Chapter 6

Prediction and Integration of Regulatory and Protein–Protein Interactions

Duangdao Wichadakul, Jason McDermott, and Ram Samudrala

Abstract

Knowledge of transcriptional regulatory interactions (TRIs) is essential for exploring functional genomics and systems biology in any organism. While several results from genome-wide analysis of transcriptional regulatory networks are available, they are limited to model organisms such as yeast *(1)* and worm *(2)*. Beyond these networks, experiments on TRIs study only individual genes and proteins of specific interest. In this chapter, we present a method for the integration of various data sets to predict TRIs for 54 organisms in the Bioverse *(3)*. We describe how to compile and handle various formats and identifiers of data sets from different sources and how to predict TRIs using a homology-based approach, utilizing the compiled data sets. Integrated data sets include experimentally verified TRIs, binding sites of transcription factors, promoter sequences, protein subcellular localization, and protein families. Predicted TRIs expand the networks of gene regulation for a large number of organisms. The integration of experimentally verified and predicted TRIs with other known protein–protein interactions (PPIs) gives insight into specific pathways, network motifs, and the topological dynamics of an integrated network with gene expression under different conditions, essential for exploring functional genomics and systems biology.

Key words: Regulog, interolog, protein–DNA interaction prediction, transcriptional regulatory interaction (TRI) prediction, protein–protein interaction (PPI) prediction, homology-based approach, transferability of homologs.

1. Introduction

Transcriptional regulation controls the production of functional gene products essential for determining cell structure and function. It controls the amount of gene product and replenishment of degraded protein. This is fundamental for the differentiation, morphogenesis, versatility, and adaptability of the cell. Expanding the knowledge of gene regulation and understanding how it

Jason McDermott et al. (eds.), *Computational Systems Biology, vol. 541*

DOI 10.1007/978-1-59745-243-4_6

relates to protein–protein interactions and gene expression provides insight into gene function and the mapping from genotypes to phenotypes. This knowledge is fundamental for advancing the design and development of biotechnology and medical treatments.

Though there have been several genome-wide studies of transcriptional regulatory networks, they have been focused on only a few model organisms *(1, 4–7)*. Aside from those formed by genome-wide studies, only networks comprising small, specific, well-studied pathways are available *(8–10)*. Based on publicly accessible databases of genome-wide transcriptional data and regulatory interactions with experimental verification *(9, 11–13)*, several computational studies have reported the building of transcriptional regulatory networks *(14–19)*, based upon the prediction of binding sequences of DNA and binding sites of transcription factors *(20–26)*.

Among these approaches, the transferability of biological functions between homologous genes, originally proposed by Yu et al. *(27)*, has been widely studied and deployed *(3, 27–35)*. Thus, this chapter explores the transferability of protein–DNA interactions between organisms, or regulogs. This approach presumes that similarities in the sequence and structure of gene products suggest similar function.

We predict TRIs for an organism based on the transferability of similar interactions from other source organisms (*see* **Fig. 6.1**). In other words, we try to map the available TRIs in a source organism onto the target organism to find similar interactions.

The similarities of a predicted TRI ($I_{TFx'\rightarrow TFTy'}$) transferred from a source organism is defined as the geometric mean (the square root) of sequence similarities between (1) a transcription factor of an interaction in a source organism (TF) and its ortholog (**Note 1**) in a target organism (TF′) defined as $I_{TF\text{-}TFx'}$, and (2) a sequence of transcription factor target in the source organism (TFT) and its ortholog in the target organism (TFT′) defined as $I_{TFT\text{-}TFTy'}$. To map a TRI from a source organism onto a target organism, the interaction in the target organism needs to satisfy the following three conditions *(27)*:

i) TF and TF′ are orthologs.

ii) TFT and TFT′ are orthologs.

iii) The binding sites and binding sequences of TF appear in the upstream region of the TFT′.

Corresponding to the above conditions, if a TF regulating a TFT in a source organism has orthologs TF′ and TFT′ in a target organism, the pair of the interactions TF → TFT and TF′→ TFT′ are called regulogs (**Note 2**) (*see* **Fig. 6.1**). We can improve the accuracy and coverage of the resulting predictions

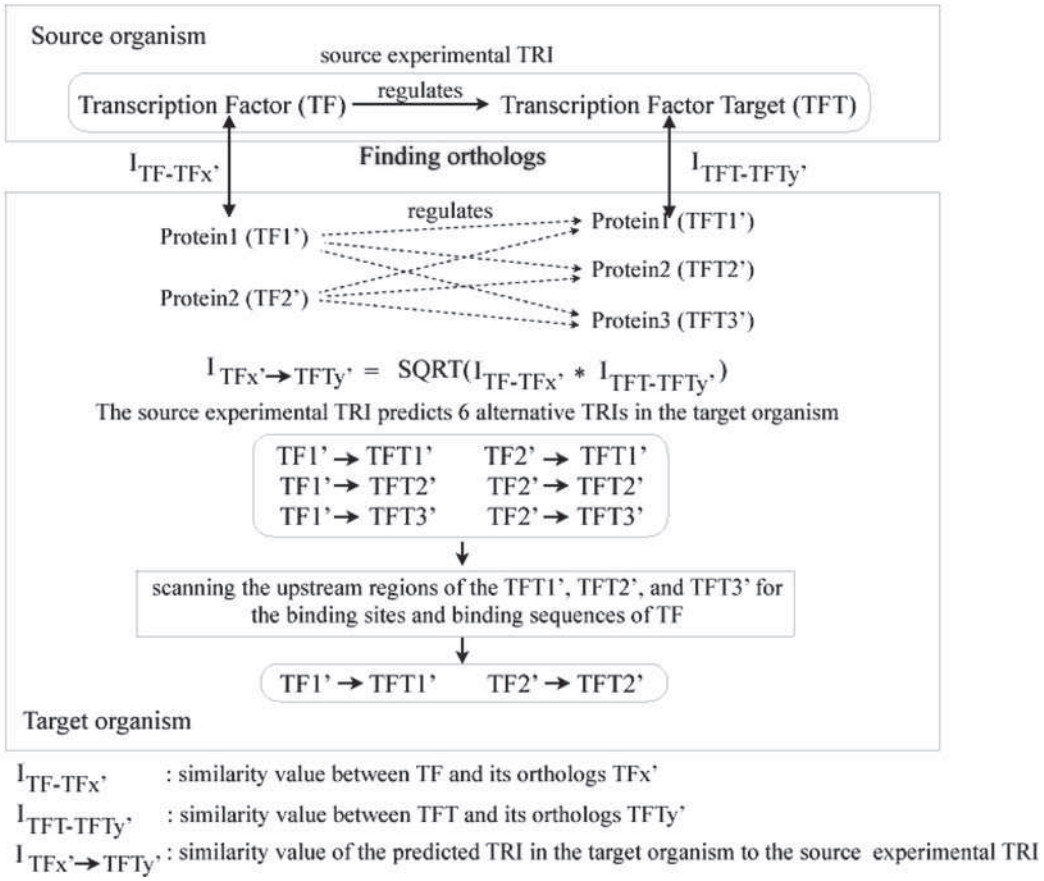

Fig. 6.1. Homology-based transcriptional regulatory interaction prediction.

by filtering out false-positive predictions using other data sources such as gene expression in differing cell cycle stages, protein localization, and membership in protein families. We have used these methods to predict regulogs for all 54 organisms in the Bioverse *(3)*.

The major procedures involved in the prediction and integration of regulatory protein–DNA and protein–protein interactions include: (1) the preparation of essential data sets, including source experimental TRIs, binding sites and binding sequences of the experimentally verified transcription factors, the upstream regions of genes in target organisms, and name mapping between different identification systems; (2) the preparation of additional data sets, such as protein localization and assignment to protein families, for filtering and improving the accuracy of the predictions; (3) the determination of similarity between two protein sequences; (4) the prediction of the TRIs; and (5) the benchmarking of the prediction (*see* **Fig. 6.2**).

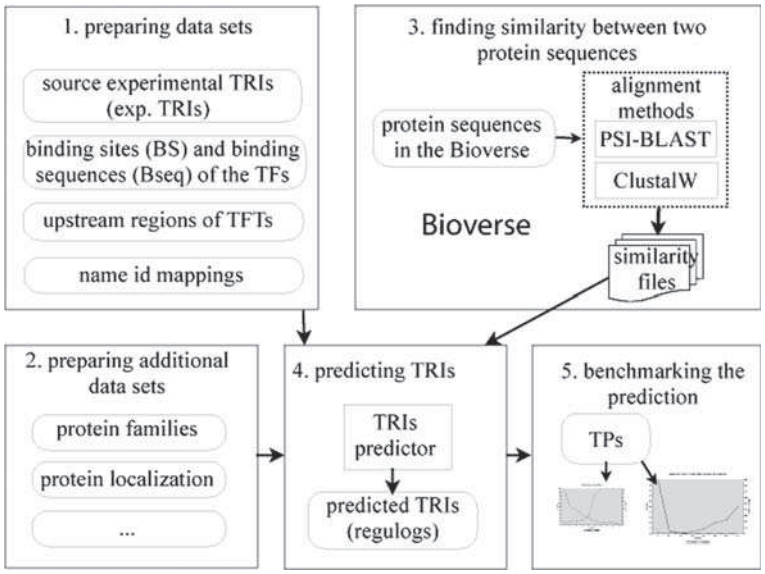

Fig. 6.2. Major steps in the prediction of transcriptional regulatory interactions.

While we present these steps in the context of TRI prediction, the methods and problems of data preparation are common steps in a large number of bioinformatics processes, especially in large-scale systems covering the genomes of multiple organisms. Specifically, the name mapping problem is ubiquitous; it makes all bioinformatics of this type difficult and decreases the coverage and certainty of predictions.

2. Methods

2.1. Preparing Data Sets

As homology-based approaches exploit the transferability of TRIs available in a source organism onto a target organism, the gathering of available interactions from different source organisms is the first essential step. This step is complicated as multiple sources provide different sets of non-comprehensive interactions for specific organisms, with varied data formats. The collection of data from several different sources, however, is essential for expanding the coverage of source TRIs for the prediction and construction of a gold-standard test set for the benchmarking. **Table 6.1** summarizes the sources of our experimental TRIs.

2.1.1. Source Experimental TRIs, Binding Sites, and Binding Sequences

Source experimental TRIs, binding sites, and binding sequences were compiled from two main sources: (1) public databases TRANSFAC® 7.0 *(12)*, SCPD *(36)*, BIND *(37, 38)*, WormBase *(39)* via WormMart *(40)*, RegulonDB *(13)*, and DBTBS *(11)*,

Table 6.1
Sources of experimental TRIs for source organisms

Source experimental TRIs	Source organisms	Description
TRANSFAC® *(1)* (+ binding sites and binding sequences)	*S. cerevisiae*, *H. sapiens*, *M. musculus*, *R. norvegicus*, *D. melanogaster*, *C. elegans*, *A. thaliana*, *O. sativa*	A database of eukaryotic transcription factors, genomic binding sites, and DNA-binding profiles
SCPD *(36)* (+binding sequences)	*S. cerevisiae*	The promoter database of *S. cerevisiae*
BIND (*37, 38)*	*H. sapiens*	The biomolecular interaction network database
WormBase *(39, 40)*	*C. elegans*	A database for genomics and biology of *C. elegans*
RegulonDB *(13)*	*E. coli*	A database of *Escherichia coli* K-12 transcriptional regulatory network, operon organization, and growth conditions
DBTBS *(11)* (+ binding sites and binding sequences)	*B. subtilis*	A database of transcriptional regulation in *B.subtilis*
Supplemental data from literature *(1)*	*S. cerevisiae*	Transcriptional regulatory networks in *S. cerevisiae*

and (2) supplemental data from experimentally determined TRIs described in the literature *(1)*. As methods for gathering and transforming experimental TRIs, binding sites, and binding sequences into a unified format vary among different sources (**Notes 3, 4, 5**), we describe each of them in the following sections. To extend the compiled TRIs for the same organism from different sources, *see* **Note 6**.

2.1.1.1. TRANSFAC®

TRANFAC® *(12)* is a database of transcription factors, their genomic binding sites, and DNA-binding profiles for eukaryotes. This database has two versions: (1) TRANSFAC® Professional, allowing bulk data downloads, and (2) TRANSFAC® Public Database, for online query only. We describe how to compile the experimental TRIs for each eukaryote in the Bioverse from the TRANSFAC® public database.

1. Go to the main searching page found at http://www.gene-regulation.com/cgi-bin/pub/databases/transfac/search.cgi, select "Factor" as the table to search.

2. On the page "searching in table *Factor*," select "Organism Species (OS)" as the table field to search, and specify a specific organism name (e.g., *Saccharomyces cerevisiae*, *Homo sapiens*) as the search item. Set the limit of hits per page to the highest available (100), and then submit the search request. Save the results and edit them to contain only a list of accession numbers (one per line). These accession numbers correspond to transcription factors of the specified species.
3. Use our Python script:run_queryWeb to query detailed information about each transcription factor listed in the results. The Python script programmatically specifies a CGI search for a specific transcription factor, retrieves the search result, and writes it into a new file with a name matching the accession number. Following this, use the script: run_extractInfoFrom TRANSFACTFHtml-Files to extract the transcription factors and their (1) target genes, (2) binding sites, and (3) synonyms, into three separate files. To extract the binding sites to binding sequences, use script: run_extractBindingSeq.

2.1.1.2. SCPD

SCPD *(36)* is a database of promoters found within the *S. cerevisiae* genome. This database contains experimentally mapped transcription factor binding sites and transcription start sites as main entries. We manually compile the experimental TRIs from SCPD, using the following steps:

1. Go to http://rulai.cshl.edu/cgi-bin/SCPD/getfactorlist, click on the link for each transcription factor (e.g., ACE1, ADR1), and the corresponding page will appear.
2. Click on "Get regulated genes" button, and a list of genes regulated by the transcription factor will appear. Click on each of the regulated gene, a new window will appear. Save this window into a local file with the name of the regulated gene. Make this local file under a directory named by the transcription factor.
3. Make a name list of the transcription factors as an input for the script: run_parseSCPDToGetTRIs. Use this script to extract the source experimental TRIs for *S. cerevisiae* from SCPD into a file. Append this file to the source experimental TRIs of *S. cerevisiae* from other sources.

2.1.1.3. BIND

BIND *(37, 38)* is a database of biomolecular interactions, reactions, complexes, and pathway information. This database includes imported experimental TRIs from published research. BIND now becomes a component database of BOND (Biomolecular Object Network Databank). In addition to TRANSFAC®, we compile the experimental TRIs of *H. sapiens* described in *(41–43)* from BIND, using the following steps:

1. Go to BOND (**Note 7**) http://bond.unleashedinformatics.com/Action? and register for a free account. Log in to BOND after getting the account. A BOND search page will appear.
2. Click on the "Identifier search", a new window will appear. Select "PubMed Id" as the identifier from the list box on the left, and input the PubMed identifiers (PMIDs, **Note 8**) of the papers *(41–43)* one at a time into the text input on the right. Then, click on the "Search" button. A search result window will appear.
3. Click on the "Interactions" tab, a new window will appear. On the "Export Results:" list box, select "Cytoscape SIF", a pop-up window for saving the exported result will appear. Save file into a local directory, edit it to have the format ready for use by the system. Append this file to the source experimental TRIs of human from other sources.

2.1.1.4. WormBase

WormBase *(39, 40)* is a database of genomics and biology of *Caenorhabditis elegans* and related nematodes. We compile the experimental TRIs of *C. elegans* from the database using the following steps:

1. Go to http://www.wormbase.org/ and select a tab "Worm Mart" at the top of the page. A martview window will appear.
2. In this window, select the latest release of WormBase (i.e., "WormBase Release WS198") for the "Version:" list box, select "Gene" for the "Dataset:" list box, and click on the "next" button. A window for filtering the queried data set will appear.
3. Under the "Identification" section on this window, check box "[Gene] Species" and select "Caenorhabditis elegans" in the list box, which corresponds to the check box. Also, check box "[Gene] Status," and select "Live" in its corresponding list box.
4. Under the "Annotation" section, check box "Limit to Entries Annotated with:," select "[Function] Trans. Regulator Gene" and "Only" in the corresponding list box, and radio box, respectively. Leave all other boxes as defaults. Click on the "next" button. A new page for formatting the output will appear.
5. Under the "IDs" section, uncheck boxes "Gene WB ID" and "Gene Public Name". Under the "Gene Regulation" section, check boxes "Regulator Gene (Public Name)" and "Regulated Gene (Public Name)." Under the "Select the output format:" section, check radio box "Text, tab separated." Under the "File compression:" section, check the radio box "gzip (.gz)." Under the "Enter a name for this result set:" section, enter a file name for the exported result. Leave all other boxes as defaults. Click on the "export" button. Save the exported file to a local directory.

6. Use the script: run_parseWormBaseToGetTRIs to extract the source experimental TRIs for *C. elegans* into a file. Append this file to the source experimental TRIs of *C. elegans* from other sources.

2.1.1.5. RegulonDB

RegulonDB *(13)* is database of the regulatory network and operon organization of *Escherichia coli* K-12. It is one of the two public databases of prokaryotes from which we compile source experimental TRIs. To compile experimental TRIs from RegulonDB, use the following steps:

1. Go to http://regulondb.ccg.unam.mx/, follow the tab "Downloads" and click on the "Data Sets" item.
2. On the page "Downloadable DataSets," save "File 1. TF – gene interactions" (for experimental TRIs) and "TF binding sites" files (for binding sites and sequences) into a local directory. Edit these two files to have the same format as of the files generated for TRANSFAC.

2.1.1.6. DBTBS

DBTBS *(11)* is a database of transcriptional regulation of *Bacillus subtilis*. It is the other public database of prokaryotes used in this study. This database provides online access, but does not allow bulk download or programmatic search via CGI interface. To get the experimental TRIs of *B. subtilis*, we contacted the authors of *(11)* and asked for the experimental TRIs. The authors kindly gave us the requested data set in XML format. We wrote two scripts: run_extractDBTBSForTFsAndBS and run_extractDBTBSForTRIs that call our Python codes to parse and extract the (1) TRIs, (2) binding sites, and (3) binding sequences from this XML file, and write them to the experimental TRIs, binding sites, and binding sequence files, respectively.

2.1.2. Upstream Regions

Upstream regions of transcription factor target genes were compiled from (1) SGD *(44)* for *S. cerevisiae*, (2) UCSC *(45)* for *H. sapiens (46)*, *Mus musculus (47)*, *Rattus norvegicus (48)*, and *Drosophila melanogaster (49)*, (3) WormBase *(39)* for *C. elegans (50)*, (4) TAIR *(51)* for *Arabidopsis thaliana (52–55)*, (5) TIGR Rice Genome Annotation database *(56)* for *Oryza sativa (57)*, and (6) NCBI for all prokaryotes, including *E. coli, B. subtilis,* etc. As methods for gathering and extracting the upstream regions and transforming them into a unified format vary among sources (**Notes 9, 10**), we describe these methods as follows.

2.1.2.1. SGD for S. cerevisiae

1. Go to http://www.yeastgenome.org/ *(44)*.
2. On the left side of this main page, in section "Download Data," select "FTP." A list of a directory in a new page will appear. From here, go into the "sequence" directory, then "genomic_sequence" directory, and then the "orf_dna"

directory. Copy the file orf_genomic_1000_all.FASTA.gz into a local directory. This file contains ORF sequences with introns, untranslated regions 1,000 bp upstream of the initial ATG and 1,000 bp downstream of the stop codon.

3. Use our code script: run_extractUpstreamRegions_1000 bp_saccharomyces_cerevisiae to parse and extract the saved file into a mapping file between the ORFs and their corresponding upstream regions.

See (**Note 11**) for an alternative way to get the upstream regions for *S. cerevisiae*.

2.1.2.2. UCSC Genome Browser for H. sapiens, M. musculus, R. norvegicus, and D. melanogaster

1. Go to http://hgdownload.cse.ucsc.edu/downloads.html *(45)*.
2. In the box "Sequence and Annotation Downloads," search for a specific organism. Select "Human," for instance. This jumps to the "Human Genome" box. In the box "Human Genome," select "Full data set," which leads to a directory page containing a list of finished human genome assemblies with their descriptions.
3. Click on the upstream<xxx>.zip to download the files. These files are zipped and are in FASTA format, with each upstream sequence associated with an identifier system that is specific to an organism (i.e., NM_xxxx RefSeq in case of human, mouse, and rat, and FlyBase symbol in case of fly). The detailed descriptions of these upstream files are described on the same page. Basically, xxx in the name of an upstream region file stands for 1,000, 2,000, and 5,000 to represent the number of bases of each upstream region in each file (**Note 12**).
4. After downloading these files, use script: run_extract_NP_NM_homo_sapiens to parse and extract the mapping between NCBI GenBank identifiers (GIs) of human proteins to their corresponding RefSeq identifiers and use run_extract UpstreamRegions_<xxx>bp_homo_sapiens to generate a mapping file from GIs to upstream regions ready for use by the system.
5. Repeat Steps 1–4 for "Mouse" and "Rat."
6. Use script: run_extractUpstreamRegions_<xxx>bp_drosophila_melanogaster to extract the upstream region file of *Drosophila* into a mapping file between FlyBase symbols and their corresponding upstream regions.

See **Note 13** for an alternative way to get the upstream regions for *H. sapiens*.

2.1.2.3. WormBase for C. elegans

1. Go to http://www.wormbase.org/db/searches/advanced/dumper *(39)*.
2. In the "1. Input Options" box, type in "I II III IV V X XX XO." These correspond to the chromosomes of *C. elegans*. In

the "2. Select one feature to retrieve," click on "5 UTRs." In the "3. Output options," check box "flanking sequences only," specify the flanking sequence lengths (i.e., 1,000 bp 5′ flank, 0 bp 3′ flank), leave the coordinates relative to "Chromosome," select the sequence orientation as "Always on canonical strand," select the output format "Save to disk (Plain TEXT)," then click "DUMP" button. The saved file is in FASTA format in which each upstream region is associated with a sequence name (gene model) and genetic nomenclature for *C. elegans*.

3. Use script: run_extractUpstreamRegions_1000 bp_caenorhabditis_elegans to parse, extract, and transform the saved file into a mapping file from GIs to upstream regions ready for use by the system.

2.1.2.4. TAIR for A. thaliana

1. Go to ftp://ftp.arabidopsis.org/home/tair/Sequences/blast_datasets/ *(51)*.
2. Save files TAIR_upstream_xxx_yyyymmdd, where xxx represents the number of base pairs, and yyyymmdd represents the date the files are generated. These files are in FASTA format, with each upstream sequence associated with an *Arabidopsis* Genome Initiative locus identifier (AGI ID) (e.g., At1g01120).
3. Use script: run_extractUpstreamRegions_1000 bp_arabidopsis_thaliana to parse, extract, and transform the saved files into a mapping file from AGI IDs to their corresponding upstream regions.

2.1.2.5. TIGR for O. sativa

1. Go to ftp://ftp.tigr.org/pub/data/Eukaryotic_Projects/o_sativa/annotation_dbs/pseudomolecules/ *(56)*, select the directory "version_x.x," where x.x is the latest (i.e., version_5.0, for the current release), and then select the directory "all_chrs." Under this directory, save file "all.1kUpstream" into a local directory (**Note 14**).
2. Use our script: run_extractUpstreamRegions_1000 bp_TIGRRice to parse, extract, and transform the saved file into a mapping file from TIGR_LOCUS IDs (e.g. LOC_Os01g01030.1) to their corresponding upstream regions.

2.1.2.6. NCBI for Prokaryotes

1. For *E. coli* K-12, go to ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria, and enter the "Escherichia_coli_K12" directory.
2. Save files "xxx.fna" and "xxx.ptt" into <organism>_Genome.fna and <organism>_ProteinMap.ptt, respectively, where file with fna extension contains complete genome sequence, and file with ptt extension contains locations, the start and stop positions, for each gene on the genome sequence. For genomes that

have only sequences for chromosomes such as *Plasmodium falciparum*, save "xxx.fna" and "xxx.ptt" into <organism>_chr<n>.fna and <organism>_chr<n>_ProteinMap.ptt, respectively.

3. Repeat Steps 1 and 2 for all bacteria and other prokaryotes.
4. Use script: run_extractGIsToUpstreamRegionsFromGenomeSeqAnd ProteinMap to parse, extract, and transform the saved files into a mapping file from GIs to upstream regions for the organisms.

2.1.3. Name Mapping

Name mapping is another essential part of data preparation. Data sets such as transcription factors and transcription factor targets in the experimental TRIs and the upstream regions of gene sequences from several sources are associated with their own identifiers (e.g., common names of TFs and TFTs in TRANSFAC®, ORFs from SGD, GenBank Identifiers at NCBI (GIs), gene IDs from Entrez Gene, WormBase IDs, FlyBase symbols, and AGI IDs for worm, fly, and *Arabidopsis*, and Refseq for upstream regions). Therefore, we map these identifiers to protein identifiers in the Bioverse (**Note 15**) for the prediction of protein–DNA interactions and the integration of protein–DNA and protein–protein interaction networks. In the following text we describe various name mappings required by the system.

2.1.3.1. Name Mapping from TFs, TFTs to Protein IDs in the Bioverse

To integrate the regulatory and protein–protein interactions, the TFs and TFTs from source experimental TRIs described in **Section 2.1.1** map to protein IDs in the Bioverse. While the Bioverse provides an ID-mapping file consisting of different ID systems (i.e., GIs from NCBI, ORFs from SGD, AGI IDs from TAIR) to protein IDs in the Bioverse, what we mainly have for the TFs and TFTs from source experimental TRIs are their common names. Hence, we establish an intermediate mapping that links these common names to protein IDs in the Bioverse (**Notes 16, 17**). The building process of an intermediate mapping file varies according to the ID system that will be used as the intermediate. We describe how to handle name mapping from the common names of TFs and TFTs in source experimental TRIs to protein IDs in the Bioverse, according to the formats for respective organisms.

Saccharomyces cerevisiae

1. Go to http://www.yeastgenome.org/gene_list.shtml *(44)*.
2. Save file SGD_features.tab into a local directory.
3. Use script: run_extractNameMappingFromSGD to extract SGD_features.tab into a mapping file between the systematic ORF names and their common names.

Prokaryotes and Other Eukaryotes

1. Go to http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene, on the left side, click on "Downloads (FTP)." A new page of directories will appear. Go into directory "DATA" and save the files gene2refseq.gz and gene_info.gz into a local directory.
2. Use scripts:run_extract_gi_to_gene_id and run_extract_gene _id_to_names to extract the gene2refseq and gene_info, respectively, and then use run_buildGIToNames to generate a mapping file from GIs to names for the organisms listed as an input of the script. For eukaryotes, we can improve the GI to name mapping with additional synonyms extracted from TRANSFAC®.

Homo sapiens

Human genes do not have well-defined gene names as do genes in organisms such as yeast (**Note 18**). As the ID-mapping file for human in the Bioverse largely contains GI records, we decided to use GIs as intermediate ID mapping from a common name to a protein ID in the Bioverse. The original mapping file from GIs to common names is generated from gene2refseq and gene_info in Entrez Gene as described above. To refine the name mapping file, we combine synonyms (aliases) from additional sources such as TRANSFAC, HUGO *(58)*, and OMIM *(59)*, using methods listed as follows (**Note 19**).

- *Method to compile and extract synonyms from TRANSFAC®*

The synonyms of transcription factors are a part of the source TRIs compiled from TRANSFAC®. Hence, we do not need a separate compilation. As the script: run_extractInfoFromTRANSFACTF HtmlFiles_homo_sapiens also extracts the synonyms for each human transcription factor, we only need to combine the resulting file to the original GI-to-name mapping file from Entrez Gene using the script: run_addSynonyms.

- *Method to compile and extract synonyms from HUGO Gene Nomenclature Committee (HGNC)*

We compile and extract synonyms from HGNC using the following steps:

  - Go to http://www.genenames.org/ and click on the "Downloads" button at the top. The new page of database downloads will appear. Click on "Custom Download" listed in a box at the top of the page. A "Custom Downloads" page will appear.
  - Check boxes: "Approved Symbol," "Approved Name," "Previous Symbols," "Previous Names," and "Aliases." Check boxes "Approved" for the select status, and "Select all Chromosomes." Scroll down and select the "ORDER BY" to change to the "Approved Symbol," and the

"Output format" to be "Text." Then, click the "submit" button. The result of the customized query will pop up in a new window. Save the result into a local file.

- Use script: run_extracHumanGeneSynonymsFrom-HUGO. This script extracts approved symbols, approved names, previous symbols, previous names, and aliases from all approved human genes into a file that will be used by script: run_addSynonyms to combine these names into the human GI-to-name mapping file.

- *Method to compile and extract synonyms from OMIM*
  - Go to http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM. On the left side, under the FAQ section, click "Download." A new OMIM FAQs page will appear. Under "Downloading OMIM" section in item 1, click on "omim.txt.Z" to download the complete text of OMIM and save it into a local directory.
  - Use script: run_extracHumanGeneSynonysFromOMIM to extract the gene_ symbols and their synonyms from file omim.txt and write these extracted names into an output file. The script: run_addSynonyms combines these names into the human GI-to- name mapping file.

Caenorhabditis elegans

To extend the mapping from GIs to names for *C. elegans* generated by run_buildGIToNames, we compile gene aliases of *C. elegans* from WormBase via WormMart using the following steps:

1. Go to http://www.wormbase.org/ and select a tab "WormMart" at the top of the page. A martview window will appear.
2. In this window, select the latest release of WormBase (i.e., "WormBase Release WS198") for the "Version:" list box, select "Gene" for the "Dataset:" list box, and click on the "next" button. A window for filtering the queried data set will appear.
3. Under the "Identification" section on this window, check box "[Gene] Species" and select "Caenorhabditis elegans" in the list box, which corresponds to the check box. Also, check box "[Gene] Status," and select "Live" in its corresponding list box. Leave all other boxes as defaults. Click on the "next" button. A new page for formatting the output will appear.
4. Under the "IDs" section, check box "Gene Names (merged)." Under the "Proteins" section, check box "NCBI Protein GI." Under the "Select the output format:" section, check radio box "Text, tab separated." Under the "File compression:" section, check the radio box "gzip (.gz)." Under the "Enter a name for this result set:" section, enter a file name for the exported result. Leave all other boxes as defaults. Click on the "export" button. Save the exported file to a local directory.

5. Use the script: run_addSynonymsForC_elegans to append these aliases into the available name mapping file for *C. elegans*.

Arabidopsis Thaliana

1. Go to ftp://ftp.arabidopsis.org/home/tair/Genes/.
2. Save file gene_aliases.20080716 into a local directory. This file contains the mapping from AGI IDs to gene aliases in the format "AGI ID name1 name2" which is ready for use by the system.

Oryza Sativa

The ID name mapping file provided by the Bioverse for *O. sativa* does not include any intermediate ID that could be linked to the common names of transcription factors and their targets in the source experimental TRIs of rice, so we use the following steps to build a mapping file from the common names of rice proteins to the protein IDs in the Bioverse.

1. Go to ftp://ftp.tigr.org/pub/data/Eukaryotic_Projects/o_sativa/annotation_dbs/pseudomolecules/ *(56)*, select the directory "version_x.x," where x.x is the latest (i.e., version_5.0, for the current release), and then select the directory "all_chrs." Under this directory, save file "all.pep" into a local directory.
2. Perform BLASTP from rice protein sequences retrieved from the Bioverse to protein sequences in all.pep using script: run_blastp_bioverse_<rice species>_to_TIGR-rice, where rice species could be "oryza_sativa_japonica_fl," "oryza_sativa_japonica_syngenta," and "oryza_sativa_indica_9311."
3. Use scripts: run_extractBLASTPSimilarity and run_extract TIGRLOCUSTo BioverseId to extract the BLASTP result and then transform them into a mapping file from TIGR_LOCUS IDs to Bioverse IDs.
4. Use script: run_buildTIGRRiceCommonNamesToBid to extract all.pep file into a mapping file from TIGR_LOCUS to names.

2.1.3.2. Name Mapping of an ID System Associated with the Upstream Sequences from Different Sources to Protein IDs in the Bioverse

In this section, we describe how we build a mapping from a specific ID system associated with the upstream sequences of a specific organism to their corresponding protein IDs in the Bioverse.

Saccahromyces cerevisiae

The upstream regions of *S. cerevisiae* are annotated with the same sets of ORFs from SGD. Hence, we do not have the problem of mapping from these ORFs to protein IDs in the Bioverse.

Homo sapiens, Mus musculus, and Rattus norvegicus

The upstream regions of human, mouse, and rat compiled from UCSC Genome Browser *(45)* are annotated with NM_xxxx, which are the RefSeq accession numbers for nucleotide sequences. However, the name mapping from common names to protein IDs in the Bioverse is via NCBI GenBank Identifiers (GIs). Hence, these RefSeq numbers are not directly usable by the system. To handle this mapping issue, we use the following steps to transform the RefSeq accession numbers to protein GIs.

1. Go to ftp://ftp.ncbi.nih.gov/refseq/H_sapiens/mRNA_Prot/ and save file human.protein.gpff.gz into a local directory. This file contains GenBank records of NP_xxxx, which are the RefSeq accession numbers for protein sequences in human that are associated with GIs and NM_xxxx.
2. Extract the mapping between NP_xxxx, NM_xxxx RefSeq numbers and protein GIs using our script: run_extract_NP_NM_homo_sapiens. This code will result in a mapping file of GIs, NP_xxxx and NM_xxxx. The extracted mapping will be used as an input of script: run_extractUpstream Regions_1000bp_homo_sapiens for extracting the upstream regions mentioned in **Section 2.1.2** for human.
3. Repeat Steps 1 and 2 for mouse and rat by accessing: ftp://ftp.ncbi.nih.gov/refseq/M_musculus/mRNA_Prot, and ftp://ftp.ncbi.nih.gov/refseq/R_norvegicus/mRNA_Prot, respectively.

Oryza sativa

The compiled upstream sequences of *O. sativa* (cultivar Nipponbare of *Oryza sativa* L. ssp. japonica) are associated with TIGR_LOCUS IDs. As we have built a mapping file from TIGR_LOCUS IDs to protein IDs in the Bioverse, we do not encounter the mapping problem for this genome.

Arabidopsis thaliana

The compiled upstream sequences of *A. thaliana* are associated with AGI IDs that are also available in the ID-mapping file provided by the Bioverse, so we do not encounter the mapping problem for this genome.

Drosophila melanogaster

While we compiled the upstream region files of *D. melanogaster* from UCSC as of human, mouse, and rat, the upstream sequences of the fly are not associated with RefSeq numbers. Instead, they are associated with FlyBase symbols that are also available in the ID-mapping file provided by the Bioverse. So, in case of fly, to extract the upstream regions, we use a script similar to the script of extracting the upstream region from FASTA format for yeast and *Arabidopsis*.

Prokaryotes

As we compile and extract the upstream regions of prokaryotes from NCBI, where genes and proteins are already associated with nucleotide and protein GIs, we do not have a mapping problem from an ID system associated with the upstream sequences to protein IDs in the Bioverse.

2.2. Preparing Additional Data Sets

This section describes the preparation of additional data sets utilized for improving the accuracy of TRI predictions.

2.2.1. Preparing Protein Localization

Protein localization is employed as a filter for improving the accuracy of the predicted TRIs. It strongly correlates with mRNA co-expression, as well as physical and functional interactions *(60, 61)*. We compile protein localization data for *S. cerevisiae* from the TRIPLES database *(62, 63)* and Yeast GFP Fusion Localization database *(60)* (**Notes 20, 21**). To retrieve the protein localization data from these databases, use the following steps.

2.2.1.1. TRIPLES

1. Go to ftp://ygac.med.yale.edu/ygac_pub_ftp/.
2. Save the file localization_pub_data_9_4_01.tab into a local directory and use script: run_extractPLOCFromTRIPLES to extract the ORFs and their localizations into a file ready for use by the system.

2.2.1.2. Yeast GFP Function Localization Database

1. Go to http://yeastgfp.ucsf.edu/.
2. Within the banner at the top of the page, click on "Go" for the advanced query. A new page will appear. In this page, leave "Search Criteria" as default, where the inputs of all search criteria including of the "Subcellular Localization" will be wildcards (*). For the "Display Options," check the box "Download the selected dataset as a tab-delimited file" and box "include localization table." Press the "submit" button. The system will write the query result into a file "downloadxxxxxxxx.txt" and put it in the "Search Results" section.
3. Save the result file into a local directory and use script: run_extract PLOCFromYeastGFP to extract the ORFs and their localizations into a file and use script: run_combinePLOCs to combine the results from both TRIPLES and Yeast GFP into a single file ready for use by the system.

2.2.2. Preparing Protein Families

The protein family is considered as another filter for improving the accuracy of the predicted TRIs. We hypothesize that a predicted transcription factor should share protein domains with its source transcription factor. At present, all protein families are compiled from TRANFAC *(12)* (**Note 22**), using the following steps:

1. Go to http://www.gene-regulation.com/cgi-bin/pub/databases/transfac/search.cgi?.

2. On this page, click on the "Class" button. The new page for searching the Class table will appear. Input a wildcard (*) in the "Search term" text field. Select "Class (CL)" as the field to search in the table, and "100" as number of hits per page. Then, click the "Submit" button. The new page of protein classes will appear.
3. Save this page into the local directory as a mapping file between class accession numbers and their descriptions. Then, click on each accession number in this page to save as a file on a local directory. These saved files will be used by the prediction method for filtering the predicted TRIs. A predicted TRI will be filtered out if its TF has no sharing of any protein families with the source TF (**Notes 23, 24**).

2.3. Finding Similarity Among Protein Sequences

As we use homology-based approaches for TRI prediction, the determination of similarity among protein sequences is an essential step. In the following section, we describe the preparation of protein sequences and the use of alignment methods for finding sequence similarity.

2.3.1. Preparing Protein Sequences

We compile protein sequences for a specific organism from the Bioverse *(3)*, using the following steps:

1. Prepare a text file that lists the names of the organisms (one per line), which will be queried for all protein sequences.
2. Use script: run_getMoleculeSeqsViaRPC to retrieve the protein sequences for the organisms listed in the prepared text file in Step 1 from the Bioverse via XML RPC server (see Chapter 22) (**Note 25**).

2.3.2. Finding Similarity Among Protein Sequences

Similarities between protein sequences can be determined using several alternative alignment methods. We summarize each method and discuss their effect on the results.

2.3.2.1. Alignment Methods

1. **BL2SEQ** *(64)* is a BLAST-based tool for aligning two protein or nucleotide sequences that are presumably known to be homologous. It utilizes the BLAST (**Note 26**) engine *(65)* for local alignment. The main purpose of BL2SEQ is to compare the similarity between two sequences and reduce the processing time of using the standard BLAST program; BL2SEQ is the fastest, but least sensitive, method compared with the other alignment methods described here.
2. **PSI-BLAST (Note 27)** (Position-specific iterative BLAST) *(66)* is a feature of BLAST 2.0. It improves the sensitivity of protein–protein BLAST (BLASTP) using a position-specific scoring matrix (PSSM) constructed from a multiple sequence alignment of the best hits in each of the most recent iteration, such that it refines the PSSM over sequential iterations. Each

position in the PSSM will contain varied scores according to the conservation of the position. A position that is highly conserved will get a higher score. PSI-BLAST is much more sensitive than the standard BLAST program in capturing the distant evolutionary relationships or weak relationships between protein sequences.

3. **SSearch** implements the Smith-Waterman algorithm *(67, 68)* for local sequence alignment. Its main purpose is to avoid misalignment due to high noise levels in the low similarity regions of the distantly related sequences. Hence, it ignores all these regions and focuses only on regions that have highly conserved signals with positive scores. The Smith-Waterman algorithm guarantees optimal local alignment with the trade-off of moderately demanding computing resources. Hence, it is too slow for searching a large genomic database such as GenBank.
4. **ClustalW** *(69)* is a progressive *global multiple alignment* method that improves the sensitivity of highly divergent sequence alignment. It incorporates (1) an individual weight for each sequence, (2) varied amino acid substitution matrices at different stages of the alignment, (3) residue-specific gap penalties, and (4) position-specific gap penalties. ClustalW consists of three main steps: (1) performing pairwise alignments for all pairs of sequences in order to generate the distance matrix, (2) building a guide tree from the calculated distance matrix, and (3) carrying out a multiple alignment guided by the tree.

In our benchmarking process, we search for similar protein sequences of a source transcription factor (TF) or a source transcription factor target (TFT) in a target organism using PSI-BLAST. Then, we use ClustalW to create multiple alignments of the source protein sequence and the similar protein sequences found with PSI-BLAST.

2.3.2.2. Similarity Assessment

BLASTP and PSI-BLAST assess the similarity between query and protein sequences in a database by creating a bit score, an E value (expectation value), and match types with identities, positives, and gaps. The "bit score" is the normalized raw score (**Note 28**) according to the statistical variables defined in the scoring system. This score allows the comparison between different alignments with different scoring matrices. The "E value" is the probability that the similarity found in this alignment might happen by chance, with a lower E value corresponding to a more significant score. The "identity" is the ratio of the number of identical residues over the total number of aligned residues and gaps between a query and a target protein sequence. The "positive" is the ratio of the number of identical plus the non-identical but

conserved residues (represented by a minus sign in the alignment section of the blast result) over the total number of aligned residues and gaps between a query and a target protein sequence. The "gap" is the number of gaps (represented by a dash symbol), either in the query or in the target protein sequence, over the total number of aligned residues and gaps between a query and a target protein sequence.

SSearch assesses the similarity between two sequences via the Z-score, Smith-Waterman score, E() value, percentage identity, and percentage similarity. The Smith-Waterman score is calculated from a scoring matrix that includes the match and mismatch scores, a gap creation penalty, and a gap extension penalty. The Z-score is a normalized score calculated from a linear regression performed on the natural log of the sequence length of the search set. SSearch uses the distribution of Z-scores to estimate the expected number of sequences (represented by E() value) produced by chance with equal or greater Z-score than that attained from the search. The greater the Z-score, the lower the E() value. The percentage identity and percentage similarity represent the number of identical (represented by two vertical dots in the alignment section of the SSearch result) and the number of conserved but not identical (represented by a single dot) residues over the number of overlapping amino acids, respectively.

While higher scores and lower E values imply a better hit (such that two sequences are significantly similar), these values are calculated based on local alignment. Likewise, the percentage identity and percentage similarity are calculated only from aligned segments. Hence, in the case of two sequences with distant evolutionary relationships or weak relationships, these values are not directly representative of the similarity between them. Therefore, in our benchmarking, we assess the similarity between two sequences using the following steps:

1. For each protein (either TF or TFT) in the test set, we use PSI-BLAST to (i) find its top hits from all other proteins in the test set, and (ii) find its top hits from proteins in the Uniprot (again, we need to handle name mapping from the Uniprot protein identifier to the protein ID in the Bioverse), and then (iii) put these two sets of protein sequences into a file (including the query protein sequence). The number of files will be equal to the number of proteins in the test set (all distinct TFs and TFTs).
2. We use ClustalW to create multiple alignments for the set of sequences in each file. Based on the global multiple alignment results, we assess the similarity between the query protein sequences to every other hit sequence in the resulting file, as the number of identical residues over the total number of residues of the hit sequence. We call this ratio the *fraction*

identity (FI). Identical protein sequences will have FI = 1.0, while the FI between two proteins with no similar sequences is equal to 0.0.

2.4. Predicting TRIs

To predict TRIs, we developed a Python script following the homology-based approach described in the introduction. This code implements the following steps (as shown in **Fig. 6.3**).

1. For each source experimental TRI compiled in **Section 2.1**, find the orthologous proteins TFx' and TFTy' of TF and TFT in a target organism, from the similarity values (i.e., $I_{TF\text{-}TFx'}$ and $I_{TFT\text{-}TFTy'}$ in **Fig. 6.3**) generated in **Section 2.3**. The numbers of orthologous proteins are limited by the cutoffs of similarity values (i.e., E-values, bit score, Z-scores, percentage identity, percentage similarity; with each varied according to the alignment methods). The predicted TRIs stem from all combinations of homologous TFx' and TFTy' in the target organism. Each is assigned a similarity of interaction $I_{TFx'\rightarrow TFTy'}$, where

$$I_{TFx'\rightarrow TFTy'} = \mathrm{sqrt}(I_{TF-TFx'} * I_{TFT-TFTy'})$$

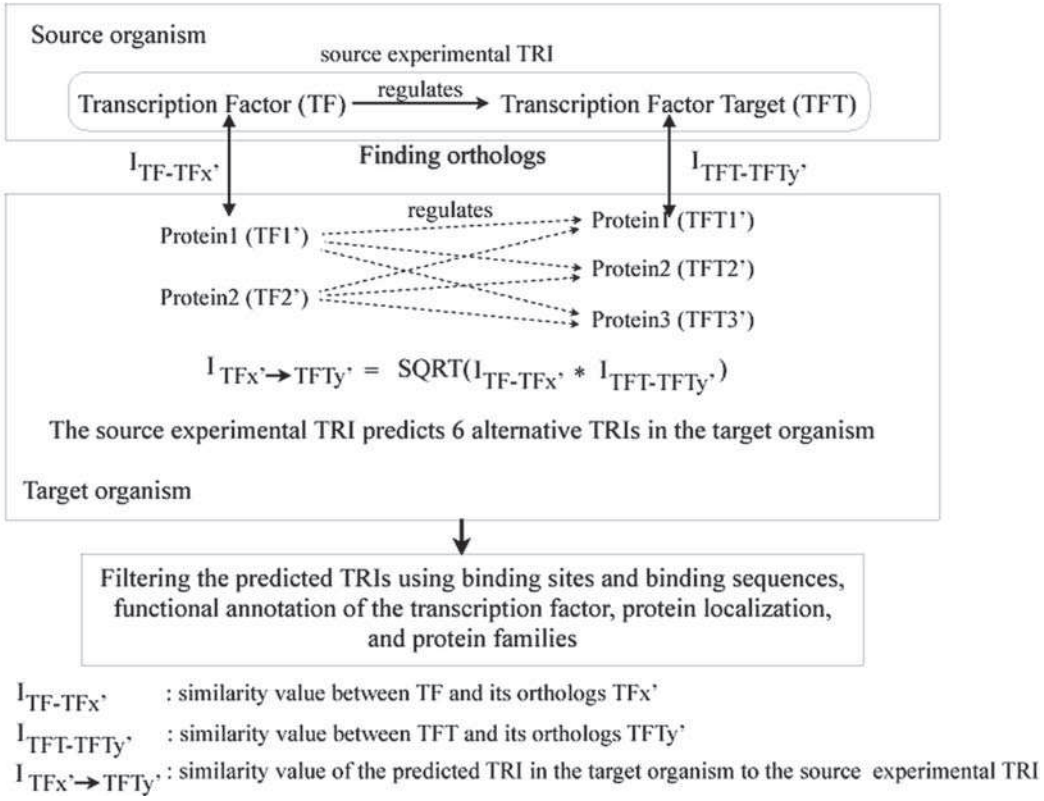

Fig. 6.3. Predicting TRIs.

The following steps are optionally used for improving the accuracy of TRI prediction.

1. Filter out the predicted TRIs for which functional annotation does not include "transcription factor." Note that the annotation of a predicted TF (TF_x') is queried from XML RPC server in the Bioverse.
2. Filter out the predicted TRIs for which an upstream region (prepared in **Section 2.1.2**) of their TFTs is not found with any binding sites and binding sequences (prepared in **Section 2.1.1**) of the TF of source experimental TRI (prepared in **Section 2.1.1**).
3. Filter out the predicted TRIs for which the TF and TFT do not share any protein localization (prepare in **Section 2.2.1**).
4. Filter out the predicted TRIs for which the protein families (prepared in **Section 2.2.2**) of its TF and of the TF of the source experimental TRI do not overlap.

2.5. Benchmarking

One of the important issues for a prediction method is its accuracy and coverage. In this section, we describe a design experiment for measuring accuracy and coverage for this prediction method.

2.5.1. TP and Test Set

Accuracy and coverage are defined as follows.

$$\textbf{Accuracy}_x = A * 100/(A + B)$$

$$\textbf{Coverage}_x = A * 100/|TP|$$

for which:

x = a similarity value cutoff of E-values, Z-scores, percentage identity, or fraction identity used for discarding the predicted TRIs.

A = the number of predicted TRIs in the true positive set (TP) at cutoff x.

B = the number of predicted TRIs not in TP at cutoff x.

|TP| = the number of source experimental TRIs in TP.

The true positive set (TP) of a target organism contains all experimental TRIs of the organism. If TRIs in the true positive set contain **N** distinct transcription factors (TFs) and **M** distinct transcription factor targets (TFTs), the test set will contain **N x M** TRIs, which result from all-against-all combinations between TFs and TFTs in the TP (**Note 29**). We define accuracy as the fraction of TRIs predicted by the system that are in TP out of all the predicted TRIs (either in TP or not in TP) at a specific cutoff. We define the coverage as the fraction of TRIs predicted by the system that are in TP at a specific cutoff over all TRIs in TP. Higher FI threshold cutoffs correspond to higher accuracy and lower coverage.

In general, if the system is optimized, the cross-point between the accuracy and the coverage plots represents an appropriate cutoff for the system to include or exclude predicted TRIs.

2.5.2. Measuring Accuracy and Coverage

To benchmark the predicted TRIs, we measure the accuracy and coverage of the predicted TRIs at specific cutoffs, we use the script: run_regulogBenchmarking_<organism>_sprot, where organism could be human, mouse, rat, yeast, and fly. This script calls our Python code that implements the following steps:

1. For each TRI in the test set, find the source experimental TRIs that give the TRI from the test set with the highest geometric mean of the FI product (**Note 30**). Assign this source experimental TRI and the geometric mean of the FI product to the TRI from the test set.
2. For each FI threshold cutoff ranging from 0.0 to 0.95, count the number of TRIs in the test set that are in TP and not in TP, and the FIs between the source TF and target TF and the source TFT and target TFTs at or above the cutoff.
3. Calculate the accuracy and coverage for each cutoff using the results from Step 2 and write the results into output files.

We use the above code to benchmark the accuracy and coverage of predicted TRIs for five organisms: human, mouse, rat, fly, and yeast. **Table 6.2** shows the numbers of pairs in the TP, TFs in TP, TFTs in TP, and TRIs in the test set of the five organisms. The source experimental TRIs came from the combination of all TRIs in TP from these organisms plus the experimental TRIs of *E. coli* and *B. subtilis* that had protein IDs in the Bioverse. For the benchmarking, we exclude TRIs in TP of the target organism from the source experimental TRIs. To obtain numbers for **Table 6.2** and generate the test

Table 6.2
Numbers of TP, TFs in TP, TFTs in TP, and TRIs in the test set of human, mouse, rat, fly, and yeast used for benchmarking

Organisms	TP	TFs in TP	TFTs in TP	TRIs in the test set (TFs in TP × TFTs in TP)
S. cerevisiae	136	38	104	3,952
D. melanogaster	171	72	65	4,680
M. musculus	309	109	159	17,331
R. norvegicus	46	18	36	648
H. sapiens	900	118	553	65,254

sets for the benchmarking, we use the script: run_generate-TestsetForBenchmarking. **Figure 6.4** shows the accuracy and coverage without any filtering for the test sets of human, mouse, rat, fly, and yeast. In general, the cross-point between the accuracy and coverage lines could be an appropriate cutoff. In this figure, the cross-points of the plots vary according to the available TRIs in TP of the target organisms.

To measure the errors of the method, we generate new test sets comprising randomly selected sets of 80% of the TRIs in TP. We repeat this benchmarking process 50 times. **Figure 6.5** shows

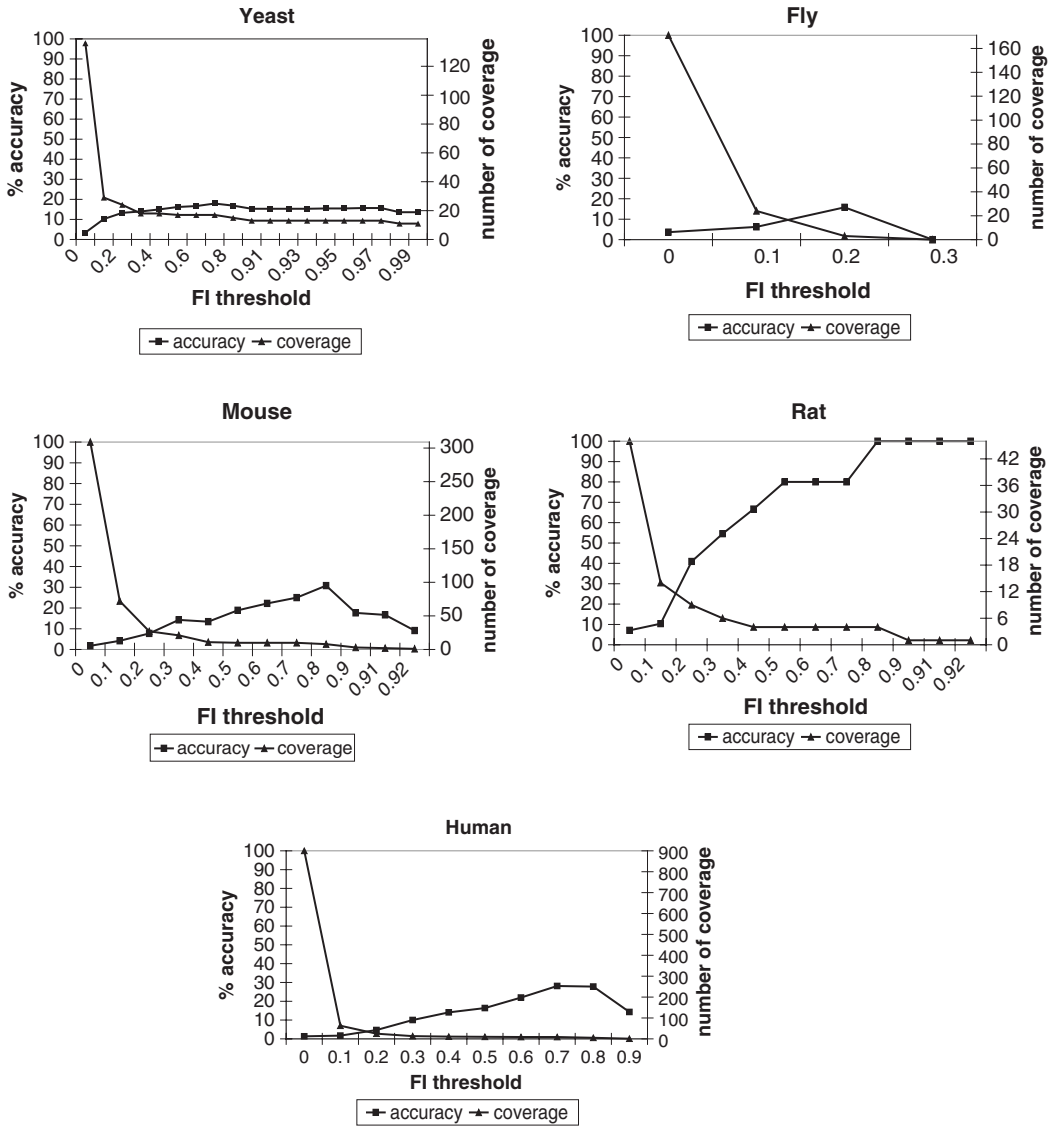

Fig. 6.4. Accuracy and coverage without any filtering for the test sets of yeast, fly, mouse, rat, and human.

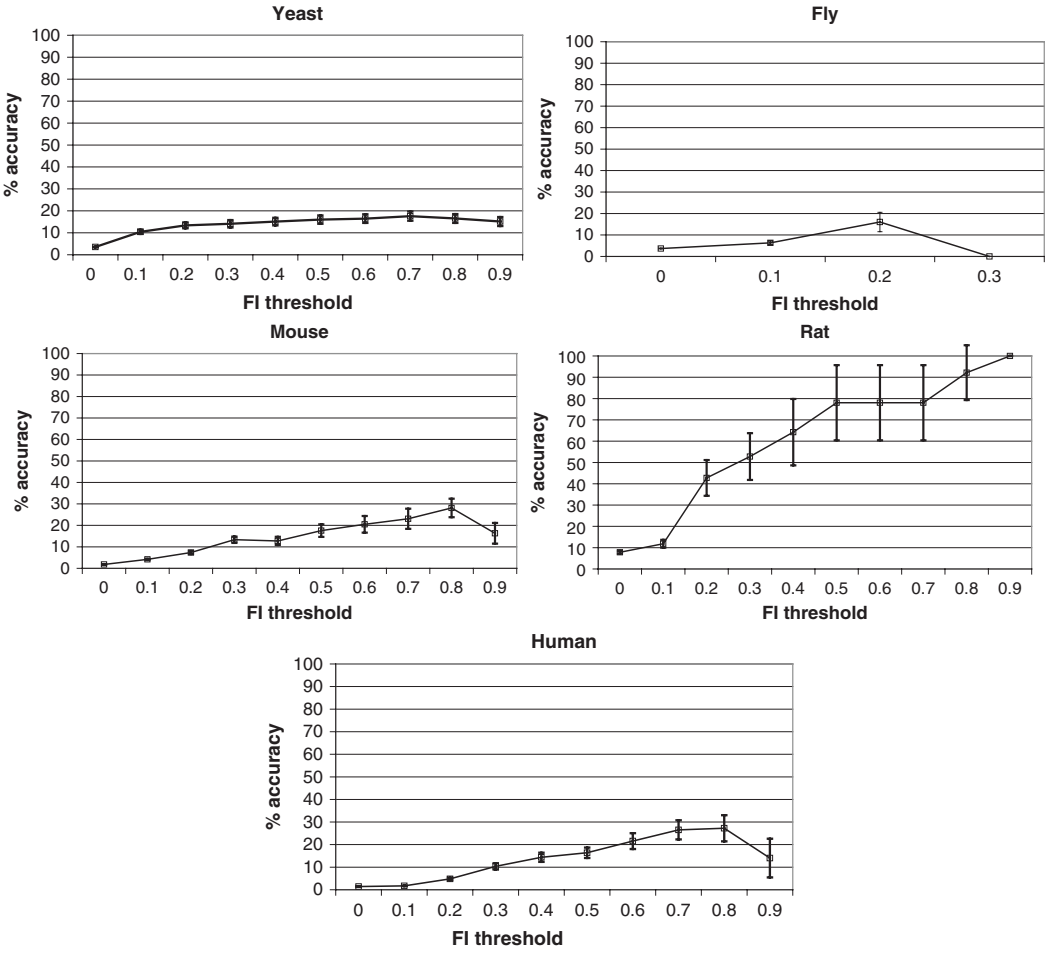

Fig. 6.5. Accuracy with error bars of the TRI prediction method without any filtering for the test sets of yeast, fly, mouse, rat, and human. *Data point* represents the mean of 50 bootstrapped data sets (randomly selected 80% of TRIs in TP) and *error bars* indicate the standard deviations above and below the mean.

the mean accuracy without any filters and the standard deviations below and above the mean at each FI threshold. Results from human, mouse, and yeast do not vary substantially among 50 tries. The method does not work well with fly data sets due to a low number of significant homologs within the available source experimental TRIs. It is likely that this also explains the results from fly in **Fig. 6.4**. The high standard deviations in the case of rat indicate heterogeneity for the rat TP. As we do not have a complete set of TRIs in TP for any target organism, the accuracy and coverage of predictions can only be evaluated as minimum accuracy and coverage.

2.5.3. Correctness of Benchmarking Method

We perform a sanity check to measure the correctness of the benchmarking method by including TRIs in the TP of the target organism as part of the source experimental TRIs and

calculate the accuracy and coverage using high threshold cutoffs ranging from 0.95 to 1.0. If the method is correct, then the predicted TRIs in TP should be the same as their source experimental TRIs and both accuracy and coverage will be 100%.

In the case of mouse TRI prediction, there is one predicted TRI (SP1→TTF-1) that is not a TRI in TP at the FI threshold cutoff >= 0.95. This TRI is transferred from the SP1→TTF-1 in the source experimental TRIs of human. While this TRI is not in the source experimental TRIs of mouse, it is highly likely to be a real but not yet experimentally validated TRI.

In case of human TRI prediction, there are two predicted TRIs (ATF-2→HIST3H2A and HOXA5→HOXA5) not in the human TP at the FI threshold cutoff >= 0.95. These TRIs are transferred from the ATF-2 → HIST1H2AC and the HOXA5→HOXA5 in the source experimental TRIs of human and mouse, respectively. HIST1H2AC is transferred to HIST3H2A with the very high cutoff value, as they are isoforms, but at the FI threshold cutoff >= 0.98, only ATF-2→HIST3H2A remains (**Note 31**). While HOXA5→HOXA5 is not in the source experimental TRIs of human, it is also likely to be a real but not yet experimentally validated TRI.

The predicted TRI of yeast (ARS→ENO1), which is not in TP at the FI threshold >= 0.95, is transferred from the ARS→ENO2 in the source experimental TRIs of yeast, where ENO2 and ENO1 are isoforms.

Overall, the results of our sanity check (shown in **Table 6.3**) for the five organisms confirm that the method is correct.

2.5.4. Effects of Filters

Figure 6.6 shows how different filters affect the accuracy of TRI prediction (**Note 32**). Judging from our results, the use of binding sites for filtering predictions improves the accuracy of TRI prediction for all organisms (except fly, due to the limited number of source experimental TRIs) (**Notes 33, 34**). The use of the functional annotation of "transcription factor" from the Bioverse as a filter slightly improves the prediction accuracy for all organisms except human, which might be caused by too narrow a search with the Bioverse options limited to "transcription factor" or unavailable. In the case of yeast, the use of protein localization as a filter slightly improves the accuracy of prediction at low-to-medium FI threshold cutoffs. **Table 6.4** shows the numbers of predicted TRIs of different filters for the five target organisms at the FI threshold cutoff >= 0.3.

We investigate how protein families relate to the predicted TRIs by counting the number of TRIs for which TFs are sharing or not sharing protein families with their corresponding source TFs. **Table 6.5** lists the counting results for

Table 6.3
Results of the sanity checks for human, mouse, rat, fly, and yeast

	FI threshold	TRIs in TPs	TRIs not in TPs	% accuracy	% coverage
D. melanogaster	0.95	171	0	100	100
	0.96	171	0	100	100
	0.97	171	0	100	100
	0.98	171	0	100	100
	0.99	171	0	100	100
	1	171	0	100	100
H. sapiens	0.95	900	2	99.778	100
	0.96	900	2	99.778	100
	0.97	900	2	99.778	100
	0.98	900	1	99.889	100
	0.99	900	1	99.889	100
	1	900	0	100	100
M. musculus	0.95	309	1	99.677	100
	0.96	309	0	100	100
	0.97	309	0	100	100
	0.98	309	0	100	100
	0.99	309	0	100	100
	1	309	0	100	100
R. norvegicus	0.95	46	0	100	100
	0.96	46	0	100	100
	0.97	46	0	100	100
	0.98	46	0	100	100
	0.99	46	0	100	100
	1	46	0	100	100
S. cerevisiae	0.95	136	1	99.27	100
	0.96	136	0	100	100
	0.97	136	0	100	100
	0.98	136	0	100	100
	0.99	136	0	100	100
	1	136	0	100	100

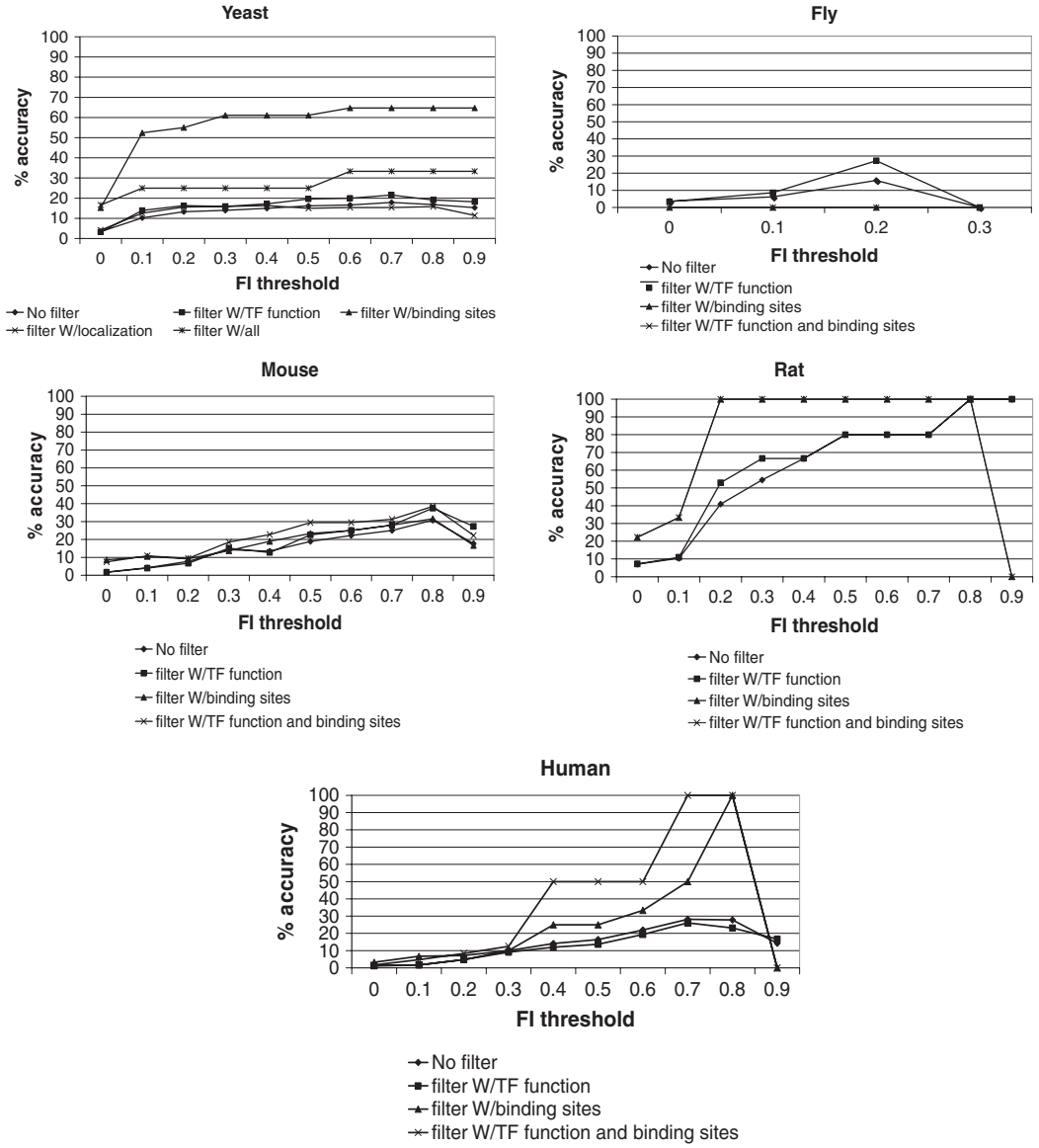

Fig. 6.6. Accuracy with no filters, accuracy with TF function filter, accuracy with binding site filter, accuracy with localization filter (in yeast only), accuracy with all filters for the test sets of yeast, fly, mouse, rat, and human.

the five organisms at the FI threshold cutoff >= 0.3. The numbers of predicted TRIs that display sharing are the largest compared to the numbers of predicted TRIs not sharing or not having protein family information, for all FI threshold cutoffs for all organisms except yeast (data not shown here). In case of yeast, most of the predicted TRIs are the TRIs for which TFs or their source TFs do not have protein family information.

Table 6.4
Numbers of predicted TRIs (in TP, not in TP) with different filters at the FI threshold cutoff >= 0.3 of yeast, fly, mouse, rat, and human from the benchmarking process

Organisms	No filter	W/TF function filter	W/binding site filter	W/all filters
S. cerevisiae	18, 111	11, 58	11, 7	1, 3
D. melanogaster	0, 1	0, 1	0, 0	0, 0
M. musculus	21, 126	13, 73	7, 44	5, 22
R. norvegicus	6, 5	6, 3	2, 0	2, 0
H. sapiens	13, 117	7, 70	1,9	1,7

Table 6.5
Numbers of predicted TRIs of TP and not in TP for which TFs (1) share protein families with their corresponding source TFs, (2) have no overlapped protein families with their source TFs, and (3) have no information of protein families, with no filters, at the FI threshold cutoff >= 0.3

Organisms	TRIs in TP	TRIs in TP, sharing protein family	TRIs in TP, no sharing protein family	TRIs in TP, no protein family info.	TRIs not in TP	TRIs not in TP, sharing protein family	TRIs not in TP, no sharing protein family	TRIs not in TP, no protein family info.
S. cerevisiae	18	3	0	15	111	32	0	79
D. melanogaster	0	0	0	0	1	0	0	1
M. musculus	21	15	3	3	126	93	13	20
R. norvegicus	6	5	0	1	5	4	0	1
H. sapiens	13	10	0	3	117	100	3	14

3. Notes

1. Orthologs are defined as best-matching homologs between a source and a target organism.
2. Similarly, an interolog is defined as the pair of interacting proteins A ←→ B in a source organism and its orthologous proteins A'←→B' in a target organism *(29)*.

3. The sources of experimental TRIs are limited, and the ways to access and gather them are varied. Among our source databases, RegulonDB is the only database that provides a way to download the TF to gene interactions in bulk. In the case of TRANSFAC®, we needed to write a script using the *urllib* module in Python to fetch data from the http server of TRANFAC®. In case of DBTBS, we resorted to a personal communication requesting the experimental TRIs from the authors.
4. The formats of the experimental TRIs differ from source to source. For instance, TRANSFAC® provides the experimental TRIs of eukaryotes via the records of transcription factors in html files, where each record will contain various information of the transcription factor and its regulating genes. DBTBS provides all experimental TRIs of *B. subtilis* in a single xml file. To handle these various formats, we wrote code for extracting the experimental TRIs from each specific source as described in **Section 2.1.1**.
5. We encountered the same problems for gathering and preparing the binding sites and binding sequences. The binding sites in TRANSFAC® came as a part of the transcription factor records and linked to their own records of specific binding sequences. Hence, we wrote code to extract the binding site accession numbers. Then we fetched the binding site records from the TRANFAC® http server and parsed these records for the binding sequences. In case of DBTBS, the binding sites and binding sequences appeared in specific xml tags. Hence, we developed code to parse and extract them. As RegulonDB provides bulk download of the TF binding sites, we only needed to edit the format of the downloaded file.
6. To extend a set of experimental TRIs for an organism from different sources, we use the script: run_appendTRIs_<organism>, where examples of organisms are "caenorhabditis_ elegans," "homo_sapiens," and "saccharomyces_cerevisiae." This script calls a code that appends additional TRIs compiled from other sources listed in an input file (e.g., ./inputs/TRIs/caenorhabditis_elegans_tris_fileList.txt, for *C. elegans*) to the available TRI file. For instance, the run_appendTRIs_caenorhabditis_elegans appends the TRIs compiled from WormBase (in file ../inputs/TRIs/WormBaseTRIs.csv) to the compiled TRIs from TRANSFAC.
7. BIND becomes a component of BOND (Biomolecular Object Network Databank), which contains not only BIND but also GenBank data and related tools.

8. We get a PubMed identifier for a paper by searching the paper at Entrez PubMed http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed.
9. The lengths of the upstream regions are specific and limited for different sources. For instance, the UCSC Genome Browser provides the upstream regions of human, rat, mouse, and fly with lengths 1,000, 2,000, and 5,000 bps whereas SGD provides only 1,000 bps upstream regions. In the case of prokaryotes, we extracted the upstream regions of each organism from their complete genome sequences, with the length of 500 bps (**Notes 35, 36**).
10. In this work, we do not take the directions (i.e., forward, reverse) of the strand of the upstream regions into account. Also, we do not consider the possible binding sites at the downstream region. At present, we are interested only in predicting the transcriptional regulatory interactions for which transcription factors bind to specific sequences in the upstream regions of a target gene. Nevertheless, the overall methods described for TRI prediction should be usable for these extensions as these affect only the scanning of the binding sequences during the filtering process. The data preparation should be extended for the downstream regions, and the scanning of binding sites should evaluate the reverse strand and downstream regions.
11. In the case of *S. cerevisiae*, instead of getting the upstream regions via the ftp server as described in **Section 2.1.2**, you might follow the following steps:
    - Go to http://www.yeastgenome.org/ *(44)*.
    - On the left side of this main page, in section "Download Data," select "Batch Download." A page "SGD Batch Download Tool" will appear. Specify the input chromosome on the right-hand side of Step 1, and then specify the type of data that you would like to retrieve in Step 2. Under the "Sequence data" section, check box "Genomic DNA + 1 kb upstream and 1 kb downstream of flanking sequence," and click the submit button.
    - After getting the result file for each chromosome, concatenate these files together for the upstream regions of all genes in the complete genome. Then, use the same script: run_extractUpstreamRegions_1000bp_saccharomyces_cerevisiae to parse, extract, and transform this file into the upstream region file from *S. cerevisiae*, ready for use by the system.
12. The upstream region files downloaded from UCSC Genome Browser contain only the upstream regions from transcription starts annotated separately from the coding initiation region.

So, they are not the complete sets of upstream regions. An alternative way to compile the upstream regions of genes in an organism is to find the location of the genes in the complete genome sequence and extract the sequence in front of the genes starting from their transcription start sites as the upstream regions.

13. In the case of *H. sapiens*, *M. musculus*, *R. norvegicus*, and *D. melanogaster*, one might use the ftp server instead of the http server:
    - Go to ftp://hgdownload.cse.ucsc.edu/goldenPath/.
    - Use the code in parentheses at the first line of a specific genome box at http://hgdownload.cse.ucsc.edu/downloads.html (e.g., hg18 for human genome) to select the directory under the ftp://hgdownload.cse.ucsc.edu/goldenPath/. Under this directory, select "bigZips" directory and save files upstreamxxx.zip into a local directory. The xxx represents the number of base pairs of each upstream region. Use our script: run_extract UpstreamRegions_1000 bp_homo_sapiens to parse, extract, and transform the saved files into the format ready for use by the system.
14. TIGR uses the results from IRGSP. Hence, the upstream regions of rice downloaded from TIGR are of the cultivar Nipponbare of *Oryza sativa* L. ssp. *japonica (57)*. The first draft sequence of *O. sativa* L. ssp. *indica* was also available and published in the same journal *(70)*.
15. The naming systems of the upstream regions of organisms are different from source to source. We needed to find the appropriate mapping from a specific ID system to the protein IDs in the Bioverse. The complexity in finding the mapping varied according to different sources. For instance, the upstream regions of eukaryotes (i.e., human, mouse, rat) downloaded from the UCSC Genome Browser were identified by RefSeq accession numbers for nucleotide sequences. However, we only had the mapping of NCBI GenBank Identifiers (GIs) to protein IDs in the Bioverse for these organisms. Hence, we needed to find the mapping from these RefSeq numbers to the GIs. On the other hand, the upstream regions of yeast, fly, worm, and *Arabidopsis* had been annotated by their specific ID systems already in the Bioverse. Hence, finding the mapping for these organisms was less complex. In case of rice, as TIGR uses TIGR_LOCUS as the main identifier for the rice genome to refer to the upstream regions, genes, and proteins, we needed to find the mapping from TIGR_LOCUS IDs to protein IDs in the Bioverse.

16. Several names of TFs and TFTs in the source experimental TRIs are not mapped with any intermediate ID system. Hence, the TRIs with these TFs and/or TFTs will be discarded, such that several source TRIs are lost during the mapping process.
17. We could improve the quality of the name mappings by finding and adding the synonyms of the common names from different sources to the name ID mapping file.
18. Building name mapping from TFs, TFTs to protein IDs in the Bioverse for *H. sapiens* is the most complicated. We encountered the following problems and limitations during the process of building the name mapping:
    - The naming of human genes is still not well defined. Even though several sources of human genes with common names are available, some of them are not updated and some others are obsolete. Sources of common names for *H. sapiens* are GenBank, the synonyms field associated with each transcription factor information files compiled from TRANSFAC®, OMIM *(59)*, Entrez Genes *(71)*, and HUGO *(58)*.
    - Human genes are much more complex than yeast. Several of them have the same common names but different protein products according to different isoforms. Hence, a straightforward many-to-one mapping of a common name and its synonyms to a specific intermediate ID (i.e., systematic ORF name from SGD) and to a protein ID in the Bioverse as in case of yeast is not always true in human.
19. Even though the name mapping could be refined with the synonyms of common genes from different sources, in general, these will not be complete. The better and more reliable mapping from the common names of TF and TFT in the experimental TRIs to protein IDs in the Bioverse uses sequence mapping. However, this is not applicable because this method involves protein sequences of all TFs and TFTs in all experimental TRIs. Nevertheless, TRANSFAC® provides only the protein sequences of TFs but not of TFTs, while other sources of experimental TRIs do not provide protein sequences. To get protein sequences for these TFs and TFTs, we return to the name mapping problem.
20. At present, we have only the protein localization of *S. cerevisiae* from two public databases, as described in **Section 2.2.1**. Also, in our current assumption for filtering using protein localization, a predicted TRI will be discarded if its TF and TFT do not share localization. This assumption might be too restrictive and needs further investigation.

Table 6.6
Public databases of protein subcellular localization from experiments

Database	Description	URL
UniProtKB/Swiss-Prot *(76)*	An annotated protein sequence database	http://www.ebi.ac.uk/swissprot/FTP/ftp.html
Comprehensive Yeast Genome Database (CYGD) *(78)*	MIPs *Saccharomyces cerevisiae* genome database	http://mips.gsf.de/genre/proj/yeast/index.jsp
MitoP2 *(79)*	A database for mitochondria-related genes, proteins, and diseases	http://www.mitop.de:8080/mitop2/
LOCATE *(80)*	A curated database of membrane, organization and subcellular localization mouse proteins of RIKEN FANTOM3	http://locate.imb.uq.edu.au/

21. In addition to TRIPLES and YEAST GFP mentioned in **Section 2.2.1**, we list additional public databases of experimental protein localization compiled from a literature search in **Table 6.6** and sources of protein localization based on the predictions in **Table 6.7**. Beside these two tables, additional systems and programs for protein subcellular localization can be found at http://www.molecularstation.com/protein/bioinformatics/subcellularlocalization/.
22. Besides protein families compiled from TRANSFAC®, we list other public databases of protein families in **Table 6.8**. Additional resources of protein families can be found at http://www.proweb.org/other.html.
23. At present, we have only protein families compiled from TRANSFAC®, as described in **Section 2.2.2**. Our current filtering method using protein families will discard all predicted TRIs for which the related TFs do not share any protein families with their corresponding TFs from source TRIs. Hence, with protein families compiled only from TRANSFAC®, the filtering might discard the predicted TRIs that are real.
24. In addition to using protein families for filtering the predicted TRIs as described in **Section 2.2.2**, we might utilize the combination of protein domains and domain architectures *(72–75)* in finding the similarity between two protein sequences. Also, instead of assigning the same weights for all overlapped locations between two aligned protein sequences,

Table 6.7
Sources of protein subcellular localization from prediction

Source	Description	URL
PSORT *(81)*	Predicting protein subcellular localization based on stored rules and prediction of sorting signals	http://psort.hgc.jp/
Yeast Protein Localization Server *(82)*	Predicting subcellular location of proteins in yeast using Bayesian formalism	http://bioinfo.mbb.yale.edu/genome/localize/
LOC3d *(83)*	A database of predicted subcellular localization for eukaryotic PDB chains	http://cubic.bioc.columbia.edu/db/LOC3d/
TargetP1.1 *(84)*	Predicting subcellular localization of protein in eukaryotes based on the predicted presence of N-terminal pre-sequences, chloroplast transit peptide (cTP), mitochondrial targeting peptide (mTP), or secretory pathway signal peptide (SP)	http://www.cbs.dtu.dk/services/TargetP/
SubLoc v1.0 *(85)*	A system for predicting protein subcellular localization using Support Vector Machine (SVM)	http://www.bioinfo.tsinghua.edu.cn/SubLoc/

Table 6.8
Public databases of protein domains and families

Database	Description	URL
InterPro (*86, 87)*	A resource of protein families, domains and functional sites, integrated from other databases such as Pfam, PROSITE, PRINTS, etc.	http://www.ebi.ac.uk/interpro/
Pfam *(88)*	A database of curated protein domains and families based on multiple sequence alignments and hidden Markov models	http://www.sanger.ac.uk/Software/Pfam/
PROSITE *(89)*	A database of protein families and domains that consists of biologically significant sites, patterns, and profiles used for identifying a family for a new protein	http://www.expasy.org/prosite/

(continued)

Table 6.8 (continued)

Database	Description	URL
ProDom *(90)*	A database of protein domain families automatically generated from Swiss-Prot and TrEMBL databases	http://prodom.prabi.fr/prodom/current/html/home.php
BLOCKS *(91, 92)*	A database of aligned ungapped segments derived from the most highly conserved regions in groups of proteins	http://blocks.fhcrc.org/
PRINTS *(93)*	A collection of protein fingerprints, where each is a group of conserved motifs used to classify a protein family	http://www.bioinf.man.ac.uk/dbbrowser/PRINTS/
TIGRFAMs *(94)*	A collection of curated protein families that consists of various models including, multiple sequence alignments, hidden Markov models (HMMs), Gene Ontology (GO) assignments, and literature references, for instance.	http://www.tigr.org/TIGRFAMs/
SYSTERS *(95)*	A database of protein families based on graph-based algorithms for protein sequences partitioning, clustering, and searching	http://systers.molgen.mpg.de/
SCOP *(96, 97)*	A database of proteins ordered by structural classification; protein domains are classified into families, superfamilies, folds, and classes	http://scop.mrc-lmb.cam.ac.uk/scop/
SMART *(98)*	A web-based tool and database, which allows the identification of signaling domains, the genetically mobile domains, and the analysis of domain architectures	http://smart.embl-heidelberg.de/
SUPERFAMILY *(99, 100)*	A database of hidden Markov models (HMMs) of known structures with protein domain classification based on SCOP	http://supfam.org/SUPERFAMILY/
CATH *(101)*	A database of protein domain structures, which categorizes proteins at four levels: Class (C), Architecture (A), Topology (T), and Homologous superfamily (H)	http://www.cathdb.info/

(continued)

Table 6.8 (continued)

Database	Description	URL
Gene3D (*102*)	A database of combined structures, functions, and evolutions of proteins, with HMMs based on CATH domain families, for structural annotation	http://gene3d.biochem.ucl.ac.uk/Gene3D/
PIRSF (*103*)	A super family classification system based on the relationships of protein evolutions	http://pir.georgetown.edu/pirsf/
PANTHER (*104, 105*)	A database of protein families, subfamilies, functions, pathways, and sequence and function evolutions	http://www.pantherdb.org/
CDD (*106*)	A conserved domain database at NCBI for protein classification	http://130.14.29.110/Structure/cdd/cdd.shtml

we need to give more weight for the locations that are considered protein domains. This higher similarity specificity should help to improve the accuracy of prediction.

25. Instead of getting protein sequences from the Bioverse via an XML RPC server, we might get protein sequences from other sources such as protein database at NCBI, UniProt (the universal protein resource) *(76)*, TAIR for *A. thaliana*, TIGR Rice Genome Annotation for *O. sativa*, WormBase for *C. elegans*, FlyBase for *D. melanogaster*, and SGD for *S. cerevisiae*.
26. For BLAST practical usage, see BLAST tutorial at http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/tut1.html for additional information.
27. For PSI-BLAST practical usage, see PSI-BLAST tutorial at http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/psi1.html for additional information.
28. Raw score, "S", is calculated as the sum of the substitution and gap scores (source: http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Blast_output.html).
29. We define accuracy as the fraction of TRIs predicted by the system that are in true positive set (TP) over the total predicted TRIs at a specific cutoff. As we did not have a set of false positives (FP), we define the test set of the TRIs that includes all combinations of individual TF to individual TFTs in the TP set. This means that if we have 121 TRIs in the TP which consists of 49

TFs and 83 TFTs, then the test set will consist of $49 \times 83 = 4{,}067$ TRIs. Note that it is possible that some TRIs in the 4,067 that are not in the TP might be real. Hence, what we have for accuracy calculation is minimum coverage. To improve accuracy and coverage, we need to find a gold standard of TP for which TRIs are known with high confidence so that any other TRI is not possible among the set of their TFs and TFTs.

30. The FI product is the multiplication of fraction identity (FI) between a source TF and a target TF and a source TFT to a target TFT. The fraction identity is the number of identical amino acids that are overlapped between the aligned source and the target TFs (or source and target TFTs) over the total number of amino acids of the target TF (or target TFT).
31. The predicted TRIs that have high similarity of interaction but are not in the TP might come from isoforms. To clean up accuracy and coverage at the high similarity of interactions for benchmarking, we omit the predicted TRIs resulting from the isoforms of target TF and TFT.
32. The available experimental TRIs are incomplete. Even though we tried to gather the experimental TRIs of each source organism from different sources such as public databases and supplemental data from the literature, there is no way to compile the complete set of true positives for a target organism. This resulted in lower accuracy and coverage in benchmarking the TRI predictions for almost all target organisms. This is a limitation of the available data set. In fact, predictions that are not in our TP might be real TRIs.

    To alleviate this limitation, we attempt to filter out the predicted TRIs using additional data sets such as binding sites, binding sequences, protein localization, protein families, and functional annotation. Also, several predicted TRIs are the same among organisms (**Note 37**). Even though they are not yet verified by experiments, nor are they in the TPs, we consider them as real interactions that should be removed from the set of predicted TRIs not in the TP.
33. The available binding sites and binding sequences are incomplete. We cannot determine a predicted TRI as false even though the upstream region of the TFT of the predicted TRI does not have the binding sites and binding sequences of the TF of its source experimental TRI. It is possible that the TF might have alternative sites and sequences that are not yet studied. Hence, we mainly use the available binding sites and binding sequences for improving the confidence of prediction.
34. The location of binding sites and their sequences do not need to be exactly the same between the TF of the source experimental TRI and the TF of a predicted TRI. Hence, we relaxed

this restriction using 80% matching for scanning the binding sequences with the varied lengths of upstream regions of target TFT in the predicted TRI. Note that the scanning of binding sequences of *B. subtilis* needed a special treatment, such that some of its binding sequences required exact matching while others could be relaxed.

35. We could change this length of upstream regions and rerunning the extraction.
36. RegulonDB also provides promoters of *E. coli* K-12. However, we have not integrated these promoters as part of the extracted upstream regions of *E. coli*.
37. A predicted TRI in a target organism is considered the same as a source experimental TRI if its TF' and TFT' have the same names as of the TF and TFT of the source experimental TRI.
38. Databases at NCBI such as Nucleotide, Protein have been providing XML as an alternative formats for their downloadable data.

4. Conclusions

Data preparation and integration is one of the major tasks in our prediction of TRIs by a homology-based approach. As data from different sources have different formats, we need to handle them differently. In this chapter, we provide a set of Python programs and shell scripts that help to retrieve and manipulate data from various sources. Definitely, these make this process easier. However, public data sets are being released every day, and they might have rearrangements of data formats and identifiers. Hence, in the long term it would be better if various sources provide their data sets in a standard data exchange format like XML (**Note 38**) and/or provide standard interfaces (e.g., SOAP, RPC, Web Services) for remote programs to be able to automatically interconnect and work together with each other. In terms of name mapping, while different sources might have their own identification systems, it would be much easier for users if they also provided a mapping from their ID to a standard ID. Hence, while what we presented in this chapter seems straightforward, it becomes very complicated in terms of bookkeeping, and this is very rarely talked about explicitly in papers that describe results like these.

The compiled experimental TRIs and the predicted TRIs can be integrated with protein–protein interactions (PPIs) and other data types such as gene expression for investigating specific pathways. **Figure 6.7** shows an example of the integrated network between TRIs and PPIs for investigating the regulation of CBF1

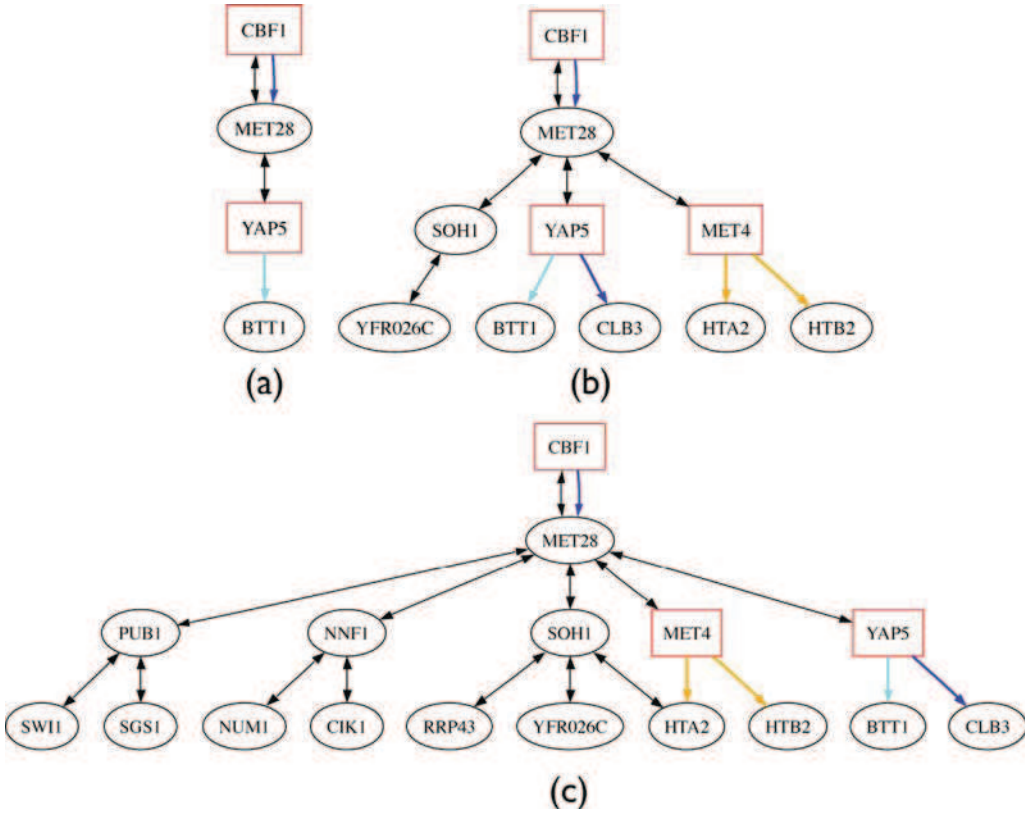

Fig. 6.7. The integrated networks that include experimentally verified TRIs from TRANSFAC and (**a**) TRIs from high-throughput experiment (1) with *p*-value <= 0.005 and PPIs in the Bioverse with confidence = 1, (**b**) TRIs from high-throughput experiment (1) with *p*-value <= 0.01 and PPIs in the Bioverse with confidence = 1, and (**c**) TRIs from high-throughput experiment (1) with *p*-value <= 0.01 and PPIs in the Bioverse with confidence >= 0.8. All these three networks had been filtered with differentially expressed genes listed in (2). An *arrow* with heads on both ends represents a PPI and an *arrow* with a head only at its end represents a TRI. A *cyan, blue, and orange* TRI has *p*-value <= 0.001, 0.005, and 0.01, respectively.

to MET28 (CBF1→MET28) and their related genes, proteins, and interactions, in yeast cell cycle phase S. **Figure 6.7a, b and c** represent the integrated networks with different settings and filtered by differentially expressed genes downloaded from Luscombe et al. *(77)*.

Acknowledgments

This work was supported in part by a Searle Scholar Award and NSF Grant DBI-0217241 to R.S., and the University of Washington's Advanced Technology Initiative in Infectious

Diseases. Also, it is supported in part by the National Center for Genetic Engineering and Biotechnology (BIOTEC), National Science Technology & Development Agency (NSTDA), Thailand.

References

1. Lee, T.I., et al., Transcriptional regulatory networks in Saccharomyces cerevisiae. Science, 2002. **298**(5594): 799–804.
2. Deplancke, B., et al., A gene-centered C. elegans protein-DNA interaction network. Cell, 2006. **125**(6): 1193–1205.
3. McDermott, J., et al., *BIOVERSE:* Enhancements to the framework for structural, functional and contextual modeling of proteins and proteomes. Nucl. Acids Res., 2005. **33**(suppl_2): W324–W325.
4. H Caron, et al., The Human Transcriptome Map reveals a clustering of highly expressed genes in chromosomal domains. Science, 2001. **291**: 1289–1292.
5. Shen-Orr, S.S., et al., Network motifs in the transcriptional regulation network of Escherichia coli. Nat. Genet., 2002. **31**(1): 64–68.
6. Martinez-Antonio, A. and J. Collado-Vides, Identifying global regulators in transcriptional regulatory networks in bacteria. Curr. Opin. Microbiol., 2003. **6**(5): 482–489.
7. Harbison, C.T., et al., Transcriptional regulatory code of a eukaryotic genome. Nature, 2004. **431**(7004): 99.
8. Proft, M., et al., Genomewide identification of Sko1 target promoters reveals a regulatory network that operates in response to osmotic stress in Saccharomyces cerevisiae. Eukaryot. Cell, 2005. **4**(8): 1343–1352.
9. Sharma, M.R., et al., Transcriptional networks in a rat model for nonalcoholic fatty liver disease: A microarray analysis. Exp. Mol. Pathol., 2006. [Epub ahead of print].
10. Reymann, S. and J. Borlak, Transcriptome profiling of human hepatocytes treated with Aroclor 1254 reveals transcription factor regulatory networks and clusters of regulated genes. BMC Genomics, 2006. 7(1): 217.
11. Makita, Y., et al., *DBTBS:* Database of transcriptional regulation in Bacillus subtilis and its contribution to comparative genomics. Nucl. Acids Res., 2004. **32**(suppl_1): D75–D77.
12. Matys, V., et al., TRANSFAC(R) and its module TRANSCompel(R): Transcriptional gene regulation in eukaryotes. Nucl. Acids Res., 2006. **34**(suppl_1): D108–D110.
13. Salgado, H., et al., RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. Nucl. Acids Res., 2006. **34**(suppl_1): D394–D397.
14. Segal, E., et al., Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. Nat. Genet., 2003. **34**(2): 166–176.
15. Pilpel, Y., P. Sudarsanam, and G. M. Church, Identifying regulatory networks by combinatorial analysis of promoter elements. Nat. Genet., 2003. **29**: 153–159.
16. Yeger-Lotem, E., et al., Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. PNAS, 2004. **101**(16): 5934–5939.
17. Yu, T. and K.-C. Li, Inference of transcriptional regulatory network by two-stage constrained space factor analysis. Bioinformatics, 2005. **21**(21): 4033–4038.
18. Zhang, L., et al., Motifs, themes and thematic maps of an integrated Saccharomyces cerevisiae interaction network. J. Biol., 2005. **4**(2): 6.
19. Jiang, R., et al., Network motif identification in stochastic networks. PNAS, 2006. **103**(25): 9404–9409.
20. Mandel-Gutfreund, Y. and H. Margalit, Quantitative parameters for amino acid-base interaction: Implications for prediction of protein-DNA binding sites. Nucl. Acids Res., 1998. **26**(10): 2306–2312.
21. Luscombe, N.M. and J.M. Thornton, Protein–DNA interactions: Amino acid conservation and the effects of mutations on binding specificity. J. Mol. Biol., 2002. **320**(5): 991–1009.
22. Kato, M., et al., Identifying combinatorial regulation of transcription factors and binding motifs. Genome Biol., 2004. **5**(8): R56.
23. Morozov, A.V., et al., Protein-DNA binding specificity predictions with structural models. Nucl. Acids Res., 2005. **33**(18): 5781–5798.
24. Gertz, J., et al., Discovery, validation, and genetic dissection of transcription factor binding sites by comparative and functional genomics. Genome Res., 2005. **15**(8): 1145–1152.

25. Tompa, M., et al., Assessing computational tools for the discovery of transcription factor binding sites. Nat. Biotechnol., 2005. **23**: 137–144.
26. GuhaThakurta, D., Computational identification of transcriptional regulatory elements in DNA sequence. Nucl. Acids Res., 2006. **34**(12): 3585–3598, doi: 10.1093/nar/gkl372.
27. Yu, H., et al., Annotation transfer between genomes: Protein-protein interologs and protein-DNA regulogs. Genome Res., 2004. **14**(6): 1107–1118.
28. Du, W., et al., RBF, a novel RB-related gene that regulates E2F activity and interacts with cyclin E in Drosophila. Genes Dev., 1996. **10**(10): 1206–1218.
29. Walhout, A.J.M., et al., Protein interaction mapping in C. elegans using proteins involved in vulval development. Science, 2000. **287**: 116–122.
30. Matthews, L.R., et al., Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or "Interologs". Genome Res., 2001. **11**(12): 2120–2126.
31. Lehner, B. and A.G. Fraser, A first-draft human protein-interaction map. Genome Biol., 2004. **5**(9): R63.1–9.
32. Huang, T.-W., et al., POINT: A database for the prediction of protein-protein interactions based on the orthologous interactome. Bioinformatics, 2004. **20**(17): 3273–3276.
33. Kemmer, D., et al., Ulysses – an application for the projection of molecular interactions across species. Genome Biol., 2005. **6**(12): R106.
34. Brown, K.R. and I. Jurisica, Online predicted human interaction database. Bioinformatics, 2005. **21**(9): 2076–2082.
35. von Mering, C., et al., STRING: Known and predicted protein-protein associations, integrated and transferred across organisms. Nucl. Acids Res., 2005. **33**(suppl_1): D433–D437.
36. Zhu, J. and M.Q. Zhang, *SCPD: A promoter database of the yeast Saccharomyces cerevisiae*. Bioinformatics, 1999. **15**(7): 607–611.
37. Bader, G.D., D. Betel, and C.W. Hogue, *BIND: The biomolecular interaction network database*. Nucl. Acids Res., 2003. **31**(1): 248–250.
38. Alfarano, C., et al., The biomolecular interaction network database and related tools 2005 update. Nucl. Acids Res., 2005. **33**(suppl_1): D418–D424, doi: 10.1093/nar/gki051.
39. Chen, N., et al., WormBase: A comprehensive data resource for Caenorhabditis biology and genomics. Nucl. Acids Res., 2005. **33**(suppl_1): D383–D389.
40. Schwarz, E.M., et al., WormBase: Better software, richer content. Nucl. Acids Res., 2006. **34**(suppl_1): D475–D478, doi: 10.1093/nar/gkj061.
41. Hayakawa, J., et al., Identification of promoters bound by c-Jun/ATF2 during rapid large-scale gene activation following genotoxic stress. Mol. Cell, 2004. **16**(4): 521.
42. Kim, J., et al., Mapping DNA-protein interactions in large genomes by sequence tag analysis of genomic enrichment. Nat. Meth., 2005. **2**(1): 47.
43. Kim, T.H., et al., Direct isolation and identification of promoters in the human genome. Genome Res., 2005. **15**(6): 830–839.
44. Hong, E.L., et al., Saccharomyces Genome Database. ftp://ftp.yeastgenome.org/yeast/September, 2006.
45. Hinrichs, A.S., et al., The UCSC genome browser database: Update 2006. Nucl. Acids Res., 2006. **34**(suppl_1): D590–D598.
46. Michael, J.M., Initial sequencing and analysis of the human genome. Nature, 2001. **409**(6822): 860.
47. Waterston, R.H., et al., Initial sequencing and comparative analysis of the mouse genome. Nature, 2002. **420**(6915): 520.
48. Gibbs, R.A., et al., Genome sequence of the Brown Norway rat yields insights into mammalian evolution. Nature, 2004. **428**(6982): 493.
49. Adams, M.D., et al., The genome sequence of drosophila melanogaster. Science, 2000. **287**(5461): 2185–2195.
50. The C. elegans Sequencing Consortium, Genome sequence of the nematode C. elegans: A platform for investigating biology. Science, 1998. **282**(5396): 2012–2018.
51. Rhee, S.Y., et al., The Arabidopsis Information Resource (TAIR): A model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. Nucl. Acids Res., 2003. **31**(1): 224–228.
52. The Arabidopsis Genome, I., Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. Nature, 2000. **408**(6814): 796.
53. Theologis, A., et al., Sequence and analysis of chromosome 1 of the plant Arabidopsis thaliana. Nature, 2000. **408**(6814): 816.

54. European Union Chromosome 3 Arabidopsis Genome Sequencing, C., R. The Institute for Genomic, and D.N.A.R.I. Kazusa, Sequence and analysis of chromosome 3 of the plant Arabidopsis thaliana. Nature, 2000. **408**(6814): 820.
55. Kazusa, D.N.A.R.I., et al., Sequence and analysis of chromosome 5 of the plant Arabidopsis thaliana. Nature, 2000. **408**(6814): 823.
56. Yuan, Q., et al., The institute for genomic research Osa1 rice genome annotation database. Plant Physiol., 2005. **138**(1): 18–26.
57. Goff, S.A., et al., A draft sequence of the rice genome (Oryza sativa L. ssp. japonica). Science, 2002. **296**(5565): 92–100.
58. HUGO Gene Nomenclature Committee http://www.genenames.org/data/gdlw_index.html September 2006.
59. Online Mendelian Inheritance in Man, OMIM (TM). McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD), {September 2006}. World Wide Web URL: http://www.ncbi.nlm.nih.gov/omim/
60. Huh, W.-K., et al., Global analysis of protein localization in budding yeast. Nature, 2003. **425**: 686–691.
61. Drawid, A., R. Jansen, and M. Gerstein, Genome-wide analysis relating expression level with protein subcellular localization. Trends Genet., 2000. **16**(10): 426.
62. Ross-Macdonald, P., et al., Large-scale analysis of the yeast genome by transposon tagging and gene disruption. Nature, 1999. **402**(6760): 413.
63. Kumar, A., et al., TRIPLES: A database of gene function in Saccharomyces cerevisiae. Nucl. Acids Res., 2000. **28**(1): 81–84.
64. Tatiana, T.A. and T.L. Madden, Blast 2 sequences – a new tool for comparing protein and nucleotide sequences. FEMS Microbiol Lett., 1999. **174**: 247–250.
65. Altschul, S.F., et al., Basic local alignment search tool. J. Mol. Biol., 1990. **215**(3): 403–410.
66. Altschul, S.F., et al., Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucl., Acids Res., 1997. **25**(17): 3389–3402.
67. Smith, T.F. and M.S. Waterman, Identification of common molecular subsequences. J. Mol. Biol., 1981. **147**(1): 195–197.
68. Pearson, W.R., Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics, 1991. **11**(3): 635–650.
69. Thompson, J.D., D.G. Higgins, and T.J. Gibson, CLUSTAL W: Improving the sensitivity of progressivemultiple sequence alignment through sequence weighting,position-specific gap penalties and weight matrix choice. Nucl. Acids Res., 1994. **22**(22): 4673–4680.
70. Yu, J., et al., A draft sequence of the rice genome (Oryza sativa L. ssp. indica). Science, 2002. **296**(5565): 79–92.
71. Donna Maglott, et al., Entrez Gene: Gene-centered information at NCBI. Nucl. Acids Res., 2005. **33**(Database): D54–D58.
72. Bashton, M. and C. Chothia, The geometry of domain combination in proteins. J. Mol. Biol., 2002. **315**(4): 927.
73. Bjorklund, A.K., et al., Domain rearrangements in protein evolution. J. Mol. Biol., 2005. **353**(4): 911.
74. Geer, L.Y., et al., CDART: Protein homology by domain architecture. Genome Res., 2002. **12**(10): 1619–1623, doi: 10.1101/gr.278202,.
75. Hegyi, H. and M. Gerstein, Annotation transfer for genomics: Measuring functional divergence in multi-domain proteins. Genome Res., 2001. **11**(10): 1632–1640, doi: 10.1101/gr. 183801.
76. The UniProt Consortium, The Universal Protein Resource (UniProt). Nucl. Acids Res., 2007. **35**(suppl_1): D193–D197, doi: 10.1093/nar/gkl929.
77. Luscombe, N.M., et al., Genomic analysis of regulatory network dynamics reveals large topological changes. Nature, 2004. **431**: 308–312.
78. Guldener, U., et al., CYGD: The comprehensive yeast genome database. Nucl. Acids Res., 2005. **33**(suppl_1): D364–D368, doi: 10.1093/nar/gki053.
79. Andreoli, C., et al., MitoP2, an integrated database on mitochondrial proteins in yeast and man. Nucl. Acids Res., 2004. **32**(1): D459–D462.
80. Fink, J.L., et al., LOCATE: A mouse protein subcellular localization database. Nucl. Acids Res., 2006. **34**(suppl_1): D213–D217, doi: 10.1093/nar/gkj069.
81. Nakai, K. and P. Horton, PSORT: A program for detecting the sorting signals of proteins and predicting their subcellular localization. Trends Biochem. Sci., 1999. **24**(1): 34–35.
82. Drawid, A. and M. Gerstein, A Bayesian system integrating expression data with sequence patterns for localizing proteins: Comprehensive application to the yeast genome. J. Mol. Biol., 2000. **301**: 1059–1075.

83. Nair, R. and B. Rost, LOC3D: Annotate subcellular localization for protein structures. Nucl. Acids Res., 2003. **31**(13): 3337–3340.
84. Olof Emanuelsson, et al., Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J. Mol. Biol., 2000. **300**: 1005–1016.
85. Hua, S. and Z. Sun, Support vector machine approach for protein subcellular localization prediction. Bioinformatics, 2001. **17**(8): 721–728.
86. Mulder, N.J., et al., InterPro, progress and status in 2005. Nuc. Acids Res., 2005. **33**(Database issue): D201–D205.
87. Mulder, N.J., et al., New developments in the InterPro database. Nucl. Acids Res., 2007. **35**(suppl_1): D224–D228, doi: 10.1093/nar/gkl841.
88. Finn, R.D., et al., Pfam: Clans, web tools and services. Nucl. Acids Res., 2006. **34**(Database issue): D247–D251.
89. Hulo, N., et al., The PROSITE database. Nucl. Acids Res., 2006. **34**(Database issue): D227–D230.
90. Catherine B., et al., The ProDom database of protein domain families: More emphasis on 3D. Nucl. Acids Res., 2005. **33**(Database Issue): D212–D215.
91. Henikoff, S., J.G. Henikoff, and S. Pietrokovski, Blocks+: A non-redundant database of protein alignment blocks dervied from multiple compilations. Bioinformatics, 1999. **15**(6): 471–479.
92. Henikoff, J.G., et al., Increased coverage of protein families with the blocks database servers. Nucl. Acids Res., 2000. **28**: 228–230.
93. Attwood, T.K., et al., PRINTS and its automatic supplement, prePRINTS. Nucl. Acids Res., 2003. **31**: 400–402.
94. Haft, D.H., J.D. Selengut, and O. White, The TIGRFAMs database of protein families. Nucl. Acids Res., 2003. **31**: 371–373.
95. Meinel, T., A. Krause, H. Luz, M. Vingron, and E. Staub, The SYSTERS protein family database in 2005. Nucl. Acids Res., 2005. **33**(Database issue): D226–D229.
96. Murzin, A.G., et al., SCOP: A structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol., 1995. **247**: 536–540.
97. Andreeva A., et al., SCOP database in 2004: Refinements integrate structure and sequence family data. Nucl. Acid Res., 2004. **32**: D226–D229.
98. Letunic, I., et al., SMART 5: Domains in the context of genomes and networks. Nucl. Acids Res., 2006. **34**(suppl_1): D257–D260, doi: 10.1093/nar/gkj079.
99. Gough, J., et al., Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. J. Mol. Biol., 2001. **313**(4): 903–919.
100. Gough, J. and C. Chothia, SUPERFAMILY: HMMs representing all proteins of known structure. SCOP sequence searches, alignments and genome assignments. Nucl. Acids Res., 2002. **30**(1): 268–272, doi: 10.1093/nar/30.1.268.
101. Pearl, F., et al., The CATH domain structure database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. Nucl. Acids Res., 2005. **33**(suppl_1): D247–D251, doi: 10.1093/nar/gki024.
102. Yeats, C., et al., Gene3D: Modelling protein structure, function and evolution. Nucl. Acids Res., 2006. **34**(suppl_1): D281–D284, doi: 10.1093/nar/gkj057.
103. Wu, C.H., et al., PIRSF: Family classification system at the protein information resource. Nucl. Acids Res., 2004. **32**(suppl_1): D112–D114, %R 10.1093/nar/gkh097.
104. Mi, H., et al., The PANTHER database of protein families, subfamilies, functions and pathways. Nucl. Acids Res., 2005. **33**(suppl_1): D284–D288, 10.1093/nar/gki078.
105. Mi, H., et al., PANTHER version 6: Protein sequence and function evolution data with expanded representation of biological pathways. Nucl. Acids Res., 2007. **35**(suppl_1): D247–D252, doi: 10.1093/nar/gkl869.
106. Marchler-Bauer, A., et al., CDD: A conserved domain database for protein classification. Nucl. Acids Res., 2005. **33**(suppl_1): D192–D196, doi: 10.1093/nar/gki069.