriomatch II reporters.

1.2.2. The LacZ *Reporter*

In a two-hybrid screen, the primary advantages of the *LacZ* reporter are that its expression can be measured quantitatively, and thus one can readily assess the magnitude of activation seen with potential positives. Further, activation of *LacZ* by a bait-interactor combination provides independent verification that activation of *HIS3* is not owing to a nonspecific mutation (e.g., in the *HIS3* promoter region). The *HIS3*/*aadA* double reporter described earlier is more stringent a selection than the single auxotrophic reporter used in yeast. Nevertheless, the AG58(RP28) *LacZ* reporter strain **(Fig. 1C)** is available to provide additional tools for studying the interaction between defined pairs of proteins or to further characterize potential interactors isolated in a library screen using *HIS3/aadA* selection.

1.3. Summary

Using auxotrophic, drug-resistant, and colorimetric bacterial reporter strains as described earlier, one can successfully screen large libraries for candidates that interact with a protein of interest. This selection system has been successfully used to isolate candidates of interest from cDNA, randomized and/or mutagenized libraries. However, direct comparison of the yeast and bacterial two-hybrid

systems has shown that both systems identified physiological interactors for common a bait; nonidentical interactors were also obtained from the screens in the different host organisms ***(8)***. Thus, the bacterial two-hybrid system can be considered to both complement and expand on the yeast two-hybrid system. This chapter provides detailed protocols for using bacterial two-hybrid to analyze protein–protein interactions with the *HIS3*-selection system. It first details methods to construct and characterize a selection strain harboring a "bait" fusion protein. It next describes methods for introducing a library of prey-fusion proteins into the selection strain and protocols for performing the selection. Finally, it suggests additional experiments for validating potential positives from the selection, including characterization of the candidate's specificity by testing their interaction with nonrelated bait, and estimating the interaction strength using *LacZ* assay. An overview of the various stages, as well as estimated time frames for each step, is given in **Fig. 2**.

2. Materials

2.1. Specific Solutions for Media Preparation

Amino acid mixture: 17 different amino acids (no His, Met, or Cys). Make the following six mixtures first; all percentages are (w/v):

1. Phe 0.99%, Lys 1.1%, and Arg 2.5% in water.
2. Gly 0.2%, Val 0.7%, Ala 0.84%, and Trp 0.41% in water.
3. Thr 0.71%, Ser 8.4%, Pro 4.6%, and Asn 0.96% in water.
4. Asp 1.04% and Gln 14.6% in 3% hydrochloric acid.
5. Glu 18.7% and Tyr 0.36% in water with 4 g NaOH.
6. Ile 0.79% and Leu 0.79% in water.

Mix together equal volumes of solutions (1–6) and filter-sterilize through a 0.2-µm nylon filter. Wrap in foil to protect from light and store at 4°C for up to 1 mo. For each 500 mL minimal media, 15 mL of amino acid mixture is required.

2.2. Antibiotics and Supplements

See **Table 1** for preparation and concentrations of antibiotic stocks and supplements. Filter all solutions through 0.2-µm nylon filter and store antibiotic and isopropyl-beta-D-thiogalactopyranoside (IPTG) at –20°C, 3-AT foil wrapped at +4°C.

2.3. Media Preparation

Standard size plates (100- or 90 mm) are used throughout this protocol.

1. Liquid NM medium: to make 500 mL, mix reagents listed next, and filter-sterilize through a 0.2-µm filter:

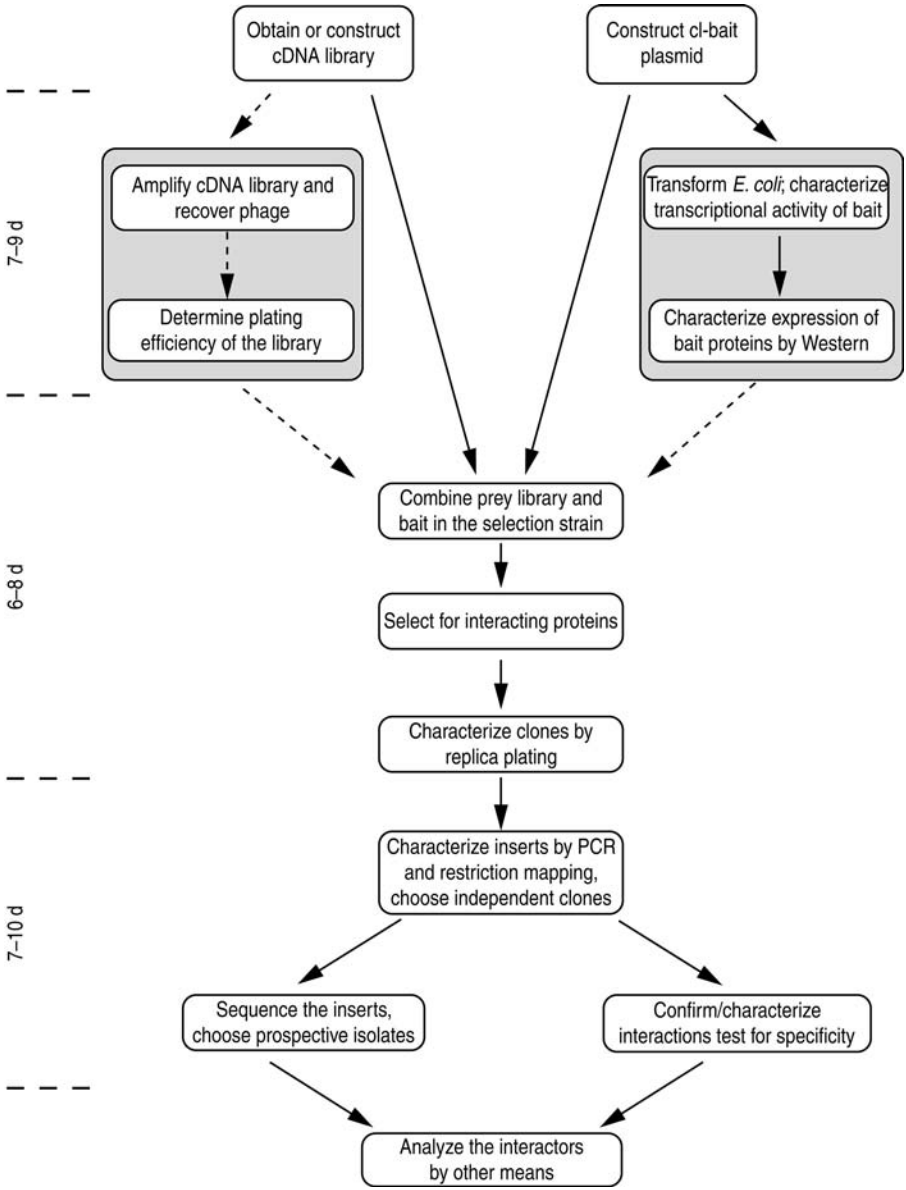


Fig. 2. Flowchart for use of the bacterial two-hybrid system. *See* text for details; dashed lines indicate optional steps.

Table 1
Concentrations of Antibiotics and Other Supplements Used in Agar Plates

Antibiotic	Stock solution	Final concentration in plates	Abbreviation[a]
Ampicillin	100 mg/mL in H_2O	100 µg/mL	A
Chloramphenicol	30 mg/mL in ethanol	15 µg/mL	C
Kanamycin	10 mg/mL in H_2O	10 µg/mL	K
Tetracycline	10 mg/mL in 80% ethanol	5 µg/mL	T
Streptomycin	10 mg/mL in H_2O	10 µg/mL	S
IPTG	1 *M* in H_2O	Up to 50 µ*M*	I
3-Aminotriazole	1 *M* in H_2O	1–40 m*M*	AT

[a]Media description consists of the name of basic medium, followed by abbreviations for antibiotics and inducers, followed by concentration of aminotriazole. For example, LB_AC indicates LB plates with ampicillin and chloramphenicol; NM_ACSI_5AT is NM medium with ampicillin, chloramphenicol, streptomycin, IPTG, and 5 m*M* 3-Aminotriazole.

418 mL	ddH_2O
50 mL	10X M9 salts (Miller recipe ***[16,17]***: for 1 L, add 60 g Na_2HPO_4, 30 g KH_2PO_4, 5 g NaCl, and 10 g NH_4Cl, autoclave)
10 mL	20% Glucose
5 mL	20 m*M* Adenine HCl
15 mL	Amino acid mixture (*see* **Subheading 2.1.**)
0.5 mL	1 *M* $MgSO_4$
0.5 mL	10 mg/mL Thiamine
0.5 mL	10 m*M* $ZnSO_4$
0.5 mL	100 m*M* $CaCl_2$ (always add this last).

2. NM agar plates: for 500 mL of plates, autoclave 418 mL ddH_2O with 7.5 g of bacto-agar and a stir bar, allow agar to cool to 65–70°C and then add the same 82 mL of basic components as for the liquid NM medium (premixed and filtered).
3. Liquid Luria-Bertani (LB) medium: add 5 g bacto-tryptone, 2.5 g yeast extract, and 5 g NaCl to 400 mL dH_2O, adjust pH to 7.5 with NaOH, adjust volume to 500 mL, autoclave 15 min.
4. LB agar plates: add 7.5 bacto-agar to liquid LB media before autoclaving.

Add antibiotics, IPTG, and 3-AT to media as needed (*see* text for details). When adding antibiotics, ensure temperature of autoclaved media is less than 70°C before addition.

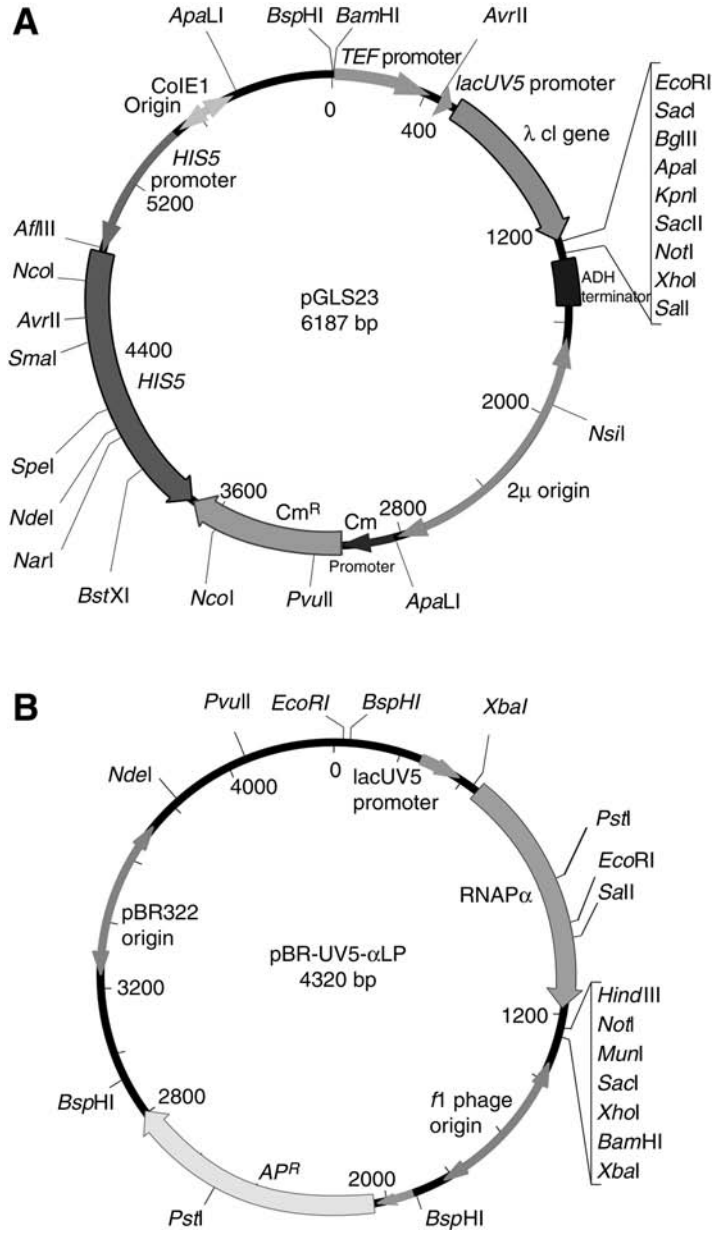


Fig. 3. Vector maps. **(A)** cI fusion vector pGLS23. This vector is referred to as pBaitC in the text. **(B)** RNAP fusion vector pBR-UV5-αLP, used to express the prey in the KJ1567 selection strain. Further maps, sequences, and polylinkers for system-compatible plasmids are given on the websites: http://www.fccc.edu/research/labs/golemis/InteractionTrapInWork.html (source of these maps) and http://www.stratagene.com/lit/vector.aspx.

Table 2
Plasmids

Short name/ source	Full name	Antibiotic marker	Comment/description
pBaitC[a]	pGLS23[a]	Cm^R	Basic plasmid to clone bait as a fusion to λcI protein
pTRG[b]	pTRG	Tc^R	Basic plasmid to clone prey as fusion to *E. coli* RNAP α-subunit residues 1–248
pLibB[a]	pBR-UV5-αLP	Ap^R	Basic plasmid to clone prey as a fusion to *E. coli* RNAP α-subunit residues 1–248; has *f*1 origin of replication
pTRG-RGL (control)	pTRG-RGL-2	Tc^R	Control plasmid that expresses an activation domain-RGL fusion protein
pBaitC-Ras[a] (control)	pGLS22-HRas	Cm^R	Bait plasmid expresses a λcI-HRasfusion; positive control for interaction with BRaf and RGL-2
pBaitC-zip[a] (control)	pGLS22-EE_{12345}	Cm^R	Bait plasmid that expresses a λcI-leucine zipper protein fusion; negative control for interaction
pLibB-Raf[a] (control)	pBR-UV5-αLP-BRaf	Ap^R	Control plasmid that expresses an activation domain-BRaf fusion

Expression of the fusion proteins in *E. coli* is driven by tandem *lpp/lacUV5* promoters.

All plasmids except for pAC-UV5-αLP have a pBR322 *E. coli* replication origin.

It is noted, pGLS23 is described as the prototypical cI-fusion (pBaitC) plasmid: this dual-host expression vector also allows the screening in a Dual Bait yeast two-hybrid system, as described in Tikhmyanova et al. Chapter 15.

[a]Golemis lab (Fox Chase Cancer Center, Philadelphia, PA).

[b]Stratagene.

2.4. Vectors

Figure 3 and **Table 2** summarize and provide maps and other information for the bacterial two-hybrid vectors used in this method (*see* **Note 1**).

2.5. Bacterial Strains

See **Table 3** for genotypes of the *E. coli* strains used for selection.

Table 3
Bacterial Screening Strains

Strain/source	Genotype	Comments/description
Bacteriomatch II Reporter[a]	*Δ(mcrA)183 Δ (mcrCB-hsdSMR-mrr)173, endA1 endA1 hisB supE44 thi-1recA1 gyrA96, relA1 lac* (*F′ laqI^q HIS3 aadA Kan^R*)	
KJ1567[b]	*ΔhisB463*, *Δ(gpt-proAB-arg-lac) XIII zaj::Tn10* (F′ *LacI^q HIS3 aadA Kan^R*)	Reporter strains in which the expression of the *HIS3* and *aadA* reporter genes is directed by a weak promoter bearing an upstream λcI DNA-binding site
AG58A(RP28)[b]	*ΔhisB463*, *Δ (gpt-proAB-arg-lac)XIII zaj::Tn10* (F′ *LacI^q LacZ Kan^R*)	Reporter strain in which the expression of the *lacZ* reporter gene is directed by a weak promoter bearing an upstream λcI DNA-binding site

[a]Stratagene.
[b]Golemis lab.

2.6. Primers for Sequencing and Polymerase Chain reaction (PCR)

Primer name	Sequence	Target	Sequence/ PCR	Direction
FPB1	ATGATCCCATGCAATGAGAG	pBaitC	Sequence	Forward
FPP1	ATCCTGAAGAGGCGATTCG	pLibB, pTRG	Sequence	Forward
FPP2	TGGAAACCAACGGCACAATC	pLibB, pTRG	Sequence, PCR	Forward
RPP1	TCTCGCCTGTGTCTT CTTACTTAGG	pLibB	Sequence, PCR	Reverse
RPP2	GACGCTCAGTGGAACG AAAACTC	pTRG	Sequence, PCR	Reverse

2.7. Miscellaneous

1. Sterile glass balls, 3–4 mm, Thomas Scientific (Swedesboro, NJ) 5663L19 or Thermo Fisher Scientific, (Waltham, MA) no. 11-312A (autoclave in jar to sterilize).
2. Sterile toothpicks (to sterilize, autoclave foil-wrapped toothpicks, or in a jar).
3. Insert grid from a rack of pipet tips (Rainin Instrument [Oakland, CA]) RT series, 200 µL capacity).
4. A metal replicator for the transfer of multiple colonies (e.g., Dankar Scientific [Reading, MA] no. MC48) (*see* **Fig. 5**) (*see* **Note 3**), or alternatively, a plastic replicator (Bel-Blotter, Bel-Art Products (Pequannock, NJ) no. 378776-0002 or Fisher no. 1371213) (*see* **Note 3**).
5. Antibody to cI (commercially available from Invitrogen Corp [Carlsbad, CA]), and other reagents for Western blotting.
6. 2X Laemmli sample buffer: 0.125 *M* Tris-HCl, 4% (w/v) sodium dodecyl sulfate, 20% (v/v) glycerol, 10% (v/v) 2-mercaptoethanol, and 0.002% (w/v) bromophenol blue (pH 6.8). Add 2-mercaptoethanol immediately before use.

3. Methods

3.1. Construction and Characterization of a Bait

The protein to be screened for interactors is fused to the bacteriophage λ repressor (λcI protein) DNA-binding protein. Two tests of this bait fusion protein are then performed. First, an activation assay verifies that the bait fusion does not activate the reporter promoter on its own (*see* **Note 4**). Second, a Western blot assay establishes the expression levels and stability of bait fusion protein inside a bacterial cell (*see* **Note 5**).

For these assays, expression of the bait-λcI fusion is induced by IPTG, and protein levels can be increased or decreased by adjusting the IPTG concentration in the medium. If the expression level of the bait fusion is too low (it cannot be detected by Western analysis, under circumstances in which the positive control cI-fused protein is clearly seen in bacterial lysate) or if the bait is degraded (seen as a ladder or smear of less than the expected molecular weight), then it is not appropriate for use in bacterial two-hybrid screens. In these circumstances, the bait construct needs to be redesigned (e.g., by truncating the bait to smaller domains).

Transcriptional activation of the reporters by the bait (in the absence of any interacting partner) is checked by assaying the expression of a *HIS3* and/or *LacZ* reporter gene directed by a weak promoter bearing an upstream λcI-binding site. Baits should *not* activate expression of the *HIS3* or *LacZ* reporter genes in the absence of the interacting RNAP-prey fusion.

3.1.1. Constructing and Transforming a lcI Bait

1. Insert DNA encoding the protein of interest (your favorite gene, [YFG]) into the polylinker of the λcI-encoding expression vector, pBaitC to create an in-frame cI-fusion protein (*see* **Note 6**). For simplicity, this plasmid will be referred to as pBaitC-YFG in this protocol.

2. Transform the KJ1567 or Bacteriomatch II *E. coli HIS3/aadA* reporter strain (and, optionally, the AG58(RP28) *LacZ* reporter strain) with the following combinations of plasmids (*see* **Notes 7–11**):
   a. pBaitC-YFG + pLibB-Raf (test for autoactivation).
   b. pBaitC-Ras + pLibB-Raf (positive control for activation and interaction).
   c. pBaitC-zip + pLibB-Raf (negative control for activation and interaction).
3. Plate each transformation on LB_AC plates and incubate at 37°C for 12–18 h, until colonies have grown.

3.1.2. Assessing Bait Activation of Reporter Genes: Replica Technique/Gridding

For each combination of plasmids, assay at least six independent colonies for their ability to activate the auxotrophic and colorimetric reporters (*see* **Note 12**). Assessment of transcriptional activation requires the transfer of colonies from master plates to a variety of selective media. This transfer can be accomplished simply, by using a sterile toothpick to move cells from individual patches on the master plate to each of the selective media. However, in cases in which large numbers of colonies and combinations of bait and prey are to be examined it is useful to use a transfer technique that facilitates high-throughput analysis. The following technique, based on microtiter plates, is an example of such an approach.

1. Add approx 100 µL of sterile NM medium to each well of one half (wells A1–H6) of a 96-well microtiter plate. As shown in **Fig. 4A**, place an insert grid from a rack of micropipet tips over the top of the microtiter plate, and attach it with tape: the holes in the insert grid should be placed exactly over the wells of the microtiter plate. Although this is not essential, the grid will stabilize the tips in the plate, and allow simultaneous removal, speeding the replication process.
2. Using sterile plastic micropipet tips, pick six colonies (1–2-mm diameter) from each of the transformation plates **a–c** (**Subheading 3.1.1.**, **step 2**). Put each set of six across one of the first top four rows, and leave the tips supported in a near-vertical position by the insert grid until all the colonies have been picked.
3. Swirl the plate gently to mix the cells into suspension, remove the sealing tape, and lift the insert grid, thereby removing all the tips at once.
4. To print the cell suspensions on a plate, place the replicator (*see* **Note 13**) into the microtiter plate, lift it and turn 180°, and then reinsert into the remaining half of the plate. This will make an approx 1:20 dilution of one's primary suspension in the bottom four rows (reversed mirror image—*do not* forget).
5. Use the replicator to plate bacterial cells suspensions on the following plates *see* **Subheadings 2.1.–2.3.**:
   a. LB_AC (backup master plate, media test).
   b. NM_AC (master plate).
   c. NM_ACI.
   d. NM_AC_5AT.
   e. NM_ACI_5AT.

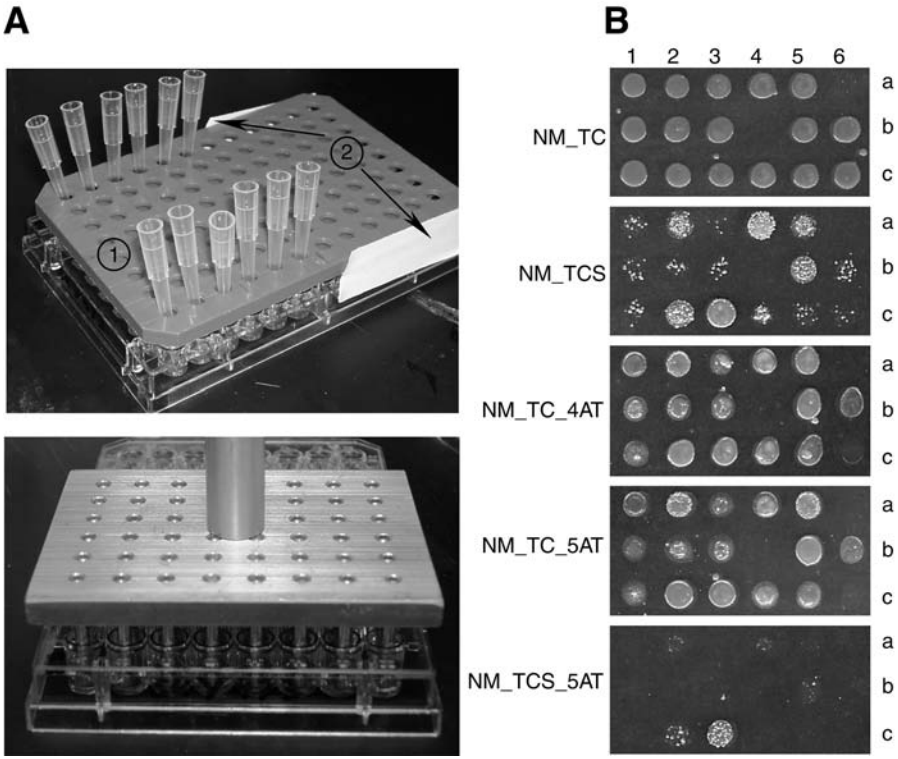


Fig. 4. Characterization of bacterial reporter activity using replica technique. **(A)** Replica techniques. **(Top)** Pick each bacterial colony (1–2-mm diameter) to be tested from the transformation plates, and resuspend it in NM medium 100 μL of sterile in a well of a 96-well microtiter plate. If sterile toothpicks are used for picking colonies, they need to be removed immediately after resuspension of a colony, to prevent absorption of the liquid. Plastic pipet tips can be also used; place an insert (as indicated by **ref.** ***1***) from a rack of pipet tips (e.g., Rainin RT series) over the top of the microtiter plate, and attach to the sides of the plate using tape (shown in **ref.** ***2***). **(Bottom)** After removal of the tips, a metal replicator is used as described in the text to transfer colonies to selection plates. **(B)** Typical results on selection plates. Herein potential H-Ras interactors from a Bacteriomatch HeLa library are analyzed. Patches were obtained after printing of the bacterial suspension on various selection plates (*see* **Note 13** for using replica tool). The NM_TC medium is nonselective for interaction, whereas use of various concentrations of 3-AT, addition of streptomycin, or the combination of these approaches represents various selection strengths. These are used to identify candidate interactors, as described in **Subheading 3.2.2.**

f. NM_ACS.
g. NM_ACIS.

Incubate the plates at 37°C for up to 12–24 h, then save the NM_AC master plate at 4°C while assays are run.

6. *Analysis:* ideally, all six colonies assayed representing the same transformation would have essentially the same phenotype, although some heterogeneity is normal (*see* **Fig. 4B**). Bacteria containing the strong positive control (from transformation **b**) should be detectably growing on *HIS3* (NM_AC_5AT) and *aadA* (NM_ACS) selection plates both with and without IPTG (i.e., even minimal quantities of proteins produced should activate these reporters). The negative control (from plasmid combination **c**) should not grow on either *HIS3* or *aadA* selection plates. If the bacteria containing the bait under test (from plasmid combination **a**) shows no growth on the *HIS3* and *aadA* selection plates within 24 h, it is probably suitable for library screening; if it is similar to the positive control, it must be reconfigured. Intermediate growth phenotypes (e.g., weak growth only in plates containing IPTG) suggest the bait may be usable, but may have background in a library screen. **Figure 4B** shows typical results from a characterization of bacterial reporter activity using the replica technique.

   Several "technical" conclusions can also be drawn on analysis of the results of this experiment. First, it allows optimization of the nonselective minimal medium, which could be especially important for the subsequent library screening. However, LB_AC is a much richer medium than NM_AC; in an optimal result, the difference in bacterial growth on LB_AC and NM_AC plates should not be dramatic by 12–24 h after plating. If bacterial growth rate and/or plating efficiency on NM is low, it should be optimized (check the components, adjust antibiotic concentrations and drying time for plates). In the meantime, use the LB_AC plate as a "healthy" master plate. Second, slow growth in the presence of the IPTG-inducer on a nonselective plate (NM_ACI) would suggest toxicity of the bait; this may lead to artifacts and many false-positives during a library screen. Third, analysis of the growth pattern of the positive and negative controls allows assessment of the selective medium. If there is no good discrimination between the growth of positive and negative controls, or if the growth of the positive control cells is very poor at 24 h, the concentrations of streptomycin and 3-AT should be adjusted.
7. *Optional:* if the AG58(RP28) *LacZ* reporter strain is also used, assay β-galactosidase activity of the emerging clones by using one of the quantitative assays (e.g., ***[8,18]***).

3.1.3. Detection of Bait Protein Expression (Western Blot)

Western analysis of lysates of bacteria containing DBD-fused baits is important for the characterization of the size and expression level of the bait size. Some proteins may either be synthesized at very low levels, or be posttranslationally clipped. Either of the above two problems can lead to complications in library screens. Where proteins are proteolytically clipped, screens might inadvertently be performed with DBD fused only to the amino-terminal end of the larger intended bait. Western analysis should be performed as follows:

1. From the NM_AC master plate, inoculate at least two colonies per test bait into 2 mL of NM_AC liquid medium (if using an LB_AC master plate, adjust to LB_AC). In parallel, set up cultures of pBaitC-Ras and/or pBaitC-zip transformants as positive controls for protein expression (*see* **Fig. 5**, lanes 1 and 5). Grow overnight cultures on an orbital shaker at 37°C. For optimal protein expression, dilute the saturated cultures 1/100 into fresh tubes containing 2 mL of the same medium with or without IPTG to obtain exponentially growing cultures (*see* **Note 14**).
2. After the OD_{600} nm of the cultures reaches 0.35–0.5 (after about 4–6 h), harvest cells from 1.5 mL of each culture by centrifuging at 13,000*g* for 3–5 min in a micro-centrifuge. Carefully remove supernatant from the cell pellet.
3. Add 50 µL of 2X Laemmli sample buffer to each tube, and rapidly vortex to resuspend each pellet. Heat the samples at 100°C for 5 min for immediate assay, or freeze (using dry ice for flash freezing) and store at –70°C for subsequent use: frozen samples should then be heated at 100°C before proceeding to **step 4**.
4. After heating, chill the samples on ice, then centrifuge for 30 s at 13,000*g* to pellet large cellular debris. Dilute 1:100 in 1X Laemmli sample buffer, and load 10–25 µL of each sample onto a 10% (w/v) sodium dodecyl sulfate polyacrylamide gel electrophoresis gel.
5. Prepare for Western blot analysis using standard transfer approaches ***(17)***, and probe membrane using an antibody to cI. This allows comparison of expression levels of the bait protein under test with control cI-fused proteins (*see* **Fig. 5** for a typical example of an immunoblot detecting cI-bait expression).

3.1.4. Troubleshooting Baits With Undesirable Characteristics

1. If a bait is expressed at inappropriately low/high levels (more than 10-fold divergent from the controls), one may wish to consider adjusting the concentration of IPTG used in the test plates.
2. If a bait autoactivates the reporter to any significant degree, it is probably worthwhile to subdivide the bait into overlapping domains, creating new baits that may have reduced autoactivation potential. In doing so, try to divide the bait according to any available information about protein structure, in order to avoid disrupting discrete domains.

3.2. Introducing the Library Into the Selection Strain and Selecting Interactors

Methods for introducing prey libraries into the selection strain cells differ depending on the library. In the most straightforward approach, the prey-plasmid library is electroporated directly into high-transformation efficiency cells along with the pBait plasmid. Alternatively, a library constructed using pLibB can be converted into a library of infectious transducing phage, and subsequently introduced into selection strain cells containing only the bait (prepared in **Subheading 3.1.1.**, *see* **Note 11**) by simple infection with the phage.

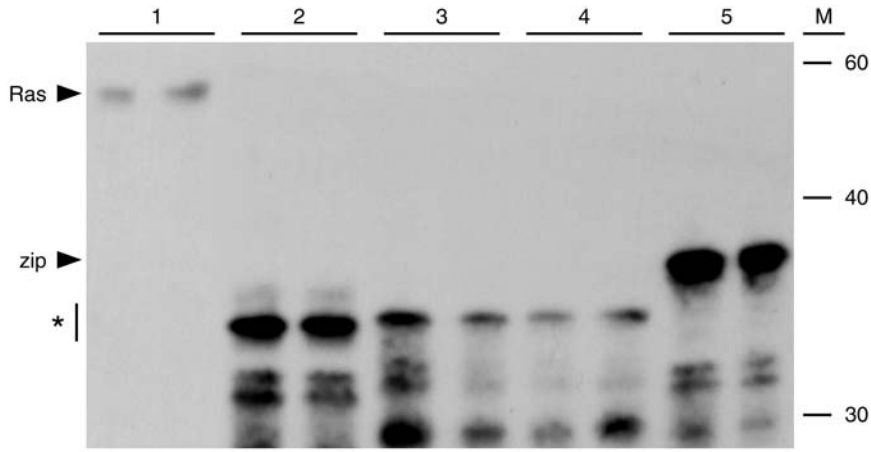


Fig. 5. Detection of bait expression by Western blot. Transformation (**lane 1**), cI–HRas fusion (pBaitC-Ras); Transformations (**lanes 2–4**), various leucine zipper fusions being evaluated for expression level (marked with ∗); transformation (**lane 5**), control pBaitC-zip; M, marker lane with the approximate size in kilodaltons. Two independent transformants are assayed for each transformation. Equivalent quantities of total lysate for each bacterial culture are loaded in each lane on the gel, and the subsequent membrane immunoblotted with α-cI primary antibody. Note that different baits are expressed at different levels.

Following plating of the transformation on selective plates, potential candidates appear as colonies over the following 24–48 h. These colonies are subjected to an initial confirmatory test (*see* **Subheading 3.2.2.**) that looks not only for increased expression of the primary selectable marker *HIS3,* but also of the *aadA* secondary reporter (*see* **Note 15**). Prey candidates that pass this initial test are carried through to additional testing in **Subheading 3.3.**

3.2.1. Introducing the Library into Selection Strains, and Performing the Selection

1. Prepare highly competent KJ1567 cells by a standard high-efficiency protocol. Alternatively, use manufacturer's instructions to render the Bacteriomatch II strain highly competent. Aim for a transformation frequency of more than 10^8. Transform 50–100 µL aliquot of bacteria with a mixture of library and bait plasmid (50 ng each). At the same time, perform a parallel transformation with pBaitC-YFG and control library plasmid: carrying yeast transformed with these two plasmids throughout the subsequent steps provides a negative control for the library screen.
2. After the transformation, bacteria typically recover 90–120 min in 1 mL of rich medium broth. After recovery, transfer the culture to sterile tubes and spin in a microcentrifuge at 2000*g* for 10 min (room temperature). Remove the medium and

resuspend the cell pellet in 1.5 mL of fresh NM_ACI medium. Grow the cells for 2 h at 37°C in this media, to allow the cells to adapt to growth in minimal medium before plating, providing selection for bait (C) and library (A) plasmids.

3. Serially dilute approx 100 µL of the transformed cells at least to 10^{-5} in NM medium. Plate 100 µL of each of the 10^{-3}–10^{-5} dilutions on:
   a. LB_C plates (to assess the number of bait-transformed cells).
   b. LB_A plates (to assess the number of library-transformed cells).
   c. LB_AC and NM_ACI plates (to assess the number of doubly transformed cells capable of growing on rich and minimal medium, respectively).
4. In parallel, plate the remainder of the transformed cells, dividing evenly among 10 NM_ACI_5AT plates. Add 10–12 sterile glass beads per plate, Thomas Scientific (Swedesboro, NJ) or Fisher no. 11-312A, and gently agitate (do not shake) plates to distribute the cells evenly on the plate. Allow to air-dry, and then invert and incubate for 24–36 h at 37°C, followed by up to 24 h at room temperature.
5. Count the colonies that grow within 24 h on the titer plates from **step 3** to verify the total number of cells transformed with the prey plasmids that were plated on the selection plate. Calculate the efficiency of double transformation, and the total number of double-transformed cells plated on selective medium. If the library is not fully represented, it may be necessary to repeat the transformation (possibly adjusting the amount of plasmid DNA) to achieve the full coverage of the library. However, plating more than 10^6 of transformants per plate is not recommended, as this might result in increased spurious background.
6. Inspect the NM_ACI_5AT selection plates for the appearance of colonies.
   a. If the predicted number of viable transformants has been plated, but no colonies appear on the NM_ACI_5AT selection plates, one can repeat the transformation procedure, but perform the selection at lesser stringency using lower concentrations of 3-AT. However, note that the number of background colonies increases with lower concentrations of 3-AT.
   b. If too many candidates appear on the NM_ACI_5AT plates, giving the impression of nonselected bacterial growth, one can redo the selection at greater stringencies using higher concentrations of 3-AT (as high as 40 m*M*). However, caution should be used, as concentrations of 3-AT more than 30 m*M* may decrease the plating efficiency of positive candidates (*see* **Note 16**).
7. Colonies that grow on the selective plates should be rapidly (within 24 h) processed to the next step (*see* **Subheading 3.2.2.**).

3.2.2. Confirmation of Positive Interactions: Test for Secondary Reporter Activity

One rapid method to characterize potential positive colonies is using replica technique (as described in **Subheading 3.1.2.**) to assess the level of expression of the secondary *aadA* reporter present in the selection strains. True-positives, as opposed to nonspecific mutations affecting histidine auxotrophy, should also show increased *aadA* activity, reflected in growth on media containing streptomycin. In addition, plating potential positive clones on a series of plates made

with a range of 3-AT concentration allows some ranking of the candidate interactors.

1. Pick potential positive colonies directly from the selection plate and resuspend in 100 µL of NM medium in a microtiter plate well, using the four upper rows. Also, include positive and negative controls from **Subheading 3.1.2.**
2. Dilute colony suspensions approx 1:30 in approx 100 µL NM, using the four bottom rows of the microtiter plate, as described in **Subheading 3.1.2.**, **step 4**.
3. Using the replica tool, print suspensions and dilutions on the following plates:
   a. NM_ACI (to confirm effective colony transfer; one master plate).
   b. NM_ACI_4AT.
   c. NM_ACI_5AT.
   d. NM_ACI_10AT.
   e. NM_ACI_20AT.
   f. NM_ACIS.
   g. NM_ACIS_5AT.
4. Incubate plates for 24 h at 37°C, and inspect for growth. If necessary, allow plates to incubate an additional 12–48 h at room temperature, inspecting periodically. Compare the growth of the candidate colonies vs positive and negative controls on the selective plates and rank them accordingly (taking photographs or scans is helpful). If none of the candidates grow on the streptomycin plates (and particularly if growth of the positive controls is slow on these plates), replate the whole set on lower concentrations of streptomycin.
5. Colonies that rank best by growth on the 3-AT and/or streptomycin plates should be designated as preferred first round positives and carried forward to **Subheading 3.3.** for further analysis. Keep the NM_ACI master plate in the refrigerator. Glycerol stocks of strongest positives should also be made at this stage (*see* **Note 17**).

3.3. Second Confirmation of Potential Positive Candidates

In this stage, candidates initially confirmed as positives in **Subheading 3.2.** are further tested to determine if the increased reporter-gene expression is linked to the expression of the specific prey isolated from the library. To perform this analysis, the plasmid encoding the prey fusion is isolated and, along with the bait, is reintroduced into naive selection strain cells. If the ability to grow on selection plates is linked to the prey plasmid, as indicated by recapitulation of the interaction phenotype, then the insert may be sequenced, or used for other tests.

3.3.1. Isolation of Purified Prey Plasmid From Bacteria With Candidate Interactors

A major strength of the optional **steps 1** and **2** in this protocol is that it will identify redundant clones before two rounds of plasmid isolation and bacterial transformation, which in some cases greatly reduces the amount of work required.

1. Use primers specific for the library plasmid (*see* **Subheading 2.6.**). Perform a PCR amplification (in ~50 µL volume with small amounts—barely visible—of the bacterial colonies as PCR template), designing the amplification program as follows:
   a. 1 min—94°C.
   b. 45 s—94°C.
   c. 45 s—56°C (31 cycles of *b–d*).
   d. 45 s—72°C.

   In parallel with candidate colonies, set up PCR reactions from the following control templates: appropriate control plasmid (either pLibB-Raf or pTRG-RGL highly diluted); *E. coli* carrying the same control plasmid (*see* **Subheading 3.1.2.** and **Note 18**).
2. Run out an aliquot of the PCR reaction on a 0.7% agarose gel. Identify fragments that appear to be of the same size. Digest some of the PCR-amplified DNA for these clones with the frequently cutting restriction enzyme *Hae*III, to determine if an equivalent digestion pattern results: if so, the colonies are likely to be identical (*see* **Note 19**). Purified PCR fragments can be sequenced using the same primers used for amplification.
3. For each potential-independent candidate, inoculate 2 mL of LB_A with bacteria patched from a spot on the master plate created in **Subheading 3.2.2.**, **step 3**. Grow at 37°C with agitation for 12 h or overnight.
4. Isolate total plasmid DNA for each independent candidate interactor from 1.5 mL of the overnight culture using any standard miniprep isolation procedure. Resuspend or elute DNA in a final volume of 40 mL water. The plasmid DNA isolated by this method will include not only the prey plasmid, but also the bait plasmid as well. Hence, another round of transformation is necessary to separate the prey from the bait (*see* **Note 20**).
5. Use 1 µL of the DNA from **step 4** to transform a naive reporter strain (or any standard *E. coli* cloning strain, such as DH-5α). To specifically rescue the library plasmid, spread 1/20th of the final transformation volume (or streak on a sector of a plate using inoculation loop) on a LB_A plate in order to get single colonies. Incubate 16–18 h at 37°C.
6. Pick two colonies from each transformation and using sterile toothpicks or tips patch in an identical grid to an LB_C plate and an LB_A plate. Let patches grow 6–8 h at 37°C.
7. Transformants harboring only the prey plasmid should not grow on the chloramphenicol (LB_C) plate. For each candidate, pick one of the colonies that meet this criteria from the LB_A patch plate made in **step 6** and inoculate a 10 mL culture of LB supplemented with100 µg/mL ampicillin and grow at 37°C for 16–18 h with agitation.
8. Isolate plasmid DNA from the 10 mL cultures using standard commercially available alkaline lysis/column purification methods (or the means of one's choice ***[17]***). Utilize procedures for low-copy number plasmids, and perform all extra wash steps to obtain plasmid DNA of optimal yield and quality. This DNA can be sequenced, or further confirmation tests (*see* **Subheading 3.3.2.**) performed.

3.3.2. Additional Confirmations of Positive Interactions, and Specificity Test for Positive Candidates

1. A strongly recommended option is to use the AG58(RP28) strain to further characterize potential interactors isolated in the library screen based on the ability of interacting proteins to activate *LacZ* activity. For this purpose, transform the desired bait/prey combinations, including controls, in the AG58(RP28) strain essentially as described in **Subheading 3.2.1.**, and then perform quantitative β-galactosidase assays ***(18,19)***.
2. One additional test that may be performed before sequencing the prey-encoding plasmids is to check that the candidate preys interact specifically with the bait used for their selection in the two-hybrid assay. This is accomplished by testing whether or not the prey can activate the weak promoter reporter in the absence of the original bait fusion protein.
   a. Use 1 µL of purified prey plasmid DNA along with the bait plasmid to cotransform reporter strain cells. In parallel, cotransform the candidate prey plasmid with the pBaitC-zip and/or pBaitC-Ras control baits (*see* **Notes 21** and **22**). Plate transform cells on NM_AC as described in **Subheading 3.3.1.**, **step 5** and characterize six independent colonies for each transformation as described in **Subheading 3.1.1.**
   b. Analyze growth on the plates. Candidates that demonstrate a specific interaction phenotype on selective medium should certainly have their inserts sequenced.

5. Notes

1. A number of modified versions of the plasmids and bacterial strains exist. These include a GATEWAY-ready ***(20,21)*** pBR-UV5-α(LP) prey vector, and also counterselectable systems ***(22)*** for assessing interaction disruption in bacterial-based interaction trap systems.
2. *See* the Stratagene website for Bacteriomatch II-compatible libraries in pTRG-series vectors [http://www.stratagene.com/products/showCategory.aspx?catId=78]. In contrast, KJ1567 libraries must be constructed in Ap^R vectors such as the pBR-UV5-αLP vector discussed in this chapter.
3. A metal replicator gives more precision, but requires more practice for reproducible use (inexperienced users often stab holes in agar plates). However, it is also much more expensive to purchase than a plastic replicator; it can easily be homemade (*see* **Fig. 4A**). If the user is constructing their own, it is important that all of the spokes have a flat surface, and that spoke ends are level. The replicator should have 48 spokes in a 6×8 configuration to fit to a standard 90-mm Petri plate (thus, a plastic replicator from Bel-Art, made to fit 12×8 plate, must be cut in half). A metal replicator can be sterilized by autoclaving or by alcohol/flaming; a plastic replicator can be sterilized by autoclaving or by rinsing with alcohol and air-drying.
4. So far, no autoactivating eukaryotic baits have been reported for the cI fusion-based bacterial two-hybrid system. However, because of the still limited number of

published uses of bacterial two-hybrid systems compared with yeast two-hybrid systems, it still remains a possibility. In any case, performance of the autoactivation assay provides an opportunity to work out the conditions for the subsequent screen. In this assay, a combination of interacting proteins is used as a positive control.

5. A phage immunity assay can be run to check for both the stability and DNA-binding capability inside a bacterial cell, as discussed in **ref. *15***.
6. Standard molecular biology techniques or alternative cloning strategies (i.e., in vivo recombination ***[23]*** or GATEWAY cloning ***[20]***) can be used. The bait-encoding cDNA should be cloned in-frame with the λcI DBD, and a translation stop codon should be created in-frame at the end of the bait sequence. If screens in both bacteria and yeast are planned using a single bait prepared in pGLS23, bait construction is subject to additional limitations required for use in the yeast system (*see* commentary in Chapter 15).
7. An efficient transformation yields approx 10^7 transformants per μg of DNA (when two plasmids are being simultaneously transformed). Therefore, this experiment also provides a good chance to assess transformation efficiency, which will be of considerable importance at the time of library transformation. If only a very small number of colonies are obtained, or colonies are not apparent within 24 h, this implies that transformation is very inefficient, and results obtained in characterization experiments may not be typical. In this case all solutions, media, and conditions must be double-checked or prepared fresh and transformation be repeated. If very few transformants containing the bait plasmid appear (compared with the controls), or bacteria expressing the bait protein grow noticeably more poorly than controls, or if colony population appears much more heterogeneous than control (e.g., presents a mix of large and small colonies), this would suggest that the bait protein is somewhat toxic to the *E. coli*.
8. If one is planning to use the Bacteriomatch II strain for screening, a pTRG-based plasmid (pTRG-RGL) should be used instead of pLibB-Raf. Accordingly, ampicillin in the medium should be replaced with Tc throughout the protocol.
9. For simplicity, the same prey plasmid pLibB-Raf (or pTRG-RGL) is used throughout this experiment, under the assumption that most bait proteins will not interact with BRaf (or RGL). At the researcher's discretion (and if the researcher is, e.g., studying Raf-interacting proteins), these plasmids can be substituted for empty vectors in reaction mix (**a**).
10. At this step, the ability of a cI-fused bait to activate transcription should be tested on both *HIS3* and *aadA* reporters, whereas potential activation of the *LacZ* reporter (in AG58[RP28] strain) is only optional. The auxotrophic reporter is the most important for the library screening because it allows direct selection for interaction phenotype. In addition, there is normally a good correlation between activation of the two reporters, so it is very unlikely that a bait that does not activate *HIS3* will significantly activate *LacZ*. Therefore, if no activation is detected on His-deficient plates, one should proceed further; if the bait causes growth on His-deficient plates, it should be modified, regardless of its phenotype with the *LacZ* reporter.

11. There are options described further in this protocol to introduce the library (as infectious phage) or the specific library plasmid into the bait-containing reporter strain. If one of these options is going to be used, then approx 5% of each transformation should be plated on an LB_C plate to produce "bait only" colonies. After these have grown, they should be transferred to a master plate and preserved while other tests are ongoing.
12. This is important, because for some baits, protein expression level is heterogeneous between independent colonies, with accompanying heterogeneity of apparent ability to activate transcription of the reporter(s). This is less of a problem with the bacterial two-hybrid system than the yeast two-hybrid system, but it is good to be careful.
13. When making prints on a plate, dip the replicator in the wells of the microtiter plate, then very gently put it on the surface of the solidified medium. Tilt slightly in circular movement, then lift replicator and put it back in the microtiter plate (keep the correct orientation). Make sure all the drops left on the surface are of approximately the same size. If only one or two drops are missing, this is easy to correct by dropping approx 3 µL of yeast suspension on the missing spots from the corresponding wells. If many drops are missing, make sure that all the spokes of the replicator are in good contact with liquid in the microtiter plate (it may be necessary to cut off the side protrusions on the edge spokes of the plastic replicator) and redo the whole plate. Continue replicating by shuttling back and forth between microtiter and media plates. Let the liquid absorb to the agar before putting the plates upside down in the incubator.
14. Basal expression levels of some baits (e.g., pBaitC-zip) may be so high that the addition of IPTG causes no visually discernible difference.
15. This initial test includes comparing the candidate clones with the positive and negative controls characterized in **Subheading 3.1.2.**, **step 5**; make sure these controls are available fresh at the time of library screening. If the master plate is more than 1 wk old, restreak and grow anew; if more than 2 wk old, retransform.
16. The plating efficiency of positive clones and appearance of background colonies can be tested by plating dilutions of the positive and negative clones from **Subheading 3.1.2.**
17. The strongest interactors are not *always* the most biologically meaningful. In **Fig. 4B**, compare growth of the spots a2, a5, and b5, all of which represent a known interaction between HRas and RGL-2, with the growth of the spot c3, which represents a previously undescribed (and probably nonphysiological) interaction between HRas and LTBP4.
18. The absence of a PCR product from the purified plasmid indicates general PCR problems (reagents, faulty amplifier). The absence of a PCR product from the positive control *E. coli* strains (under conditions in which the PCR from the purified plasmid works well) indicates a need to adjust the amount of bacteria taken for the reaction (either too low, so there is not enough template, or too high, which inhibits the DNA polymerase).
19. Perform a restriction digest of up to 10 µL of the unpurified PCR product with *Hae*III in a total volume of 20 µL. Rearrange the loading order according to the

results obtained with nondigested PCR, and load the digestion products on a 1.5% agarose gel. Run out the DNA fragments a sufficient distance to get good resolution of DNA products in the 200–1000-bp size range. This will generally yield distinctive and unambiguous groups of inserts, confirming whether multiple isolates of a small number of cDNAs have been obtained.

20. Primer FPP1 provides enough specificity to sequence library plasmid even from this crude mixture.
21. It is possible to use 5 µL of unpurified plasmid mixture (*see* **Subheading 3.2.1., step 2)** to transform the original reporter strain (and, in parallel, pBaitC-zip and/or pBaitC-Ras control bait strains) used in **Subheading 3.1.1.** (*see also* **Note 11**).
22. A further option is to test a few additional bait proteins that are related to one's protein of interest and a few that are unrelated. Interaction with related bait proteins and not with unrelated bait proteins might indicate that the isolated prey specifically interacts with a family of proteins, whereas interaction with any nonspecific baits tested can indicate a widespread or a nonspecific interaction.

References

1. Fields, S. and Song, O. (1989) A novel genetic system to detect protein-protein interaction. *Nature* **340,** 245–246.
2. Fashena, S. J., Serebriiskii, I. G., and Golemis, E. A. (2000) The continued evolution of hybrid screening approaches in yeast: how to outwit different baits with different preys. *Gene* **250,** 1–14.
3. Dove, S. L., Joung, J. K., and Hochschild, A. (1997) Activation of prokaryotic transcription through arbitrary protein-protein contacts. *Nature* **386,** 627–630.
4. Dove, S. L. and Hochschild, A. (1998) Conversion of the omega subunit of *Escherichia coli* RNA polymerase into a transcriptional activator or an activation target. *Genes Dev*. **12,** 745–754.
5. Joung, J. K., Ramm, E. I., and Pabo, C. O. (2000) A bacterial two-hybrid selection system for studying protein-DNA and protein-protein interactions. *Proc. Natl. Acad. Sci. USA* **97,** 7382–7387.
6. Vallet-Gely, I., Donovan, K. E., Fang, R., Joung, J. K., and Dove, S. L. (2005) Repression of phase-variable cup gene expression by H-NS-like proteins in Pseudomonas aeruginosa. *Proc. Natl. Acad. Sci. USA* **102,** 11,082–11,087.
7. Dove, S. L. and Hochschild, A. (2004) A bacterial two-hybrid system based on transcription activation. *Methods Mol. Biol*. **261,** 231–246.
8. Serebriiskii, I. G., Fang, R., Latypova, E., et al. (2005) A combined yeast/bacteria two-hybrid system: development and evaluation. *Mol. Cell Proteomics* **4,** 819–826.
9. Dove, S. L., Huang, F. W., and Hochschild, A. (2000) Mechanism for a transcriptional activator that works at the isomerization step. *Proc. Natl. Acad. Sci. USA* **97,** 13,215–13,220.
10. Shaywitz, A. J., Dove, S. L., Kornhauser, J. M., Hochschild, A., and Greenberg, M. E. (2000) Magnitude of the CREB-dependent transcriptional response is determined by the strength of the interaction between the kinase-inducible domain of CREB and the KIX domain of CREB-binding protein. *Mol. Cell Biol*. **20,** 9409–9422.

11. Struhl, K., Cameron, J. R., and Davis, R. W. (1976) Functional genetic expression of eukaryotic DNA in *Escherichia coli. Proc. Natl. Acad. Sci. USA* **73,** 1471–1475.
12. Struhl, K. and Davis, R. W. (1977) Production of a functional eukaryotic enzyme in *Escherichia coli*: cloning and expression of the yeast structural gene for imidazole-glycerolphosphate dehydratase (his3). *Proc. Natl. Acad. Sci. USA* **74,** 5255–5259.
13. Brennan, M. B. and Struhl, K. (1980) Mechanisms of increasing expression of a yeast gene in *Escherichia coli. J. Mol. Biol*. **136,** 333–338.
14. Hollingshead, S. and Vapnek, D. (1985) Nucleotide sequence analysis of a gene encoding a streptomycin/spectinomycin adenylyltransferase. *Plasmid* **13,** 17–30.
15. Giesecke, A. V. and Joung, J. K. (2005) Bacterial Two-hybrid System for Studying Protein-Protein Interactions, in *Protein-Protein Interactions: A Molecular Cloning Manual*, (Golemis, E. A., ed.), Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, pp. 195–216.
16. Miller, J. H. (1992) *A short course in bacterial genetics*, *1st ed.*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
17. Sambrook, J. and Russell, D. (eds.), (2001) *Molecular cloning: a laboratory manual*, *3rd ed.*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
18. Thibodeau, S. A., Fang, R., and Joung, J. K. (2004) High-throughput beta-galactosidase assay for bacterial cell-based reporter systems. *Biotechniques* **36,** 410–415.
19. Serebriiskii, I. G. and Golemis, E. A. (2000) Uses of lacZ to study gene function: evaluation of beta-galactosidase assays used in the yeast two-hybrid system. *Anal. Biochem*. **285,** 1–15.
20. Hartley, J. L., Temple, G. F., and Brasch, M. A. (2000) DNA cloning using in vitro site-specific recombination. *Genome Res*. **10,** 1788–1795.
21. Walhout, A. J., Temple, G. F., Brasch, M. A., et al. (2000) GATEWAY recombinational cloning: application to the cloning of large numbers of open reading frames or ORFeomes. *Methods Enzymol*. **328,** 575–592.
22. Meng, X., Smith, R. M., Giesecke, A. V., Joung, J. K., and Wolfe, S. A. (2006) Counter-selectable marker for bacterial-based interaction trap systems. *Biotechniques* **40,** 179–184.
23. Stanyon, C. A., Limjindaporn, T., and Finley, R. L., Jr. (2003) Simultaneous cloning of open reading frames into several different expression vectors. *Biotechniques* **35,** 520–522, 524–526.

17

Engineering Cys2His2 Zinc Finger Domains Using a Bacterial Cell-Based Two-Hybrid Selection System

Stacey Thibodeau-Beganny and J. Keith Joung

Summary

Synthetic Cys2His2 zinc finger domains with novel DNA-binding specificities can be identified from large randomized libraries using selection methodologies such as phage display. It has been previously demonstrated that a bacterial cell-based two-hybrid system is at least as effective as phage display for selecting zinc fingers with desired specificities from these libraries. In this chapter the authors provide updated and detailed protocols for performing zinc finger selections using the bacterial two-hybrid system.

Key Words: Artificial transcription factor; bacterial two-hybrid; Cys2His2 zinc finger; zinc finger nuclease; gene therapy; gene targeting; protein engineering; synthetic biology.

1. Introduction

Artificial Cys2His2 zinc finger domains (C2H2 ZFs) with engineered DNA-binding specificities have shown promise for applications in both biological research and gene therapy ***(1–9)***. Selection-based methods for altering the specificities of individual C2H2 ZFs typically involve randomization of amino acid residues in the DNA-recognition helix to generate large libraries followed by use of a selection method to identify variants with desired DNA-binding specificities. Early studies utilized phage display for the selection method ***(2,10,11)*** but more recent studies have demonstrated that a bacterial cell-based two-hybrid (B2H) system works as well as phage display, and might be, in certain cases, more effective ***(12,13)***. In addition, the B2H system is somewhat easier to use than phage display because it directly selects for proteins in an in vivo, cellular context whereas phage display requires multiple rounds of in vitro selection.

In this chapter, detailed methods are described for using the B2H system to identify individual C2H2 ZFs with desired DNA-binding specificities from randomized libraries more than 10^8 in size. Although, similar protocols have

From: *Methods in Molecular Biology, vol. 408: Gene Function Analysis*
Edited by: M. Ochs © Humana Press Inc., Totowa, NJ

been outlined in previous publications ***(12,13)***, the overall approach has evolved in our laboratory as we have gained experience with the method. The protocol described in detail herein is the most up-to-date method currently utilized by the laboratory for selections.

2. Materials

2.1. Molecular Biology Reagents

10X Annealing buffer (1 mL): 400 µL 1 *M* Tris-HCl, pH 8.0 200 µL 1 *M* $MgCl_2$, 100 µL 5 *M* NaCl, and 300 µL H_2O.

2.2. Primer Sequences

1. OK.181 sequencing primer: 5′-CCAGAGCATGTATCATATGGTCCAGAAA CCC-3′.
2. OK.5 polymerase chain reaction (PCR) primer: 5′-AAAATAGGCGTATCAC-GAGGCCCT-3′.
3. OK.163 PCR primer: 5′-CGCCAGGGTTTTCCCAGTCACGAC-3′.
4. OK.61 sequencing primer: 5′-GGGTAGTACGATGACGGAACCTGTC-3′.

2.3. Bacterial Strains

1. CSH100 (genotype: F′ *lac proA*$^+$*B*$^+$ *[lacI*^{q} *lacPL8]/ara*$^-$ Δ *[gpt-lac]5*).
2. KJ1C (genotype: F – Δ*hisB463* Δ *[gpt-proAB-arg-lac] XIII zaj::Tn10*).

These strains are both available from the Joung laboratory (Massachusette General Hospital).

2.4. Bacterial Media

1. Luria Bertani (LB)/TKS plates contain tetracycline, kanamycin, and sucrose.
2. LB/TK plates contain tetracycline and kanamycin.
3. LB/Kan plates contain kanamycin.
4. LB/CCK plates contain carbenicillin, chloramphenicol, and kanamycin.
5. LB/CK plates contain chloramphenicol and kanamycin.
6. LB/Tet plates contain tetracycline.
7. M9 minimal medium plates.

   For 500 mL: autoclave 439 mL H_2O with 7.5 g bacto agar and magnetic stir bar. After agar has cooled to approx 65°C, add 50 mL 10X M9 salts, 1 mL 1 *M* $MgSO_4$, 10 mL 20% glucose, and 0.5 mL 100 m*M* $CaCl_2$ and then pour plates.
8. NM medium (1 L): 836 mL H_2O, 100 mL 10X M9 salts, 20 mL 20% glucose, 10 mL 20 m*M* adenine, 30 mL 200X amino acid mixture (*see* **Step 10**), 1 mL 1 *M* $MgSO_4$, 1 mL 10 mg/mL thiamine, 1 mL 10 m*M* $ZnSO_4$, and 1 mL 100 m*M* $CaCl_2$.
   a. Filter-sterilize and store at 4°C.
   b. Add antibiotics, isopropyl-β-D-thiogalactopyranoside (IPTG), and 3-aminotriazole (3-AT) as desired.
9. NM agar plates (1 L).
   a. Autoclave 836 mL H_2O, 15 g bacto agar, and a magnetic stir bar together.

b. While agar is cooling, mix together the following components in a sterile flask: 100 mL 10X M9 salts, 20 mL 20% glucose, 10 mL 20 m*M* adenine, 30 mL 200X amino acid mixture (*see* below), 1 mL 1 *M* $MgSO_4$, 1 mL 10 mg/mL thiamine, 1 mL 10 m*M* $ZnSO_4$, 1 mL 100 m*M* $CaCl_2$, and antibiotics, IPTG, and 3-AT as desired.
c. When agar has reached a temperature of approx 65°C, add the above mixture to the agar, stir well, and pour plates.
d. NM/CCKI plates are NM agar plates supplemented with carbenicillin (100 µg/mL), chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and IPTG (50 µ*M*).

10. 200X Amino acid mixture: each of the six solutions below should be made separately with ingredients mixed together in the order listed. The six solutions are then mixed together, filter-sterilized, and stored at 4°C (*see* **Note 1**). This yields a 200X stock containing all amino acids except histidine, cysteine, and methionine.
    a. Solution I (100 mL): dissolve 0.99 g phenylalanine, 1.10 g lysine, and 2.50 g arginine in H_2O.
    b. Solution II (100 mL): dissolve 0.20 g glycine, 0.70 g valine, 0.84 g alanine, and 0.41 g tryptophan in H_2O.
    c. Solution III (100 mL): dissolve 0.71 g threonine, 8.40 g serine, 4.60 g proline, and 0.96 g asparagine in H_2O.
    d. Solution IV (100 mL): add 9.1 mL 36.5% HCl to 80 mL H_2O, dissolve 1.04 g aspartate and 14.60 g glutamine. Adjust final volume to 100 mL with H_2O.
    e. Solution V (100 mL): dissolve 18.70 g K-glutamate in 80 mL H_2O, dissolve 0.36 g tyrosine and 4 g NaOH pellets, and then adjust final volume to 100 mL with H_2O to 100 mL.
    f. Solution VI (100 mL): dissolve 0.79 g isoleucine and 0.79 g leucine in H_2O.
11. Final concentrations of antibiotics and other additives in plates.
    a. Carbenicillin (100 µg/mL); stock is 50 mg/mL in ddH_2O.
    Note: Carbenicillin is used at a final concentration of 50 µg/mL in liquid media.
    b. Chloramphenicol (30 µg/mL); stock is 30 mg/mL in EtOH.
    c. Kanamycin (30 µg/mL); stock is 30 mg/mL in ddH_2O.
    d. Tetracycline (12.5 µg/mL); stock is 12.5 mg/mL in 80% EtOH.
    e. Sucrose (5%); stock is 50% in ddH_2O.
    f. IPTG (50 µ*M*); stock is 50 m*M* in ddH_2O.
    g. 3-AT (varies: 10–60 m*M*); stock is 1 *M* in ddH_2O (*see* **Note 2**).
    h. Streptomycin (varies: 20–80 µg/mL); stock is 100 mg/mL in ddH_2O.

3. Methods

The B2H system, as used in this protocol, links the occurrence of a protein-DNA interaction to the activation of two reporter genes: the yeast *HIS3* and the bacterial *aadA* genes. To do this, a "selection strain" harboring a cocistronic *HIS3/aadA* reporter on a single copy episome is constructed. This reporter also contains a target DNA site of interest positioned just upstream of the weak promoter directing *HIS3/aadA* expression. If a zinc finger domain capable of binding the target DNA site of interest (and fused to a fragment of

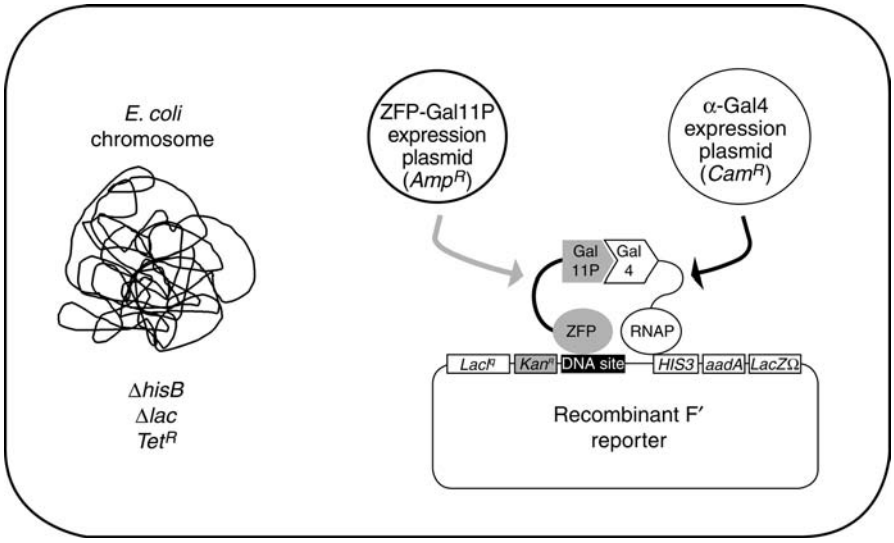


Fig. 1. Schematic overview of the bacterial two-hybrid selection system A selection strain harboring the *HIS3/aadA* reporter and a kanamycin-resistance (*Kan*^{R}) gene on a single copy recombinant F′ is transformed with plasmids encoding a hybrid αGal4 protein (harboring chloramphenicol resistance [*Cam*^{R}]) and a zinc finger domain–Gal11P hybrid protein (harboring carbenicillin resistance [*Amp*^{R}]). If the zinc finger domain binds to the target DNA site present on the F′ reporter (black box), transcription of the *HIS3/aadA* reporter gene cistron is activated through recruitment of RNA polymerase to the reporter promoter mediated by interaction of the Gal11P and Gal4 domains. *See* text for additional details.

the yeast Gal11P protein) is expressed in the selection strain, this leads to recruitment of RNA polymerase to the weak promoter and subsequent activated expression of *HIS3/aadA* transcription; this occurs because the Gal11P fragment interacts with a yeast Gal4 protein fragment that is fused to a subunit of the *Escherichia coli* RNA polymerase α-subunit (a hybrid protein referred to as the αGal4 protein) (*see* **Fig. 1**) ***(13)***. Cells harboring such a zinc finger domain can be identified on medium lacking histidine and containing the antibiotic streptomycin. The stringency of the *HIS3 or aadA* selections can be increased by adding higher concentrations of 3-AT (a competitive inhibitor of the HIS3 enzyme) or streptomycin, respectively, to the medium.

In this section, first the methods for engineering "selection strains" harboring target DNA sequences of interest are described. Then the methods for using these strains to identify zinc finger variants of interest from large randomized libraries are described.

3.1. Selection Strain Construction

Selection strains are constructed in two steps: In an initial step, a target DNA site of interest is synthesized and then cloned into a multicopy plasmid reporter vector designed in the lab. In a second step, a portion of this reporter plasmid is recombined to a single copy F′ episome in bacterial strain CSH100 and the resulting recombinant F′ is then transferred by conjugation to KJ1C, an F^- strain in which one can select for *HIS3* and *aadA* expression ***(13)***. The method of selection strain construction is similar to one previously described by Whipple ***(14)*** but utilizes a counter-selection step that simplifies identification of desired double-recombinants (*see* **Fig. 5**).

3.1.1. Reporter Plasmid Construction

1. Cut approx 1 µg of the reporter plasmid pKJ1712 with *Sap*I (NEB, New England Biolabs). pKJ1712 contains two closely positioned *Sap*I sites (*see* **Fig. 2**), and therefore, digestion with *Sap*I releases a small 45-bp fragment.
   a. 1 µg Stuffer plasmid.
   b. 5 µL 10X NEB buffer 4.
   c. 5 µL *Sap*I (2 U/µL).
   d. Fill to 50 µL with H_2O.
   e. Incubate at 37°C for 2 h.
2. Isolate the 8678-bp pKJ1712 vector backbone on either an agarose or polyacrylamide gel using standard methods (Sambrook and Russell, 2001). Resuspend the final purified digested vector in 20 µL of ddH_2O.
3. Treat the purified vector with Pfu to create extended overhangs. Cloned Pfu DNA polymerase (Stratagene) has a 3′- to 5′-exonuclease activity, and by providing only one nucleotide (dCTP) to the reaction, the enzyme will degrade DNA until it reaches a position that can be filled in with dCTP. At this point, the forward synthesis and reverse exonuclease activities will reach equilibrium, thereby leaving extended overhangs as shown in **Fig. 2**. Reaction conditions for Pfu treatment are as follows.
   2 µL 10 m*M* dCTP.
   2 µL 10X Cloned Pfu buffer (Stratagene).
   10 µL pKJ1712 *Sap*I-digested vector.
   1.2 µL Cloned Pfu (2.5 U/µL).
   4.8 µL H_2O.
   72°C for 15 min, 4°C for more than 2 min.
4. As illustrated in **Fig. 3**, design a pair of oligonucleotides, which when annealed together will form a double-stranded DNA fragment bearing the target DNA-binding site and extended overhangs compatible with the pKJ1712 vector prepared in **step 3**, **Subheading 3.1.1.**
5. Anneal the target DNA-binding site oligonucleotides together as follows:
   a. 1 µL Oligo 1 (10 pmol/µL).
   b. 1 µL Oligo 2 (10 pmol/µL).

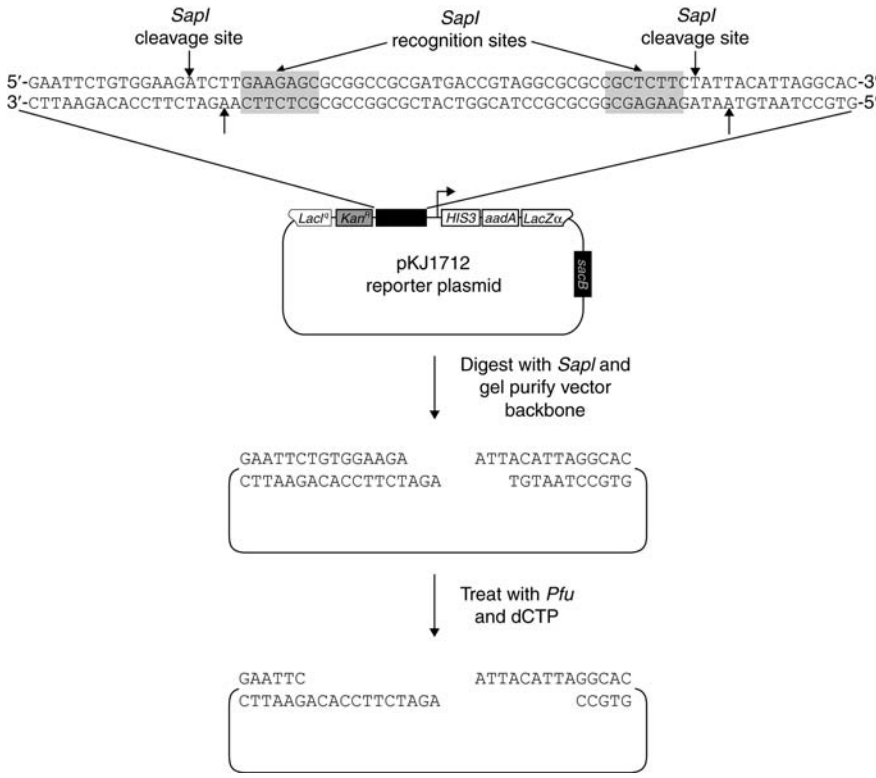


Fig. 2. Structure and sequence of the reporter plasmid vector pKJ1712. A schematic of reporter plasmid pKJ1712 is shown. The region of the plasmid into which target DNA-binding sites are cloned is represented as a black rectangle with corresponding detailed sequence shown. Digestion of pKJ1712 with *Sap*I Type IIS restriction enzyme releases a 45-bp fragment. Further treatment of the *Sap*I-digested vector backbone with *Pfu* DNA polymerase in the presence of dCTP nucleotide results in formation of the DNA overhangs illustrated

c. 20 µL 10X Annealing buffer.
d. 178 µL H_2O.
e. Incubate at 95°C for 2 min, then shut off heat block and let tubes slowly cool to 35°C and then place on ice or 95°C for 2 min → –1°C/min → 25°C → 4°C in a thermocycler. Store at –80°C.

6. Ligate the purified pKJ1712 vector backbone to the annealed oligonucleotide binding site as follows:
   a. 2 µL Purified *Sap*I-digested, Pfu-treated pKJ1712 vector.
   b. 8 µL Annealed binding site oligos.
   c. 10 µL 2X Quick ligase buffer (NEB).
   d. 1 µL T4 DNA ligase (400 U/µL) (NEB).

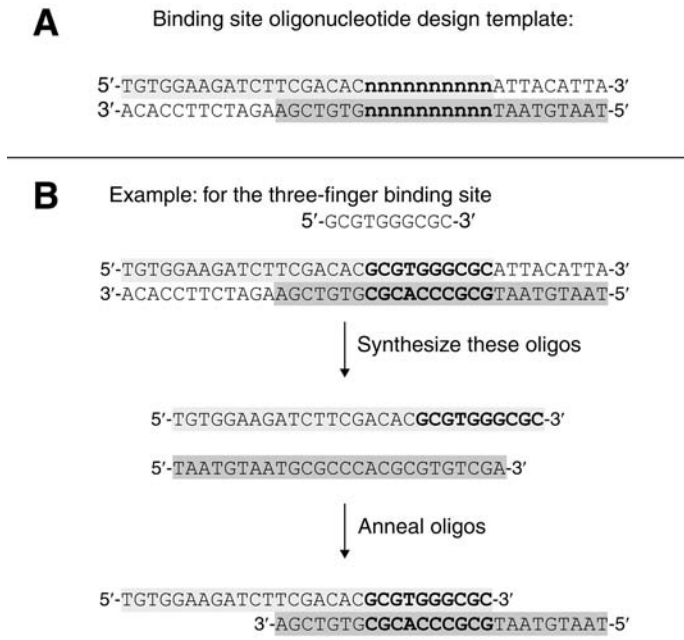


Fig. 3. **(A)** Template for design of oligonucleotides harboring a target DNA site. The target DNA site of interest is substituted for the "nnnnnnnnnn" sequence illustrated in the template. (Note that this sequence can be longer than 10 bp if desired). Oligonucleotides corresponding to the highlighted sequences are then synthesized and annealed together to create an insert that can be ligated to reporter plasmid pKJ1712 (compare overhangs of the annealed oligo complex with the overhangs of *Sap*I-digested, Pfu-treated pKJ1712 shown at the bottom of **Fig. 2**). **(B)** Design of binding site oligonucleotides for the 10-bp Zif268 target DNA site. As an example, this figure illustrates the design of binding site oligonucleotides for the 10-bp Zif268 binding site 5′-GCGTGGGCGC-3′. In this example, the amino-terminal finger of Zif268 binds to the 3′-end of the target sequence whereas the carboxy-terminal finger binds to the 5′-end.

e. Room temperature for 5 min → ice.

A control ligation containing only the *Sap*I-digested pKJ1712 vector (i.e., without annealed oligos) is also performed.

7. Transform ligations into XL-1 Blue *E. coli* competent cells using standard protocols and plate 1/3 of the transformations on LB/kan plates. Incubate plates at 37°C overnight.
8. Isolate miniprep plasmid DNA from transformants. Typically, if there are at least 20-fold more colonies than the control transformation plate, DNA from two different colonies are prepared. As pKJ1712 is a low-copy number plasmid, 10 mL overnight cultures in liquid LB/kanamycin (30 µg/mL) will yield an adequate DNA concentration when using the QIAgen's QIAprep Spin Miniprep Kit.

Typically, DNA is eluted in 50 µL 0.1X EB (a 1X EB stock is provided in the QIAgen kit).

9. To verify uptake of the annealed oligonucleotides in the pKJ1712 plasmid, 5 µL of each candidate DNA is digested with *Eco*RI and *Hind*III. As a control, plasmid pKJ1712 is also digested with *Eco*RI and *Hind*III. Recombinants that have taken up the annealed oligonucleotide insert should yield five bands of sizes 6109, 1006, 963, 431, and 190 bp. By contrast, pKJ1712 should yield five bands of sizes 6109, 1006, 963, 456, and 190 bp (*see* **Note 3**).
10. Plasmids that look correct by restriction digest should be sequenced to confirm the target DNA-binding site. Primer OK.181, a primer which anneals to the sense DNA strand just downstream of (and pointing back toward) the target binding site, to verify the sequence of the insert is used (*see* **Note 4**).

3.1.2. F′ Episome Recombination and Transfer

3.1.2.1. Recombination of Reporter Plasmid Sequences Onto the F′ of Strain CSH100

As shown in **Fig. 4**, the reporter plasmid contains portions of the *lacI*^{q} and *lacZ* gene that are also present in the F′ found in strain CSH100. These regions of matching sequence can serve as point of recombination between the reporter plasmid and the F′. A double crossover event at both regions of sequence identity can lead to transfer of a portion of the reporter plasmid onto the F′.

1. Streak out F^{-} strain KJ1C on a LB/Tet plate and grow at 37°C overnight.
2. The next day, transform F^{+} strain CSH100 with the reporter plasmid and plate enough to obtain a confluent lawn of transformants on LB/Kan plate. Incubate overnight at 37°C.

3.1.2.2. Transfer of F′s From CSH100 to KJ1C by Bacterial Mating

The population of transformed CSH100 cells will contain a small number of cells in which a single recombination event has led to integration of the reporter onto the F′ (**Fig. 5**). In an even smaller number of cells, a double recombination event will have exchanged the target DNA-binding site and promoter present on the reporter plasmid with the promoter on the F′ (**Fig. 5**). As is described in this step, all F′s (recombinant and nonrecombinant) in the CSH100 transformants are transferred to the F^{-} strain KJ1C by mating. In a subsequent step, the desired double recombinant F′ that have been transferred to strain KJ1C can be identified by plating on appropriate selective medium (**Fig. 5** and *see* **Step 9**).

1. Scrape the confluent plate of CSH100 transformants with a sterile wooden stick and transfer cell paste to a sterile 25-mm glass tube containing 10 mL of LB (*see* **Note 5**).

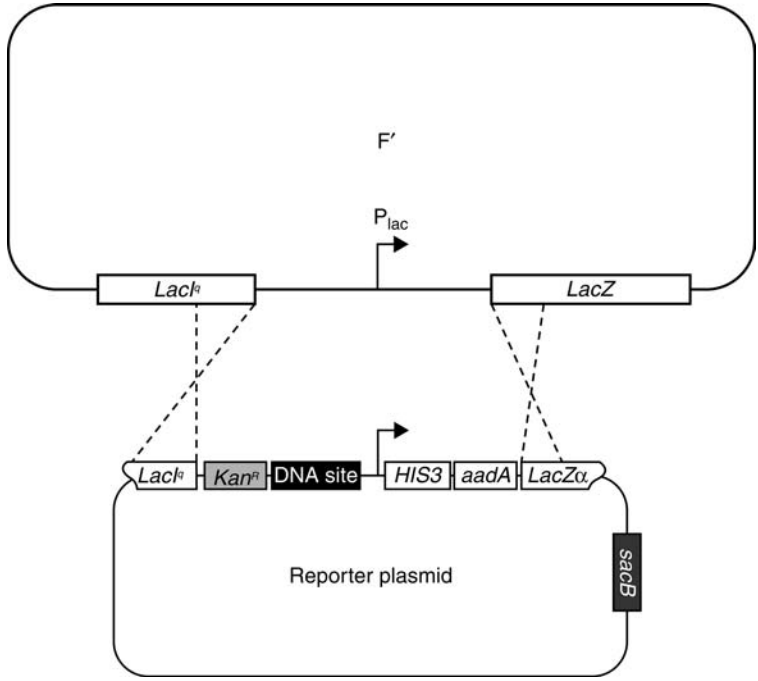


Fig. 4. Homologous recombination between reporter plasmids and the CSH100 strain F′ mediated by regions of sequence identity. Schematic of a reporter plasmid showing its structure and its regions of sequence identity with the F′ from *E. coli* strain CSH100. A double homologous recombination event between the two regions of sequence identity (from the *lacI*^{q} and *lacZ* genes) leads to insertion of a fragment consisting of the *Kan*^{R} gene, the target DNA-binding site, and the *HIS3/aadA* reporter into the F′. Note that a double homologous recombination event does not transfer the counter-selectable *sacB* gene from the reporter plasmid to the F′.

2. Vortex to resuspend the CSH100 transformants (*see* **Note 6**) and transfer approx 200 µL of this cell resuspension to a fresh 25-mm tube with 5 mL of LB (*see* **Note 7**).
3. Transfer approx 200 µL of an overnight culture of KJ1C cells (inoculated the night before, grown in LB containing 12.5 µg/mL of tetracycline) to a sterile 25-mm tube containing 10 mL of LB.
4. Prepare a control 25-mm tube containing 10 mL of LB.
5. Incubate all tubes from **steps 2** to **4** for 2 h at 37°C without agitation, thereby allowing cells to grow to log phase and for CSH100 cells to form F pili.
6. To perform matings, mix together the following combinations of the cultures from **steps 2** to **4** in sterile 18-mm glass tubes. Use 1 mL of each liquid culture (i.e., each mating will consist of a total of 2 mL).

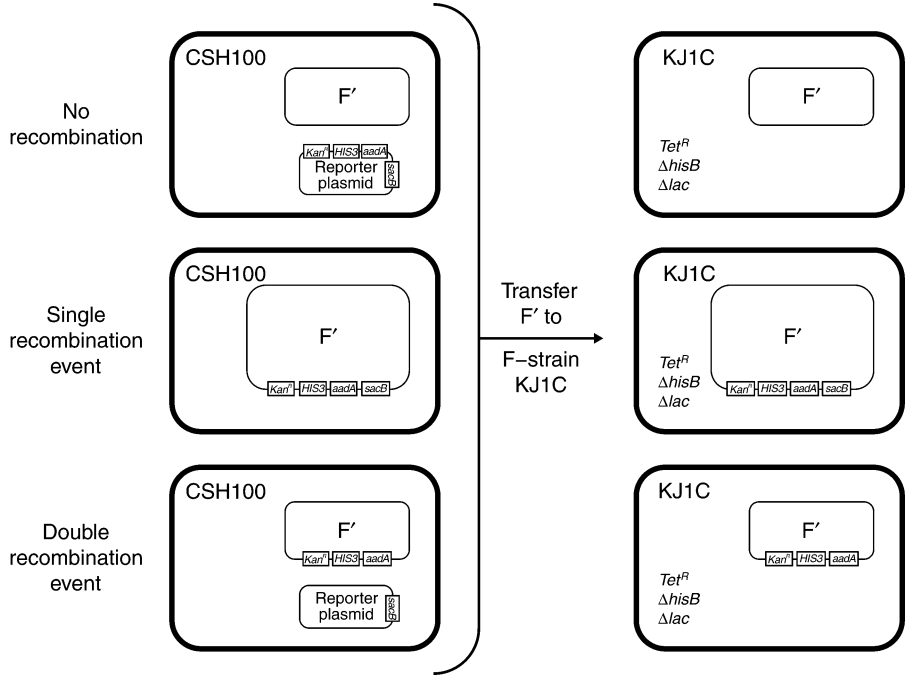


Fig. 5. Construction and identification of selection strains by recombination, mating, and selection. Reporter plasmids transformed into CSH100 either undergo no recombination **(top left)**, single recombination **(middle left)**, or double recombination **(bottom left)** with the F′ present in this strain. Mating of these CSH100 cells with the tetracycline-resistant, F^- strain KJ1C results in transfer of the three different F′s into KJ1C cells (**top**, **middle**, and **bottom right**). When these cells are plated on medium containing tetracycline, kanamycin, and sucrose, only KJ1C cells that have received the double recombinant F′ (the desired selection strain, **bottom right**) will survive. All of CSH100 cells **(left side)** will fail to grow because of sensitivity to tetracycline. KJ1C cells that do not receive an F′, or that receive a nonrecombinant F′ **(top right)** will fail to grow because of sensitivity to kanamycin. KJ1C cells that receive a single recombinant F′ **(middle right)** will fail to grow because of their sensitivity to sucrose owing to the presence of the *sacB* gene. KJ1C cells harboring a double recombinant F′ will be resistant to kanamycin, tetracycline, and sucrose and thus will be the only surviving cells on the plate.

a. CSH100 transformants + KJ1C (actual mating).
b. CSH100 transformants + LB (negative control).
c. KJ1C + LB (negative control).
d. LB + LB (negative control).

Allow matings to proceed at 37°C for 1 h without agitation.

7. Put tubes on the wheel, 37°C for 90 min.
8. Plate 300 µL of the actual mating from **step 6a** above (CSH100 transformants + KJ1C) on a LB/TK plate and on a LB/TKS plate. Spot 5 µL of each of the negative controls on a LB/TK plate and on a LB/TKS plate. Incubate all plates overnight at 37°C.

   As shown in **Fig. 5**, only KJ1C cells harboring the desired double recombinant F′ should grow on LB/TKS plates. All the matings are plated on LB/TK plates as well to check that the counter-selectable marker (*sacB*) is working to eliminate KJ1C cells harboring single recombinant and nonrecombinant F′s (*see* **Fig. 5**). (Note that expression of the *sacB* gene is lethal in *E. coli* cells when they are plated on medium containing sucrose.) Typically, at least a 10-fold reduction in the number of colonies is observed when comparing the number of colonies on LB/TK with the number of colonies on LB/TKS plates.
9. After KJ1C cells harboring the desired double recombinant F′ have been identified on the basis of growth on LB/TKS plates (*see* **Note 8**), clonal isolates have been purified by restreaking candidates two times to LB/TKS plates. (*Note:* KJ1C cells grow quickly, so if they are streaked early in the morning, new pickable colonies will typically be ready by evening to streak again.). Typically, two candidates are chosen (which are designated "A" and "B") for each desired clone because a small percentage of cells that survive on LB/TKS plates will fail additional subsequent verification tests (as described next).

3.1.3. Selection Strain Verification

3.1.3.1. Genetic Confirmation of F′ Transfer

1. Resuspend each colony (A and B candidates) in 100 µL of 1X M9 Salts (this is typically done in wells of a sterile 96-well plate). Spot 5 µL of each cell suspension on an M9 minimal medium plate (*see* **Note 9**).
2. After overnight growth at 37°C, verify growth of cells within the spots. Candidates that fail to grow should be discarded.

3.1.3.2. Sequencing of The Recombinant F′ Reporter

For candidates that successfully demonstrate growth on M9 minimal medium plates, the portion of the F′ harboring the target DNA site is amplified and sequenced to verify the reporter.

1. Use 20 µL of the cell suspension from **step 1**, **Subheading 3.1.3.1.**, to inoculate a 4 mL LB/kanamycin (30 µg/mL) overnight culture.
2. Transfer approx 100 µL of saturated overnight culture to a 1.5-mL microcentrifuge tube. Spin at maximum speed for 1 min in a microfuge. Remove media using a pipet tip and then resuspend cells in 100 µL PCR-grade ddH_2O. Heat the cell resuspension at 95°C for 10 min and then spin at maximum speed for 1 min in a microfuge. Remove 50 µL of the supernatant to a fresh tube and use this crude preparation of genomic DNA as template for a PCR reaction as follows:

PCR conditions	Cycling
5 µL crude template	95°C for 5 min
5 µL 10X Expand buffer (Roche)	94°C for 30 s[a]
4 µL 10 m*M* dNTPs	60°C for 30 s[a]
1 µL OK5 (10 pmol/µL)	72°C for 2.5 min[a]
1 µL OK163 (10 pmol/µL)	72°C for 10 min
0.375 µL Expand enzyme (Roche)	33.625 µL H_2O
	50 µL

[a]Indicates repeated steps in 25 cycles.

3. Run PCR products out on a 5% polyacrylamide gel or 1% agarose gel and isolate the expected approx 1.8-kb DNA fragment. Sequencing of the target DNA-binding site can be performed using primer OK181.
4. Typically, glycerol stocks and competent cells of the selection strain are also prepared using the overnight culture inoculated in **step 1**, **Subheading 3.1.3.2.**

3.1.3.3. Transformation of Recombinant F′ KJ1C Strain With Plasmid pAC-αGal4

The final step in preparation of the selection strain is to transform the KJ1C strain, harboring a sequence verified F′ reporter episome, with the pAC-αGal4 plasmid ***(13)*** (*see* **Note 10**).

1. KJ1C cells bearing a sequence-verified recombinant F′ reporter episome are transformed with plasmid pAC-αGal4 using standard chemical transformation protocols. Transformations are plated on LB/CK plates because the recombinant F′ episome in the KJ1C cells confers resistance to kanamycin whereas the pAC-αGal4 plasmid encodes a chloramphenicol resistance gene.
2. Transformants are inoculated into overnight LB cultures containing chloramphenicol (30 µg/mL) and kanamycin (30 µg/mL) grown overnight at 37°C.
3. Glycerol stocks of the final selection strains are prepared using the overnight culture.

3.2. Selection of C2H2 ZFs Using the Bacterial Two-Hybrid System

To perform selections, phagemid libraries encoding randomized zinc fingers are introduced into a selection strain and then plated on selective media. These zinc finger variants are expressed as fusions to a fragment of the yeast Gal11P protein, which interacts with the fragment of the yeast Gal4 protein present in the αGal4 protein expressed in selection strains. Binding of a zinc finger domain to the target DNA sequence leads to recruitment of RNA polymerase complexes that have incorporated the αGal4 hybrid protein. This recruitment in turn leads to activation of reporter gene expression and survival on selective medium.

Typically, selections are performed in two stages. In stage A selections, large numbers of transformants (typically ~10^9) are plated on a low-stringency selection plate. Zinc finger-encoding phagemids are rescued from surviving cells. In stage B selections, this enriched population of phagemids are then reintroduced into the selection strain cells and plated on a series of higher stringency selection plates. Phagemid DNA is then isolated from cells that grow on the highest stringency plate and sequenced to determine the identity of the fingers.

3.2.1. Stage A Selections

1. Streak out the selection strain on an LB/CK plate and incubate overnight at 37°C.
2. Use a fresh, well-isolated colony to start an overnight culture of the strain in 20 mL of NM media supplemented with chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and 50 m*M* IPTG. Because selection strain cells are sensitive to detergents and rapid agitation, this culture should be grown in a sterile 125-mL glass flask that was rinsed thoroughly with deionized distilled water before autoclaving and with shaking at 110 rpm at 37°C for approx 16–24 h.
3. Introduction of zinc finger phagemid libraries into selection strain cells. The construction of the randomized zinc finger phagemid libraries used in this step has been previously described ***(12,13)***.
   a. Thaw phagemid phage library on ice. Extreme care is required to prevent phage contamination in the lab as it may persist.
   b. Transfer 5 mL of the saturated overnight culture of selection strain cells to a sterile 125-mL flask.
   c. Add approx 6×10^8 ampicillin transducing units (*see* **Note 11**) of phagemid library to the selection strain cells and gently swirl immediately. Allow the cell/phagemid mixture to sit at room temperature for 30 min without agitation.
   d. Add 20 mL of prewarmed NM media supplemented with chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and 50 m*M* IPTG to the infected cells. Shake at 110 rpm, 37°C for 90 min.
   e. Transfer culture to a sterile 50-mL conical tube and pellet cells by spinning at 2500 rpm in a table top centrifuge for 25 min at room temperature.
   f. Pour off medium and resuspend the cell pellet in 2.5 mL of prewarmed NM medium supplemented with chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and 50 m*M* IPTG.
   g. Serially dilute 100-µL aliquots of the cell resuspension in a sterile 96-well microtiter plate (Corning, Cat. no. 3596). Perform three independent dilution sets using NM medium supplemented with chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and 50 m*M* IPTG. Perform dilutions from 10^{-1} to 10^{-8} (note that one will only plate dilutions 10^{-3}–10^{-8}).
   h. Spot 5 µL of the 10^{-3} to 10^{-8} dilutions each in triplicate on LB/CK, LB/CCK, and NM/CCK/50 µ*M* IPTG plates. Incubate LB/CK and LB/CCK plates for

16–18 h at 37°C and NM/CCK/50 μ*M* IPTG plates for 24 h at 37°C. Titers from these plates can be calculated the next day after colonies have formed (*see* **Step m**).

i. Pour some sterile glass beads (Fisher Scientific, cat. no. 11-312A) (3 mm) onto a large (245 × 245 mm^2) NM/CCK/50 μ*M* IPTG/10 m*M* 3-AT plate.
j. Measure the remainder of the cell suspension, record the value (this volume will be used when calculating titers), and then add the cells to the NM/CCK/50 μ*M* IPTG/10 m*M* 3-AT plate.
k. Spread the cells with circular motions using the beads to distribute the cells evenly.
l. When plates have dried, turn plates over, and tap beads from the agar onto the inverted plate cover. Incubate at 37°C for 24 h and then at room temperature for 18 h.
m. The next day, count colonies on the serial dilution plates to calculate the total number of cells and total number of transformed cells plated on the selection plate. LB/CK plates are used to determine the total number of cells plated and the LB/CCK and NM/CCK/I50 plates are used to determine the total number of transformed cells plated. The following formula is used to perform these calculations:
(No. of colonies/volume of spots [μL]) × dilution factor × volume (μL) plated on large dish.
Example 65 total colonies in nine 5 μL spots of a 10^{-6} dilution together with 2650 μL plated on the large plate would give the following equation: (65 colonies/45 μL) × 10^6 × 2650 μL = 3.83 × 10^9 cells

4. Recovery of zinc finger-encoding plasmids from cells surviving the selection. In this step, zinc finger-encoding plasmids from surviving cells are rescued as phagemids by infecting these cells with M13K07 helper phage.
   a. Turn the large selection plate over and tap the glass beads back onto the agar and add 15 mL prewarmed NM media to the plate. Move the plates in a circular motion using the glass beads to resuspend the cells in the media.
   b. Transfer the suspension to a sterile 25-mm glass tube.
   c. Remove 3 mL of cell resuspension to make glycerol stocks in case this recovery step needs to be redone.
   d. Add enough of the cell suspension to 90 mL 2XYT supplemented with carbenicillin (50 μg/mL) and kanamycin (30 μg/mL) to give it a prelog appearance (i.e., an OD_{600} of ~0.1). Shake this culture at 120 rpm, 37°C for 1 h.
   e. Infect the log-phase culture with 10^{12} kanamycin-transducing units of M13K07 helper phage. Allow the phage to adsorb to the cells, without shaking, at room temperature for 30 min.
   f. Add kanamycin to a final concentration of 100 μg/mL (including the original 30 μg/mL present in the culture). Shake the culture at 125 rpm, 37°C for 6 h. During this incubation, zinc finger-encoding phagemids from the cells will be packaged as infectious phage particles harboring single-stranded DNA molecules and extruded into the culture medium.

g. Harvest the zinc finger-encoding phagemid phage by filtering the culture through a 0.2 µm polyethersulfone (PES) filter membrane (no need to centrifuge the cells away first). This enriched phagemid phage library can be stored at 4°C for several weeks (for long-term storage, one freezes the phage at −80°C).

3.2.2. Stage B Selections

3.2.2.1. Introduction of Enriched Library of Zinc Finger-Encoding Phagemid Phage Into the Selection Strain

1. Start a 20 mL overnight culture of the selection strain in NM medium supplemented with chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and 50 m*M* IPTG in a sterile 125-mL flask. Shake 16–24 h at 110 rpm, 37°C.
2. In a 96-well plate, aliquot 50 mL of the selection strain overnight culture into six wells.
3. In another column of a 96-well plate, add 100 µL of the enriched phagemid phage library to one well. Perform serial fivefold dilutions of the enriched library by removing 20 µL of phage and adding it to a well containing 80 µL of NM medium supplemented with chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and IPTG (50 m*M*). Repeat to create 2E-1, 4E-2, 8E-3, 1.6E-3, and 3.2E-4 dilutions.
4. Infect each of the wells containing 50 µL of selection strain overnight culture with 10 µL of each of the following: undiluted enriched phagemid library and 2E-1, 4E-2, 8E-3, 1.6E-3, and 3.2E-4 dilutions of the phagemid library. Allow phage to adsorb by leaving them (without shaking) at room temperature for 30 min.
5. Add 190 µL of prewarmed NM medium containing chloramphenicol (30 µg/mL), kanamycin (30 µg/mL), and IPTG (50 m*M*) to each well. Incubate for 2 h at 37°C (no shaking).
6. Spot 5-µL aliquots of the phagemid-infected selection strain cells on the following plates:
   a. NM/CCKI plates (six 5 µL spots on standard Petri dishes).
   b. NM/CCKI/20 m*M* 3-AT/20 µg/mL streptomycin (10 spots [5-µL each] on small square 100×100 mm^2 plates).
   c. NM/CCKI/25 m*M* 3-AT/40 µg/mL streptomycin (10 spots [5-µL each] on small square 100×100 mm^2 plates).
   d. NM/CCKI/40 m*M* 3-AT/60 µg/mL streptomycin (10 spots [5-µL each] on small square 100×100 mm^2 plates).
7. Incubate plates 37°C for 48 h and inspect for colonies. Colonies may not form on NM/CCKI/40 m*M* 3-AT/60 µg/mL streptomycin plates until approx 72–96 h of incubation at 37°C.

3.2.2.2. Isolation and Sequencing of Plasmid DNA From Selected Colonies

1. Pick 8–12 well-isolated colonies from the highest stringency selection plate on which colonies appear and inoculate them into 4 mL of LB supplemented with carbenicillin (50 µg/mL). Incubate overnight at 37°C with agitation.

2. Prepare miniprep plasmid DNA from the saturated 4 mL overnight cultures using QIAgen's QIAprep Spin Miniprep Kit (cat. no. 27106) and their protocol with the following differences:
   a. Perform triple washes with both PB and PE buffers (*see* **Note 12**).
   b. Elute the DNA with 60 µL of prewarmed (60°C) 0.1X EB (*see* **Note 13**).
3. Send the plasmids for sequencing with sequencing primer OK.61, a sense strand primer, which anneals just upstream of the region encoding the zinc finger domains.

4. Notes

1. It is important to dissolve the amino acids in each of the six solutions in precisely the order listed as this avoids potential solubility issues. Typically, the amino acid mixture is kept for no more than 2–3 mo.
2. 3-AT should be prepared using gloves. In addition, it has been found that the solubility and purity of 3-AT varies from lot to lot. Some preparations have the appearance of a white powder whereas others look like brown flakes. For certain lots, it has been found that heating the solution to 50°C can aid with solubility.
3. These digests are run on 5% polyacrylamide gels made with 0.5X TBE buffer to visualize the relatively small change in fragment size in clones that have taken up the annealed oligonucleotide insert.
4. The entire sequence between the unique *Eco*RI site (positioned just upstream of the zinc finger domain-binding site) and the unique *Sal*I site (positioned at the start site of transcription) will be sequence verified. Verifying this entire span of sequence ensures that both the zinc finger-binding site and the promoter do not have undesired mutations.
5. It has been found that using a resuspension of multiple transformed CSH100 colonies rather than an overnight culture grown from a single transformed colony helps ensure that a relatively consistent percent of transformed CSH100 cells contain the desired double-recombinant F′.
6. Set vortex to half-maximum speed to ensure that resuspension does not spill over the top of the glass tube.
7. The initial density of this subculture should correspond to OD_{600} of approx 0.1 (i.e., prelogarithmic phase). Depending on the density of the resuspension culture, more or less culture will be added as needed, to achieve this target OD_{600}.
8. Occasionally, some small colonies are observed on the LB/TKS plates. Picking these colonies is avoided because it has been found that these colonies do not yield the desired recombinants.
9. KJ1C cells will not grow unless proline and histidine are provided in their media owing to deletion of the *proAB* gene cluster and a deletion within the *hisB* gene, respectively. A double recombinant F′ transferred from CSH100 cells harbors an intact *proAB* gene cluster and expresses a low level of the yeast *HIS3* gene, which is sufficient to complement the *hisB* deletion of strain KJ1C. Thus, KJ1C cells that receive a double recombinant F′ should be able to grow on M9 minimal medium lacking proline and histidine.

10. pAC-αGal4 encodes a fusion protein consisting of the N-terminal domain and interdomain linker of the *E. coli* RNA polymerase α-subunit, fused to amino acid residues 58–97 of the yeast Gal4 protein. Expression of the αGal4 hybrid protein from pAC-αGal4 is directed by a strong, IPTG-inducible semisynthetic lpp/lacUV5 promoter. The pAC-aGal4 plasmid possesses a p15A origin of replication and confers resistance to chloramphenicol.
11. Typically, it is aimed for a threefold oversampling of the size of the randomized library being interrogated and for a fivefold ratio of total cells to transformed cells. For example, for a randomized library with a complexity of approx 2×10^8, one would aim to plate a total of approx 6×10^8 transformed cells and of more than 3×10^9 total cells on the selection plate.
12. It has been found that these triple washes are critical for obtaining good quality sequencing reads. It is believed that these washes help reduce contaminating endonuclease activity from the endA$^+$ selection strains.
13. 0.1X EB is buffer EB from the QIAgen miniprep kit diluted 10-fold with ddH_2O.

Acknowledgments

This work was supported by grants from the National Institutes of Health (K08 DK002883 and R01 GM069906) and start-up funds from the Massachusetts General Hospital Department of Pathology. J.K.J. dedicates this work to Robert L. Burghoff, Ph.D., who always taught and shared his best protocols.

References

1. Jamieson, A. C., Miller, J. C., and Pabo, C. O. (2003) Drug discovery with engineered zinc-finger proteins. *Nat. Rev. Drug Discov.* **2,** 361–368.
2. Beerli, R. R. and Barbas, C. F., 3rd. (2002) Engineering polydactyl zinc-finger transcription factors. *Nat. Biotechnol.* **20,** 135–341.
3. Blancafort, P., Segal, D. J., and Barbas, C. F., 3rd. (2004) Designing transcription factor architectures for drug discovery. *Mol. Pharmacol.* **66,** 1361–1371.
4. Durai, S., Mani, M., Kandavelou, K., Wu, J., Porteus, M. H., and Chandrasegaran, S. (2005) Zinc finger nucleases: custom-designed molecular scissors for genome engineering of plant and mammalian cells. *Nucleic Acids Res.* **33,** 5978–5990.
5. Porteus, M. H. (2005) Mammalian gene targeting with designed zinc finger nucleases. *Mol. Ther*. **13(2),** 438–446.
6. Porteus, M. H. and Baltimore, D. (2003) Chimeric nucleases stimulate gene targeting in human cells. *Science* **300,** 763.
7. Porteus, M. H. and Carroll, D. (2005) Gene targeting using zinc finger nucleases. *Nat. Biotechnol.* **23,** 967–973.
8. Urnov, F. D., Miller, J. C., Lee, Y. L., et al. (2005) Highly efficient endogenous human gene correction using designed zinc-finger nucleases. *Nature* **435,** 646–651.
9. Alwin, S., Gere, M. B., Guhl, E., et al. (2005) Custom zinc-finger nucleases for use in human cells. *Mol. Ther.* **12(4),** 610–617.
10. Choo, Y. and Klug, A. (1995) Designing DNA-binding proteins on the surface of filamentous phage. *Curr. Opin. Biotechnol*. **6,** 431–436.

11. Pabo, C. O., Peisach, E., and Grant, R. A. (2001) Design and selection of novel Cys2His2 zinc finger proteins. *Annu. Rev. Biochem.* **70,** 313–340.
12. Hurt, J. A., Thibodeau, S. A., Hirsh, A. S., Pabo, C. O., and Joung, J. K. (2003) Highly specific zinc finger proteins obtained by directed domain shuffling and cell-based selection. *Proc. Natl. Acad. Sci. USA* **100,** 12,271–12,276.
13. Joung, J. K., Ramm, E. I., and Pabo, C. O. (2000) A bacterial two-hybrid selection system for studying protein-DNA and protein-protein interactions. *Proc. Natl. Acad. Sci. USA* **97,** 7382–7387.
14. Whipple, F. W. (1998) Genetic analysis of prokaryotic and eukaryotic DNA-binding proteins in *Escherichia coli*. *Nucleic Acids Res*. **26,** 3700–3706.

Index