10

Reverse Engineering Gene Regulatory Networks by Integrating Multi-Source Biological Data

Yuji Zhang[1], Habtom W. Ressom[2] and Jean-Pierre A. Kocher[1]
[1]*Division of Biomedical Statistics and Informatics, Department of Health Sciences Research, Mayo Clinic College of Medicine, Rochester, MN*
[2]*Department of Oncology, Lombardi Comprehensive Cancer Center at Georgetown University Medical Center, Washington, DC*
USA

1. Introduction

Gene regulatory network (GRN) is a model of a network that describes the relationships among genes in a given condition. The model can be used to enhance the understanding of gene interactions and provide better ways of elucidating environmental and drug-induced effects (Ressom et al. 2006).

During last two decades, enormous amount of biological data generated by high-throughput analytical methods in biology produces vast patterns of gene activity, highlighting the need for systematic tools to identify the architecture and dynamics of the underlying GRN (He et al. 2009). Here, the system identification problem falls naturally into the category of **reverse engineering**; a complex genetic network underlies a massive set of expression data, and the task is to infer the connectivity of the genetic circuit (Tegner et al. 2003). However, reverse engineering of a global GRN remains challenging because of several limitations including the following:

1. Tens of thousands of genes act at different temporal and spatial combinations in living cells;
2. Each gene interacts virtually with multiple partners either directly or indirectly, thus possible relationships are dynamic and non-linear;
3. Current high-throughput technologies generate data that involve a substantial amount of noise; and
4. The sample size is extremely low compared with the number of genes (Clarke et al. 2008).

These inherited properties create significant problems in analysis and interpretation of these data. Standard statistical approaches are not powerful enough to dissect data with thousands of variables (i.e., semi-global or global gene expression data) and limited sample sizes (i.e., several to hundred samples in one experiment). These properties are typical in microarray and proteomic datasets (Bubitzky et al. 2007) as well as other high dimensional data where a comparison is made for biological samples that tend to be limited in number, thus suffering from curse of dimensionality (Wit and McClure 2006).

One approach to address the curse of dimensionality is to integrate multiple large data sets with prior biological knowledge. This approach offers a solution to tackle the challenging task of inferring GRN. Gene regulation is a process that needs to be understood at multiple levels of description (Blais and Dynlacht 2005). A single source of information (e.g., gene expression data) is aimed at only one level of description (e.g., transcriptional regulation level), thus it is limited in its ability to obtain a full understanding of the entire regulatory process. Other types of information such as various types of molecular interaction data by yeast two-hybrid analysis or genome-wide location analysis (Ren et al. 2000) provide complementary constraints on the models of regulatory processes. By integrating limited but complementary data sources, we realize a mutually consistent hypothesis bearing stronger similarity to the underlying causal structures (Blais and Dynlacht 2005). Among the various types of high-throughput biological data available nowadays, time course gene expression profiles and molecular interaction data are two complementary sets of information that are used to infer regulatory components. Time course gene expression data are advantageous over typical static expression profiles as time can be used to disambiguate causal interactions. Molecular interaction data, on the other hand, provide high-throughput qualitative information about interactions between different entities in the cell. Also, prior biological knowledge generated by geneticists will help guide inference from the above data sets and integration of multiple data sources offers insights into the cellular system at different levels.

A number of researches have explored the integration of multiple data sources (e.g., time course expression data and sequence motifs) for GRN inference (Spellman et al. 1998; Tavazoie et al. 1999; Simon et al. 2001; Hartemink et al. 2002). A typical approach for exploiting two or more data sources uses one type of data to validate the results generated independently from the other (i.e., without data fusion). For example, cluster analysis of gene expression data was followed by the identification of consensus sequence motifs in the promoters of genes within each cluster (Spellman et al. 1998). The underlying assumption behind this approach is that genes co-expressed under varying experimental conditions are likely to be co-regulated by the same transcription factor or sets of transcription factors. Holmes et al. constructed a joint likelihood score based on consensus sequence motif and gene expression data and used this score to perform clustering (Holmes and Bruno 2001). Segal et al. built relational probabilistic models by incorporating gene expression and functional category information as input variables (Segal et al. 2001). Gene expression data and gene ontology data were combined for GRN discovery in B cell (Tuncay et al. 2007). Computational methodologies that allow systematic integration of data from multiple resources are needed to fully use the complementary information available in those resources.

Another way to reduce the complexity in reverse engineering a GRN is to decompose it into simple units of commonly used network structures. GRN is a network of interactions between transcription factors and the genes they regulate, governing many of the biological activities in cells. Cellular networks like GRNs are composed of many small but functional modules or units (Wang et al. 2007). Breaking down GRNs into these functional modules will help understanding regulatory behaviors of the whole networks and study their properties and functions. One of these functional modules is called network motif (NM) (Milo et al. 2004). Since the establishment of the first NM in Escherichia coli (Rual et al.

2005), similar NMs have also been found in eukaryotes including yeast (Lee et al. 2002), plants (Wang et al. 2007), and animals (Odom et al. 2004; Boyer et al. 2005; Swiers et al. 2006), suggesting that the general structure of NMs are evolutionarily conserved. One well known family of NMs is the feed-forward loop (Mangan and Alon 2003), which appears in hundreds of gene systems in E. coli (Shen-Orr et al. 2002; Mangan et al. 2003) and yeast (Lee et al. 2002; Milo et al. 2002), as well as in other organisms (Milo et al. 2004; Odom et al. 2004; Boyer et al. 2005; Iranfar et al. 2006; Saddic et al. 2006; Swiers et al. 2006). A comprehensive review on NM theory and experimental approaches could be found in the review article (Alon 2007). Knowledge of the NMs to which a given transcription factor belongs facilitates the identification of downstream target gene modules. In yeast, a genome-wide location analysis was carried out for 106 transcription factors and five NMs were considered significant: autoregulation, feed-forward loop, single input module, multi-input module and regulator cascade.

In this Chapter, we propose a computational framework that integrates information from time course gene expression experiment, molecular interaction data, and gene ontology category information to infer the relationship between transcription factors and their potential target genes at NM level. This was accomplished through a three-step approach outlined in the following: first, we applied cluster analysis of time course gene expression profiles to reduce dimensionality and used the gene ontology category information to determine biologically meaningful gene modules, upon which a model of the gene regulatory module is built. This step enables us to address the scalability problem that is faced by researchers in inferring GRNs from time course gene expression data with limited time points. Second, we detected significant NMs for each transcription factor in an integrative molecular interaction network consisting of protein-protein interaction and protein-DNA interaction data (hereafter called molecular interaction data) from thirteen publically available databases. Finally, we used neural network (NN) models that mimic the topology of NMs to identify gene modules that may be regulated by a transcription factor, thereby inferring the regulatory relationships between the transcription factor and gene modules. A hybrid of genetic algorithm and particle swarm optimization (GA-PSO) methods was applied to train the NN models.

The organization of this chapter is as follows. Section 2 briefly reviews the related methods. Section 3 introduces the proposed method to infer GRNs by integrating multiple sources of biological data. Section 4 will present the results on two real data sets: the yeast cell cycle data set (Spellman et al. 1998), and the human Hela cell cycle data set (Whitfield et al. 2002). Finally, Section 5 is devoted to the summary and discussions.

2. Review of related methods

In recent years, high throughput biotechnologies have made large-scale gene expression surveys a reality. Gene expression data provide an opportunity to directly review the activities of thousands of genes simultaneously. The field of system modeling plays a significant role in inferring GRNs with these gene expression data.

Several system modeling approaches have been proposed to reverse-engineer network interactions including a variety of continuous or discrete, static or dynamic, quantitative or qualitative methods. These include biochemically driven methods (Naraghi and Neher

1997), linear models (Chen et al. 1999; D'Haeseleer et al. 1999), Boolean networks (Shmulevich et al. 2002), fuzzy logic (Woolf and Wang 2000; Ressom et al. 2003), Bayesian networks (Friedman et al. 2000), and recurrent neural networks (RNNs) (Zhang et al. 2008). Biochemically inspired models are developed on the basis of the reaction kinetics between different components of a network. However, most of the biochemically relevant reactions under participation of proteins do not follow linear reaction kinetics, and the full network of regulatory reactions is very complex and hard to unravel in a single step. Linear models attempt to solve a weight matrix that represents a series of linear combinations of the expression level of each gene as a function of other genes, which is often underdetermined since gene expression data usually have far fewer dimensions than the number of genes. In a Boolean network, the interactions between genes are modeled as Boolean function. Boolean networks assume that genes are either "on" or "off" and attempt to solve the state transitions for the system. The validity of the assumptions that genes are only in one of these two states has been questioned by a number of researchers, particularly among those in the biological community. Woolf and Wang 2000 proposed an approach based on fuzzy rules of a known activator/repressor model of gene interaction. This algorithm transforms expression values into qualitative descriptors that is evaluated by using a set of heuristic rules and searches for regulatory triplets consisting of activator, repressor, and target gene. This approach, though logical, is a brute force technique for finding gene relationships. It involves significant computational time, which restricts its practical usefulness. In (Ressom et al. 2003), we proposed the use of clustering as an interface to a fuzzy logic–based method to improve the computational efficiency. In a Bayesian network model, each gene is considered as a random variable and the edges between a pair of genes represent the conditional dependencies entailed in the network structure. Bayesian statistics are applied to find certain network structure and the corresponding model parameters that maximize the posterior probability of the structure given the data. Unfortunately, this learning task is NP-hard, and it also has the underdetermined problem. The RNN model has received considerable attention because it can capture the nonlinear and dynamic aspects of gene regulatory interactions. Several algorithms have been applied for RNN training in network inference tasks, such as fuzzy-logic (Maraziotis et al. 2005) and genetic algorithm (Chiang and Chao 2007).

3. Proposed method

3.1 Overview of the proposed framework

In the proposed framework, we consider two different layers of networks in the GRN. One is the molecular interaction network at the factor-gene binding level. The other is the functional network that incorporates the consequences of these physical interactions, such as the activation or repression of transcription. We used three types of data to reconstruct the GRN, namely protein-protein interactions derived from a collection of public databases, protein-DNA interactions from the TRANSFAC database (Matys et al. 2006), and time course gene expression profiles. The first two data sources provided direct network information to constrain the GRN model. The gene expression profiles provided an unambiguous measurement on the causal effects of the GRN model. Gene ontology annotation describes the similarities between genes within one network, which facilitates further characterization of the relationships between genes. The goal is to discern dependencies between the gene expression patterns and the physical inter-molecular interactions revealed by complementary data sources.

The framework model for GRN inference is illustrated in Figure 1. Besides data pre-processing, three successive steps are involved in this framework as outlined in the following:

Gene module selection: genes with similar expression profiles are represented by a gene module to address the scalability problem in GRN inference (Ressom et al. 2003). The assumption is that a subset of genes that are related in terms of expression (co-regulated) can be grouped together by virtue of a unifying cis-regulatory element(s) associated with a common transcription factor regulating each and every member of the cluster (co-expressed) (Yeung et al. 2004). Gene ontology information is used to define the optimal number of clusters with respect to certain broad functional categories. Since each gene module identified from clustering analysis mainly represents one broad biological process or category as evaluated by FuncAssociate (Berriz et al. 2003), the regulatory network implies that a given transcription factor is likely to be involved in the control of a group of functionally related genes (De Hoon et al. 2002). This step is implemented by the method proposed in our previous work (Zhang et al. 2009).

Network motif discovery: to reduce the complexity of the inference problem, NMs are used instead of a global GRN inference. The significant NMs in the combined molecular interaction network are first established and assigned to at least one transcription factor. These associations are further used to reconstruct the regulatory modules. This step is implemented using FANMOD software (Wernicke and Rasche 2006). We briefly describe it in the following section.

Gene regulatory module inference: for each transcription factor assigned to a NM, a NN is trained to model a GRN that mimics the associated NM. Genetic algorithm generates the candidate gene modules, and particle swarm optimization (PSO) is used to configure the parameters of the NN. Parameters are selected to minimize the root mean square error (RMSE) between the output of the NN and the target gene module's expression pattern. The RMSE is returned to GA to produce the next generation of candidate gene modules. Optimization continues until either a pre-specified maximum number of iterations are completed or a pre-specified minimum RMSE is reached. The procedure is repeated for all transcription factors. Biological knowledge from public databases is used to evaluate the predicted results. This step is the main focus of this chapter.

3.2 Network motif discovery

The NM analysis is based on network representation of protein-protein interactions and protein-DNA interactions. A node represents both the gene and its protein product. A protein-protein interaction is represented by a bi-directed edge connecting the interacting proteins. A protein-DNA interaction is an interaction between a transcription factor and its target gene and is represented by a directed edge pointing from the transcription factor to its target gene.

All connected subnetworks containing three nodes in the interaction network are collated into isomorphic patterns, and the number of times each pattern occurs is counted. If the number of occurrences is at least five and significantly higher than in randomized networks, the pattern is considered as a NM. The statistical significance test is performed by generating 1000 randomized networks and computing the fraction of randomized networks in which the pattern appears at least as often as in the interaction network, as described in (Yeger-Lotem et al. 2004). A pattern with p≤0.05 is considered statistically significant. This NM discovery procedure was performed using the FANMOD software. For different organisms, different NMs may be identified. As shown in Figure 2 and Figure 3, different sets of NMs were detected in yeast and human. Both NM sets shared some similar NM structures. For example, Figure 2(B) and Figure 3(C) were both feed-forward loops. This NM has been indentified and studied in many organisms including E. coli, yeast, and human. Knowledge of these NMs to which a given transcription factor belongs facilitates the identification of downstream target gene modules.

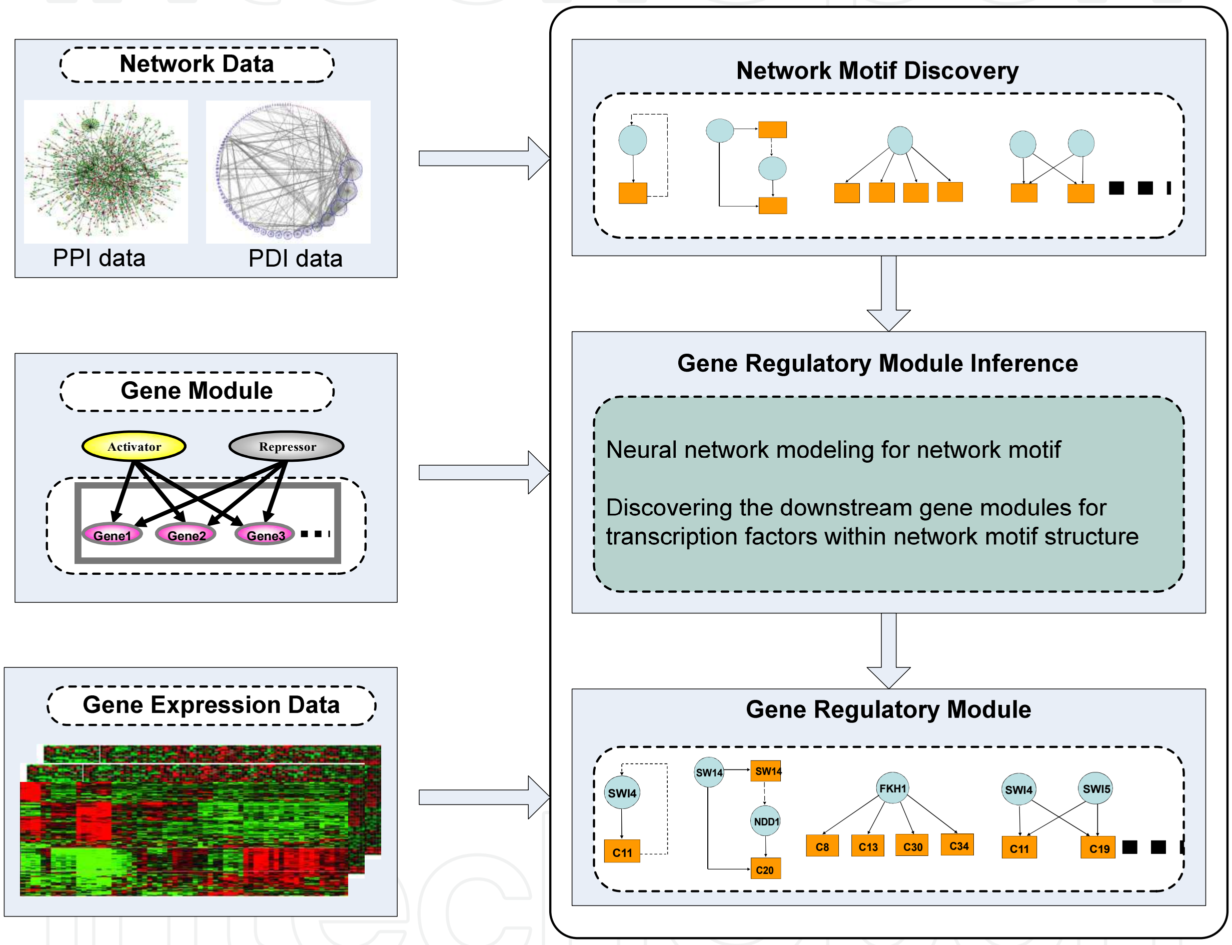


Fig. 1. Schematic overview of the computational framework used for the gene regulatory network inference.

3.3 Reverse engineering transcriptional regulatory modules

In building NNs for inferring GRNs, the identification of the correct downstream gene modules and determination of the free parameters (weights and biases) to mimic the real data is a challenging task given the limited available quantity of data. For example, in inferring a GRN from microarray data, the number of time points is considerably low compared to the number of genes involved. Considering the complexity of the biological system, it is difficult to adequately describe the pathways involving a large number of genes with few time points. We addressed this challenge by inferring GRNs at NM modular level instead of gene level. Neural network models were built for all NMs detected in the molecular interaction data. A hybrid search algorithm, called GA-PSO, was proposed to select the candidate gene modules (output node) in a NN and to update its free parameters simultaneously. We illustrate the models and training algorithm in detail in the following sections.

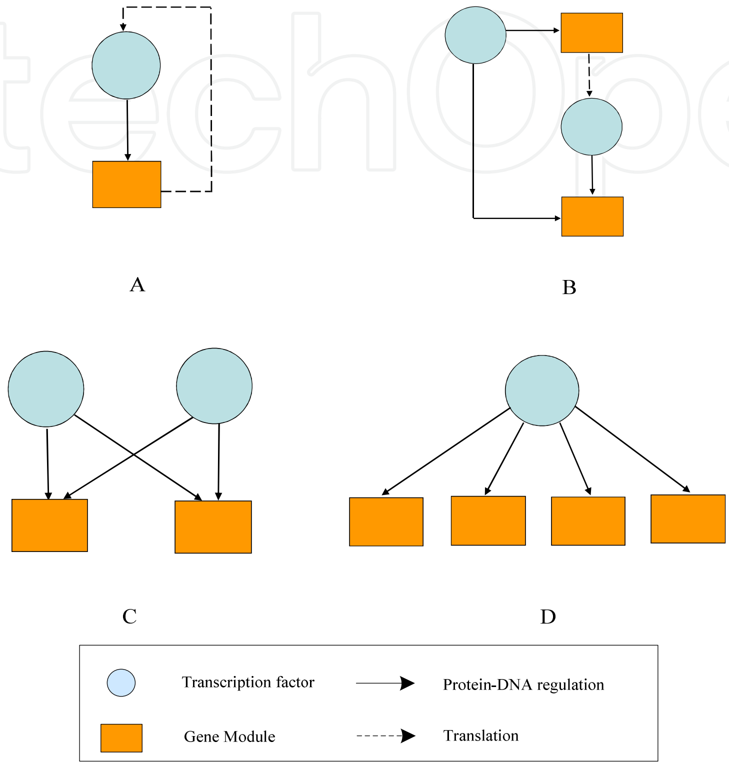


Fig. 2. Four NMs discovered in yeast: (A) auto-regulatory motif; (B) feed-forward loop; (C) single input module; and (D) multi-input module.

3.3.1 Neural network model

The neural network model is based on the assumption that the regulatory effect on the expression of a particular gene can be expressed as a neural network (Figure 4(A)), where each node represents a particular gene and the wirings between nodes define regulatory interactions. Each layer of the network represents the expression level of genes at time *t*. The output of a node at time $t + \Delta t$ is derived from the expression levels at the time *t* and the connection weights of all genes connected to the given gene. As shown in the figure, the output of each node is fed back to its input after a unit delay and is connected to other nodes. The network is used as a model to a GRN: every gene in the network is considered as a neuron; NN model considers not only the interactions between genes but also gene self-regulations.

Figure 4 (B) illustrates the details of the *i*th self-feedback neuron (e.g. *i*th gene in the GRN), where v_i, known as the induced local field (activation level), is the sum of the weighted inputs (the regulation of other genes) to the neuron (ith gene); and $\varphi()$ represents an activation function (integrated regulation of the whole NN on *i*th gene), which transforms the activation level of a neuron into an output signal (regulation result). The induced local field and the output of the neuron, respectively, are given by:

$$v_i(T_j) = \sum_{k=1}^{N} w_{ik} x_k(T_{j-1}) + b_i(T_j) \tag{1}$$

$$x_i(T_j) = \varphi(v_i(T_j)) \tag{2}$$

where the synaptic weights w_{i1}, w_{i2},…, w_{iN} define the strength of connections between the *i*th neuron (e.g. *i*th gene) and its inputs (e.g. expression level of genes). Such synaptic weights exist between all pairs of neurons in the network. $b_i(T_j)$ denotes the bias for the *i*th neuron at time T_j. We denote $\vec{w}$ as a weight vector that consists of all the synaptic weights and biases in the network. $\vec{w}$ is adapted during the learning process to yield the desired network outputs. The activation function $\varphi()$ introduces nonlinearity to the model. When information about the complexity of the underlying system is available, a suitable activation function can be chosen (e.g. linear, logistic, sigmoid, threshold, hyperbolic tangent sigmoid or Gaussian function.) If no prior information is available, our algorithm uses the hyperbolic tangent sigmoid function.

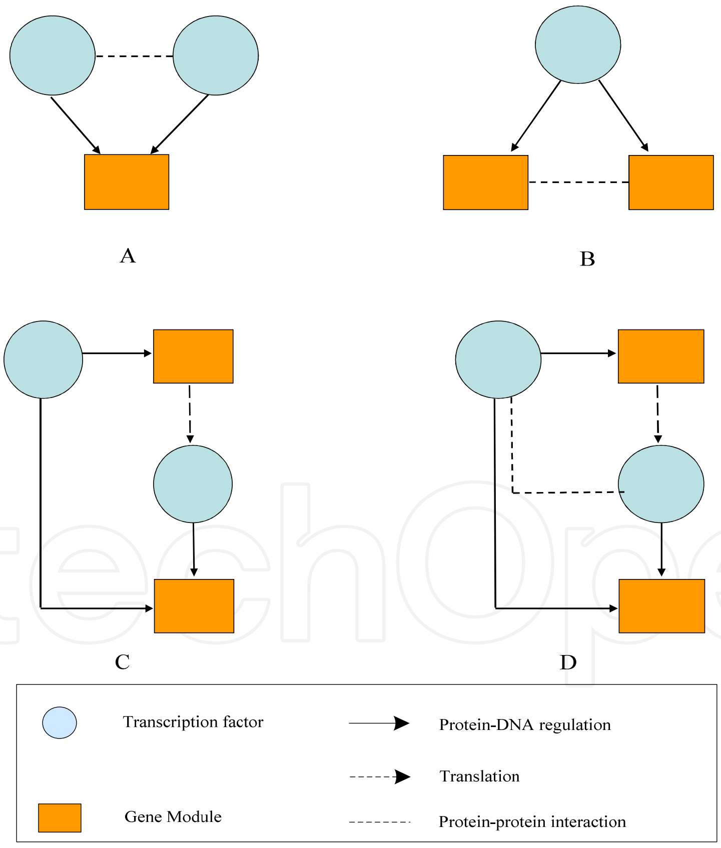


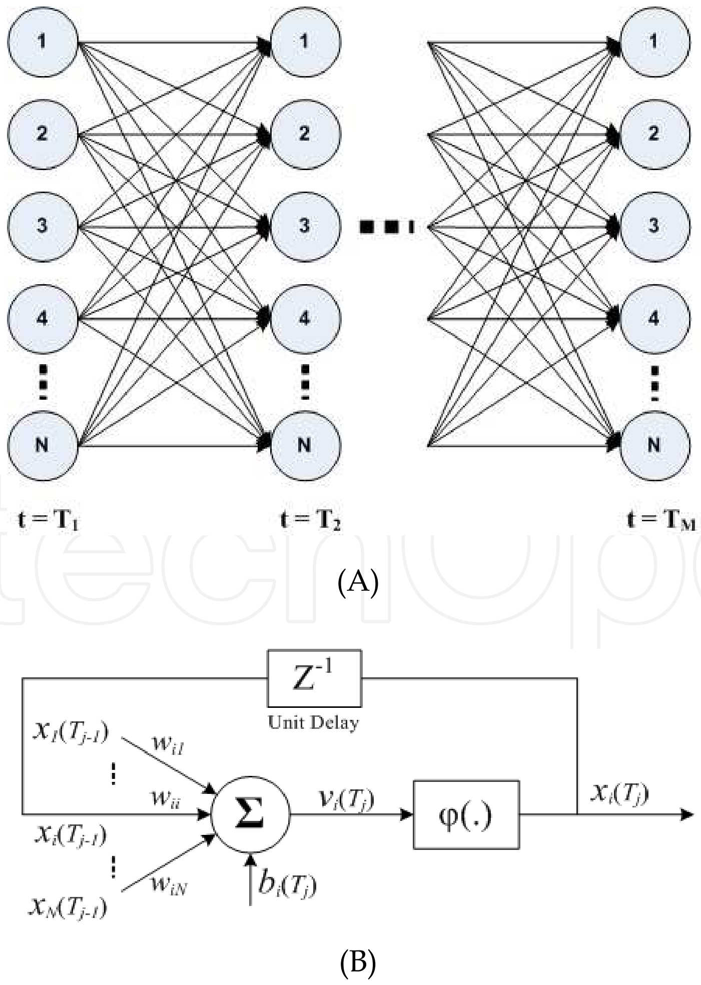
Fig. 3. Four NMs discovered in human: (A) multi-input module; (B) single input module; (C) feed-forward loop - 1; and (D) feed-forward loop - 2.

As a cost function, we use the RMSE between the expected output and the network output across time and neurons in the network. The cost function is written as:

$$E(\vec{w}) = \sqrt{\frac{1}{Nm}\sum_{t=T_i}^{T_m}\sum_{i=1}^{N}[[x_i(t) - \hat{x}_i(t)]]^2} \tag{3}$$

where $x_i(t)$ and $\hat{x}_i(t)$ are the true and predicted values (expression levels) for the *i*th neuron (gene) at time *t*. The goal is to determine weight vector $\vec{w}$ that minimize this cost function. This is a challenging task if the size of the network is large and only few samples (time points) are available.

For each NM shown in Figure 2 and 3, a corresponding NN is built. Figure 5 presents the detailed NN model for each NM in Figure 2. For auto-regulatory motif, the NN model is the same as single input module except that its downstream gene module contains the transcription factor itself. Since the expression level of gene module is the mean of expression level of its member genes, the input node and output node have different expression profiles. This avoids an open-loop problem which may cause the stability of the model. For single input and multiple input modules, instead of selecting multiple downstream gene modules simultaneously, we build NN models to find candidate gene modules one at one time. The selected gene modules are merged together to build the final NM gene regulatory modules.



(A)

(B)

Fig. 4. Architecture of a fully connected NN (A) and details of a single recurrent neuron (B).

Based on NMs to which a given transcription factor belongs, the next process is to find out the target gene modules and the relationships between them. To resolve this problem, we propose a hybrid GA-PSO training algorithm for NN model.

3.3.2 Genetic algorithm

Genetic algorithms are stochastic optimization approaches which mimic representation and variation mechanisms borrowed from biological evolution, such as selection, crossover, and mutation (Mitchell 1998). In a GA, a candidate solution is represented as a linear string analogous to a biological chromosome. The general scheme of GAs starts from a population of randomly generated candidate solutions (chromosomes). Each chromosome is then evaluated and given a value which corresponds to a fitness level in the objective function space. In each generation, chromosomes are chosen based on their fitness to reproduce offspring. Chromosomes with a high level of fitness are more likely to be retained while the ones with low fitness tend to be discarded. This process is called selection. After selection, offspring chromosomes are constructed from parent chromosomes using operators that resemble crossover and mutation mechanisms in evolutionary biology. The crossover operator, sometimes called recombination, produces new offspring chromosomes that inherit information from both sides of parents by combining partial sets of elements from them. The mutation operator randomly changes elements of a chromosome with a low probability. Over

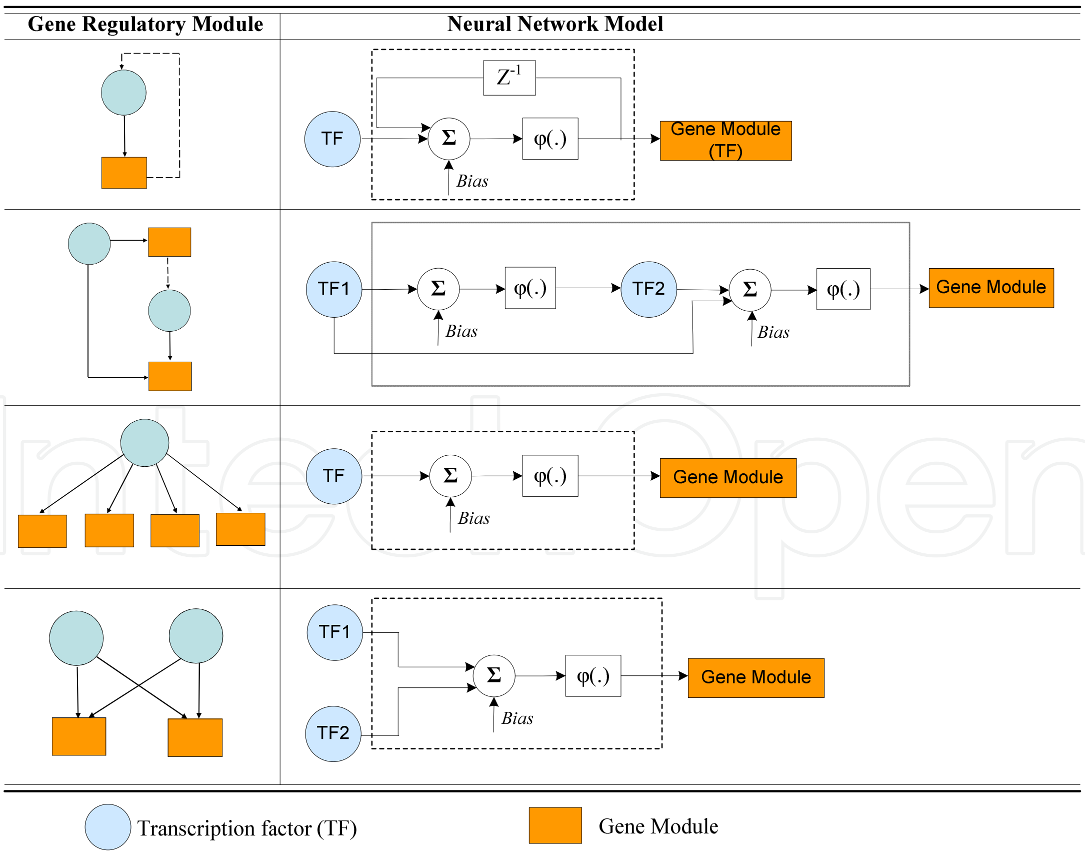


Fig. 5. NN models mimicking the topologies of the four NMs shown in Figure 2. Z^{-1} denotes a unit delay and $\varphi()$ is a logistic sigmoid activation function.

multiple generations, chromosomes with higher fitness values are left based on the survival of the fitness. A detailed description of GA is shown in Figure 6. In this chapter, we propose to apply to GA to select the best suitable downstream gene module(s) for each transcription factor and the NN model(s) that mimic its NM(s). The Genetic Algorithm and Direct Search Toolbox (Mathworks, Natick, MA) is used for implementation of GA.

3.3.3 Particle swarm optimization

After the candidate downstream gene modules are selected by GA, PSO is proposed to determine the parameters in the NN model. Particle swarm optimization is motivated by the behavior of bird flocking or fish blocking, originally intended to explore optimal or near-optimal solutions in sophisticated continuous spaces (Kennedy and Eberhart 1995). Its main difference from other evolutionary algorithms (e.g., GA) is that PSO relies on cooperation rather than competition. Good solutions in the problem set are shared with their less-fit ones so that the entire population improves.

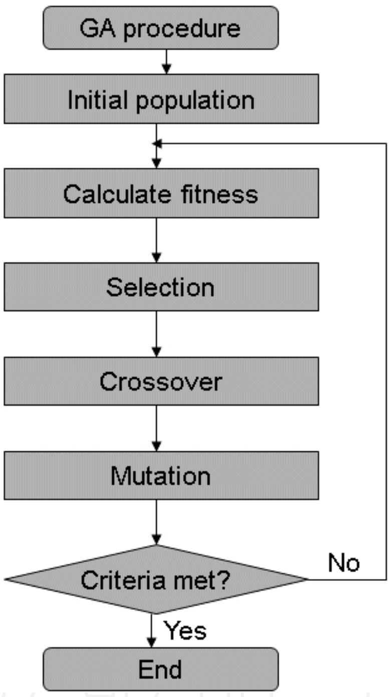


Fig. 6. A detailed description of GA.

Particle swarm optimization consists of a swarm of particles, each of which represents a candidate solution. Each particle is represented as a D-dimensional vector $\vec{w}$, with a corresponding D-dimensional instantaneous trajectory vector $\Delta\vec{w}(t)$, describing its direction of motion in the search space at iteration *t*. The index *i* refers to the *i*th particle. The core of the PSO algorithm is the position update rule (Eq. (4)) which governs the movement of each of the n particles through the search space.

$$\vec{w}_i(t+1) = \vec{w}_i(t) + \Delta\vec{w}_i(t+1) \tag{4}$$

$$\Delta\vec{w}_i(t+1) = \chi[\Delta\vec{w}_i(t) + \Phi_1(\vec{w}_{i,best}(t) - \vec{w}_i(t)) + \Phi_2(\vec{w}_{G,best}(t) - \vec{w}_i(t))] \tag{5}$$

$$\text{where} \quad \Phi_1 = c_1 \begin{bmatrix} r_{1,1} & & & \\ & r_{1,2} & & \\ & & \ddots & \\ & & & r_{1,D} \end{bmatrix} \quad \text{and} \quad \Phi_2 = c_2 \begin{bmatrix} r_{2,1} & & & \\ & r_{2,2} & & \\ & & \ddots & \\ & & & r_{2,D} \end{bmatrix}$$

At any instant, each particle is aware of its individual best position, $\vec{w}_{i,best}(t)$, as well as the best position of the entire swarm, $\vec{w}_{G,best}(t)$. The parameters *c1* and *c2* are constants that weight particle movement in the direction of the individual best positions and global best positions, respectively; and $r_{1,j}$ and $r_{2,j}$, $j = 1,2,\ldots D$ are random scalars distributed uniformly between 0 and 1, providing the main stochastic component of the PSO algorithm. Figure 7 shows a vector diagram of the contributing terms of the PSO trajectory update. The new change in position, $\Delta\vec{w}_i(t+1)$, is the resultant of three contributing vectors: (i) the inertial component, $\Delta\vec{w}_i(t)$, (ii) the movement in the direction of individual best, $\vec{w}_{i,best}(t)$, and (iii) the movement in the direction of the global (or neighborhood) best, $\vec{w}_{G,best}(t)$.

The constriction factor, χ, may also help to ensure convergence of the PSO algorithm, and is set according to the weights *c1* and *c2* as in Eq.(6).

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \varphi = c_1 + c_2, \varphi > 4 \tag{6}$$

The key strength of the PSO algorithm is the interaction among particles. The second term in Eq. (5), $\Phi_2(\vec{w}_{G,best}(t) - \vec{w}_i(t))$, is considered to be a "social influence" term. While this term tends to pull the particle towards the globally best solution, the first term, $\Phi_1(\vec{w}_{i,best}(t) - \vec{w}_i(t))$, allows each particle to think for itself. The net combination is an algorithm with excellent trade-off between total swarm convergence, and each particle's capability for global exploration. Moreover, the relative contribution of the two terms is weighted stochastically.

The algorithm consists of repeated application of the velocity and position update rules presented above. Termination occurs by specification of a minimum error criterion, maximum number of iterations, or alternately when the position change of each particle is sufficiently small as to assume that each particle has converged.

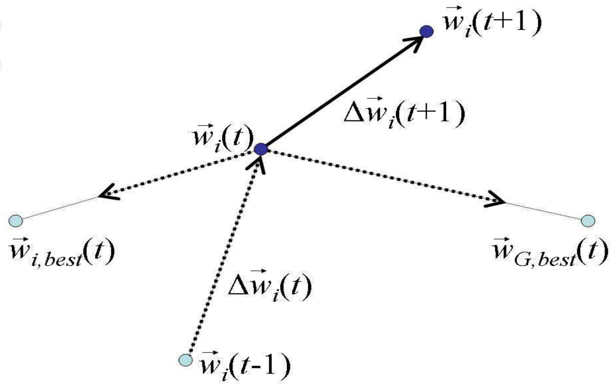


Fig. 7. Vector diagram of particle trajectory update.

A pseudo-code description of the PSO algorithm is provided below:

1. Generate initial population of particles, $\vec{w}_i$, i=1,2,…,n, distributed randomly (uniform) within the specified bounds.
2. Evaluate each particle with the objective function, $f(\vec{w}_i)$; if any particles have located new individual best positions, then replace previous individual best positions, $\vec{w}_{i,best}$, and keep track of the swarm global best position, $\vec{w}_{G,best}$.
3. Determine new trajectories, $\Delta\vec{w}_i(t+1)$, according to Eq. (5).
4. Update each particle position according to Eq. (4).
5. Determine if any $\vec{w}_i(t+1)$ are outside of the specified bounds; hold positions of particles within the specified bounds.

If termination criterion is met (for example completed maximum number of iterations), then $\vec{w}_{G,best}$ is the best solution found; otherwise, go to step (2).

Selection of appropriate values for the free parameters of PSO plays an important role in the algorithm's performance. In our study, the parameters setting is presented in Table 1, and the constriction factor χ is determined by Eq.(6). The PSOt Toolbox (Birge 2003) was used for implementation of PSO.

Parameter	Value
Maximum velocity, Vmax	2
Maximum search space range, |Wmax|	[-5,5]
Acceleration constants, c1 & c2	2.05, 2.05
Size of Swarm	20

Table 1. PSO parameter setting

3.3.4 GA-PSO training algorithm

A hybrid of GA and PSO methods (GA-PSO) is applied to determine the gene modules that may be regulated by each transcription factor. Genetic algorithm generates candidate gene modules, while the PSO algorithm determines the parameters of a given NN represented by a weight vector . The RMSE between the NN output and the measured expression profile is returned to GA as a fitness function and to guide the selection of target genes through reproduction, cross-over, and mutation over hundreds of generations. The stopping criteria are pre-specified minimum RMSE or maximum number of generations. The GA-PSO algorithm is run for each transcription factor to train a NN that has the architecture mimicking the identified known NM(s) for the transcription factor. Thus, for a given transcription factor (input), the following steps are carried out to identify its likely downstream gene modules (output) based on their NM(s):

1. Assign the NM to the transcription factor it belongs to.
2. Use the following GA-PSO algorithm to build a NN model that mimics the NM to identify the downstream gene modules.
   a. Generate combinations of M gene modules to represent the target genes that may be regulated by the transcription factor. Each combination is a vector/chromosome. The initial set of combinations is composed of the initial population of chromosomes.
   b. Use the PSO algorithm to train a NN model for each chromosome, where the input is the transcription factor and the outputs are gene modules. The goal is to

determine the optimized parameters of the NN that maps the measured expression profiles of the transcription factor to the gene modules.

  c. For each chromosome, calculate the RMSE between the predicted output of the NN and measured expression profiles for the target gene modules.
  d. Apply GA operators (reproduction, cross-over, mutation) based on the RMSE calculated in Step 2.3 as a fitness value. This will generate new vectors/chromosomes altering the choice of output gene module combinations.
  e. Repeat steps 2.1 – 2.4 until stop criteria are met. The stopping criteria are numbers of generations or minimum RMSE, depending on which one is met first.
  f. Repeat Steps 2.1 – 2.5 for each NM the transcription factor is assigned to.

3. Repeat Steps 1 and 2 for each transcription factor.

When the process is completed, NM regulatory modules are constructed between transcription factors and their regulated gene modules.

4. Results

4.1 Yeast cell cycle dataset

4.1.1 Data sources

Gene expression dataset: the yeast cell cycle dataset presented in (Spellman et al. 1998) consists of six time series (*cln3*, *clb2*, *alpha*, *cdc15*, *cdc28*, and *elu*) expression measurements of the mRNA levels of *S. cerevisiae* genes. 800 genes were identified as cell cycle regulated based on cluster analysis in (Spellman et al. 1998). We used the *cdc15* time course data of the 800 genes since it has the largest number of time points (24). Missing values in the data were imputed using KNN imputation (Troyanskaya et al. 2001). The expression pattern of each gene was standardized between 0 and 1.

Molecular interaction data: data of transcription factors and their target genes were extracted from the SCPD database (Zhu and Zhang 1999), from the YPD database (Costanzo et al. 2001), and from recent publications on genome-wide experiments that locate binding sites of given transcription factors (Ren et al. 2000; Iyer et al. 2001; Simon et al. 2001; Lee et al. 2002). For data extraction from the latter we used the same experimental thresholds as in the original papers. Protein-protein interaction data was extracted from the DIP database (Przulj et al. 2004), from the BIND database (Bader et al. 2003), and from the MIPS database (Mewes et al. 2002). In total the molecular interaction dataset consisted of 8184 protein pairs connected by protein-protein interactions and 5976 protein pairs connected by protein-DNA interactions.

4.1.2 Gene module Identification

We grouped 800 cell cycle-regulated genes into clusters by Fuzzy c-means method, where genes with similar expression profiles are represented by a gene module. The optimal cluster number was determined by the proposed method in (Zhang et al. 2009). The highest *z* score was obtained when the number of clusters was 34 by Fuzzy c-means clustering with optimal parameter m = 1.1573. We evaluated the resulting clusters through the gene set enrichement analysis method. All clusters except 10, 18, 21, 22, 25 and 26 are enriched in some gene ontology categories (data not shown). We used these clusters as candidate gene modules in our subsequent analyses to reduce the search space for gene regulatory module inference.

4.1.3 Network motif discovery

Among the 800 cell cycle related genes, 85 have been identified as DNA-binding transcription factors (Zhang et al. 2008). Four NMs were considered significant: auto-regulatory motif, feed-forward loop, single input module, and multi-input module (Figure 2). These NMs were used to build NN models for corresponding transcription factors.

4.1.4 Reverse engineering gene regulatory networks

Neural network models that mimic the topology of the NMs were constructed to identify the relationships between transcription factors and putative gene modules. The NN models

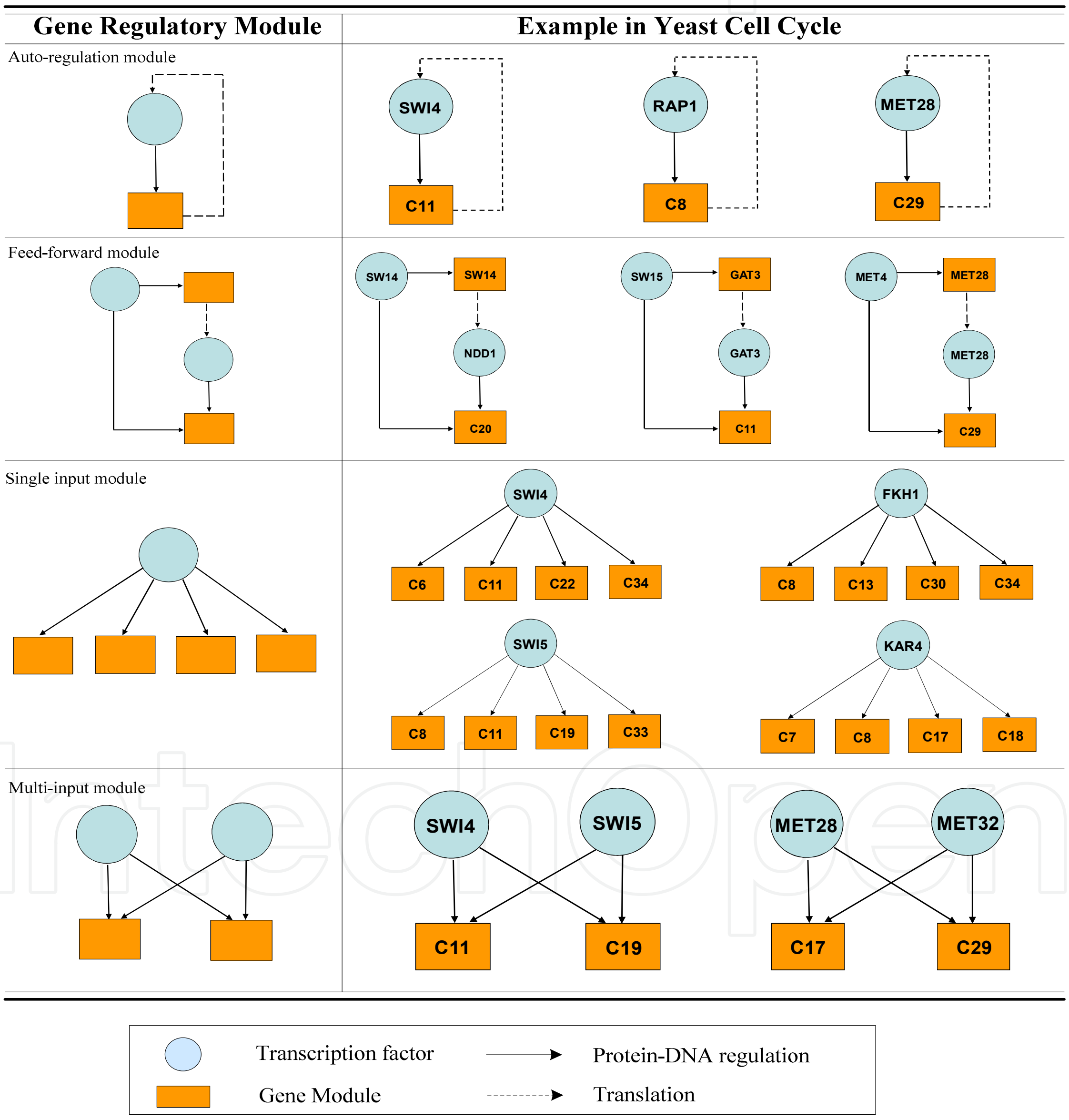


Fig. 8. Predicted gene regulatory modules from eight known cell cycle dependent transcription factors in yeast cell cycle dataset. The left panel presents the four gene regulatory modules, and the right panel depicts inferred gene regulatory modules for eight known cell cycle dependent transcription factors.

were trained to select for all 85 transcription factors the downstream targets from the 34 gene modules. The parameter settings of GA-PSO algorithm are set as described in our previous work (Ressom et al. 2006). Our inference method mapped all 85 transcription factors to the target gene modules and inferred the most likely NMs.

We evaluated the predicted gene regulatory modules for the following eight well known cell cycle related transcription factors: *SWI4*, *SWI5*, *FKH1*, *NDD1*, *ACE2*, *KAR4*, *MET28* and *RAP1*. Since the "true" GRN is not available, the accuracy of putative regulatory relationship was determined by searching known gene connections in databases. Based on the results of the NM module prediction, we collected literature evidences from SGD (Cherry et al. 1997) and BIND (Bader et al. 2003) databases. We examined the inferred relationships for each of the eight transcription factors. An inferred relationship is assumed to be biologically significant if the transcription factors are correlated with the biological functions associated with the critical downstream cluster(s). Figure 8 lists the significant relationships; the eight transcription factors yielded an average precision of 82.9%. NMs for four of these transcription factors were identified in Chiang et al. (Chiang and Chao 2007) together with other four transcription factors. The eight transcription factors in Chiang et al. yielded an average precision of 80.1%.

The regulatory relationships inferred by the proposed method are expected to correspond more closely to biologically meaningful regulatory systems and naturally lead themselves to optimum experimental design methods. The results presented in Figure 8 are verified from previous biological evidences. For example, *FKH1* is a gene whose protein product is a fork head family protein with a role in the expression of G2/M phase genes. It negatively regulates transcriptional elongation, and regulates donor preference during switching. To further investigate the possibilities that the predicted downstream gene clusters are truly regulated by *FKH1*, we applied the motif discovery tool, WebMOTIFS (Romer et al. 2007) to find shared motifs in these gene clusters (data not shown). The results revealed that a motif called Fork_head, GTAAACAA, is identified as the most significant motif among these gene clusters (Weigel and Jackle 1990). This finding strongly supports our NM inference results. Another example is the FFL involving *SWI5*, *GAT3* and Gene Cluster 10. *SWI5* has been identified as the upstream regulator of *GAT3* (Simon et al. 2001; Lee et al. 2002; Harbison et al. 2004). Genes in cluster 10 are mostly involved in DNA helicase activity and mitotic recombination, both of which are important biological steps in the regulation of cell cycle. Although no biological evidences have shown that *SWI5* and *GAT3* are involved in these processes, there are significant numbers of genes in cluster 10 which are characterized (according to yeastract.com) as genes regulated by both transcription factors (24 for *GAT3* and 23 for *SWI5* out of 44 genes in cluster 10, respectively).

4.2 Human hela cell cycle dataset

4.2.1 Data sources

Gene expression dataset: the human Hela cell cycle dataset (Whitfield et al. 2002) consists of five time courses (114 total arrays). RNA samples were collected for points (typically every 1-2 h) for 30 h (Thy- Thy1), 44 h (Thy-Thy2), 46 h (Thy-Thy3), 36 h (Thy-Noc), or 14 h (shake) after the synchronous arrest. The cell-cycle related gene set contains 1,134 clones corresponding to 874 UNIGENE clusters (UNIGENE build 143). Of these, 1,072 have corresponding Entrez gene IDs, among which 226 have more than one mapping to clones. In total, 846 genes were used for GRN inference. We chose the Thy-Thy3 time course gene expression pattern for 846 genes, since it has the largest number of time points (47). Missing values in the data were imputed using KNN imputation (Troyanskaya et al. 2001). The expression pattern of each gene was standardized between 0 and 1.

Molecular interaction data: Protein-protein interactions were extracted from twelve publicly available large-scale protein interaction maps, seven of which are based on information from scientific literature literature-based, three on orthology information, and two on results of

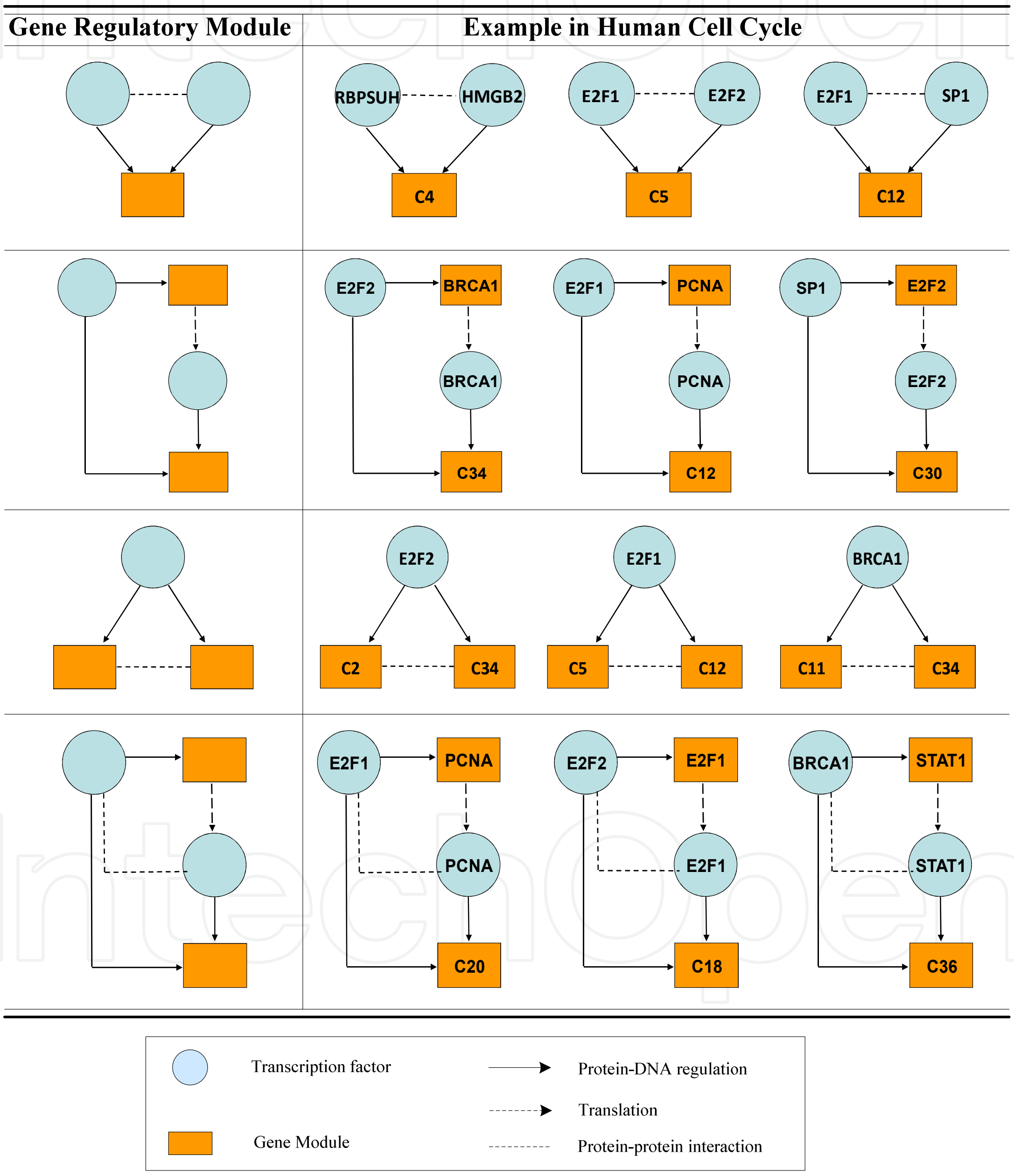


Fig. 9. Predicted gene regulatory modules from known human cell cycle dependent genes. The left panel presents the four gene regulatory modules, and the right panel depicts inferred transcription factor-target gene relationships for eight cell cycle dependent transcription factors.

previous yeast two-hybrid (Y2H) analyses (Zhang et al. 2010). The analysis was restricted to binary interactions in order to make consistent between Y2H-based interactions and the remaining maps. To merge twelve interaction maps into one combination map, all proteins were mapped to their corresponding Entrez gene IDs. The protein-DNA interaction data was extracted from the TRANSFAC database (Wingender et al. 2001). In total the molecular interaction data consisted of 20,473 protein pairs connected by protein-protein interactions and 2,546 protein pairs connected by protein-DNA interactions.

4.2.2 Gene module identification

A total of 846 genes associated with the control of cell cycle have been identified previously in Hela cells. We further partitioned these genes into more specific functional groups by Fuzzy c-means. The optimal value of m for the dataset used in this study was 1.1548. The highest *z* score was obtained with 39 clusters, indicating an optimal condition to reduce the search space for GRN inference. To evaluate the optimal clusters selected based on gene ontology, gene set enrichment analysis was applied using the optimal value. The total set of genes involved in cell cycle regulation was further subdivided into 39 clusters. Of these clusters, 31 were clearly associated with gene ontology categories that imply a more specific function that unifies the members of one but not other clusters, thereby establishing more direct relationships among certain smaller sub-groups of genes. For example, clusters 8 and 29 are both associated with pre-mitotic, mitotic and post-mitotic events (M-phase). However, members of cluster 8 are distinguished from the members of cluster 29 by virtue of their specific roles in chromosome doubling (DNA replication) and cytokinesis. Conversely, members of cluster 29 are distinguished from the members of cluster 8 by virtue of their specific roles in spindle fiber assembly and disassembly.

Biological significance of these highly specific functional relationships, established by our clustering scheme, can further be extended in terms of relationships within the regulatory context. For instance, members of both gene modules 8 and 29 have been identified previously as direct downstream targets of *E2F* factors (Ren et al. 2002). Similar relationships are established with other clusters such as gene module 32, which is comprised of genes with biochemical roles of a DNA ligase. Thus, the genes in gene module 32 are involved in processes associated with gap repair or Okazaki fragment processing during DNA replication and chromosome doubling. Previous studies have established that genes associated with this function are under the regulatory control of *E2F1* and *PCNA* (Shibutani et al. 2008).

Based on all these relationships, we demonstrated that one specific strength of the proposed method is its ability to distinguish genes that are related by function in a broad sense and sub-categorize them into highly specific (narrow) functional categories, resulting in the prediction of regulatory relationships that are consistent with biologically validated relationships.

4.2.3 Network motif discovery

All genes with either direct or indirect roles in the regulation of transcription were first identified from the total set of 846 cell cycle associated genes according to gene ontology categories that denote possible roles in transcription (Ashburner et al. 2000). Candidate genes that remained after filtering other gene function categories are those that are assigned to the following putative functions: transcription factor activity (GO: 0003700), regulation of transcription (GO: 0061019), and transcription factor complex (GO: 0005667). Since gene ontology information alone may not be sufficient to identify the genes with bona fide roles as transcription factors, we further filtered our list of candidate transcription factors by adding another layer of confirmatory information based on the results of PubMed database searches. This additional annotation allowed us to validate the gene ontology classification of our candidate genes. Among the 846 cell cycle related genes, 46 were annotated with functions related to transcriptional regulation based on both gene ontology and PubMed databases. These genes were considered as putative transcription factors (Zhang et al. 2010).

In the microarray gene expression data, genes are often represented by multiple oligonucleotide probes. Genes represented by probe sets with larger variance were further considered in this study. We decomposed the collected human molecular interaction network into several NMs, with each NM potentially associated with a given transcription factor(s). A total of four NMs were found to be significant in the combined molecular interaction network (Figure 3), thus each transcription factor was assigned to at least one of these NMs.

4.2.4 Reverse engineering gene regulatory networks

The relationships between transcription factors and gene modules were determined based on NN models. For each of the four NMs (Figure 3), a suitable NN was built as we previously described (Zhang et al. 2008). The NN models were trained using the GA-PSO algorithm to find the downstream gene clusters for all 46 putative transcription factors. Associations between each transcription factor and 39 gene modules was determined by training the NN model that mimics the specific NM for a given transcription factor. Due to a reduction in the computational complexity (mapping between 46 transcription factors and 39 gene clusters instead of 846 genes), the numbers of GA and PSO generations needed to reach the pre-specified minimum RMSE was significantly reduced. The proposed inference method successfully assigned all 46 putative transcription factors to their target gene modules and inferred the most likely gene regulatory modules (see Figure 9 for representative gene regulatory modules).

The validity and accuracy of the network depicted by the gene regulatory modules are assessed by comparison with a network model constructed based on actual biological data. In the absence of such information, we performed an initial validation of the network by searching for known gene connections in databases. Based on the prediction results, we collected literature evidence from the NCBI and TRANSFAC databases. We reviewed each predicted NM and examined the relationships between the transcription factor and its target gene module(s). Subsequent analysis was performed under the basic assumption that the inferred NM is more likely to be biologically meaningful if the transcription factors therein are correlated with the enriched biological functions in the downstream clusters.

Significant NMs resulting from the survey of available literature cell cycle dependent genes such as *E2F1*, *E2F2*, *SP1*, *BRCA1*, *STAT1*, *PCNA*, *RBPSUH*, and *HMGB2* are listed in Figure 9. Based on the combined information, the biological implication of the network is further explained. For instance, *E2F* is a transcription factor that plays a crucial role in cell-cycle progression in mammalian cells (Takahashi et al. 2000). *E2F1*, which contains two overlapping *E2F*-binding sites in its promoter region, is activated at the G1/S transition in an *E2F*-dependent manner. *E2F2* interacts with certain elements in the *E2F1* promoter and both genes are involved in DNA replication and repair (Ishida et al. 2001), cytokinesis, and tumor development (Zhu et al. 2001). According to the gene set enrichment analysis results, gene module 8 is enriched with genes involved in mitosis and cytokinesis, and gene module 34 is enriched with genes involved in several functional categories associated with tumor development. As shown in Figure 9, both gene module 8 and 34 are predicted to be regulated by *E2F1* and *E2F2*, and these results are in agreement with previous reports based on biological data.

Our analysis predicts that *E2F1* and *PCNA* are components of the same network. Both of these genes are involved in the regulation of gene modules 32 and 34. The best understood molecular function of the *PCNA* protein is its role in the regulation of eukaryotic DNA polymerase delta processivity, which ensures the fidelity of DNA synthesis and repair (Essers et al. 2005). However, recent studies have provided evidence that the *PCNA* protein also functions as a direct repressor of the transcriptional coactivator p300 (Hong and Chakravarti 2003). Another study shows that *PCNA* represses the transcriptional activity of retinoic acid receptors (RARs) (Martin et al. 2005). Thus, the involvement of these genes in the same network, as predicted by our network inference algorithm, is strongly supported by knowledge of regulatory relationships already established in experimental data. The results of our prediction are in agreement with these reports since both gene modules 8 and 32 are enriched with genes involved in DNA synthesis and regulatory processes.

We proposed three approaches to investigate further whether the genes predicted to be regulated by *E2F* genes in gene modules 8, 32 and 34 are validated in classical non-genome wide methods. First, we investigated how many "known" *E2F1* and *E2F2* targets are predicted by our proposed method. According to Bracken et al. (Bracken et al. 2004), 130 genes were reviewed as *E2F* targets, 44 of which were originally identified by classical, non-genome-wide approaches. Since we restricted our analysis to the 846 cell cycle related genes, 45 genes matched the *E2F* target genes listed in Bracken et al., 21 of which were known from studies using classical molecular biology analyses. The gene targets predicted by our method match 15 of 45 genes, all 15 of which are among those found originally using standard molecular biology experiments. One possible reason is that genome-wide approaches are usually highly noisy and inconsistent across different studies.

Second, we wanted to see whether our predicted gene target clusters are enriched in the corresponding binding sites for the transcription factors in their upstream region. For both *E2F1* and *E2F2*, 7 out of 17 genes in gene module 8 contain binding sites in their upstream regions as confirmed by data in the SABiosciences database[1].

Finally, we determined how many genes in the gene clusters have *E2F* binding sites. We applied WebMOTIFS to find shared motifs in the gene modules predicted to the *E2F* targets using binding site enrichment analysis. The results revealed that a motif called E2F_TDP, GCGSSAAA, is identified as the most significant motif amon[1]g gene modules 2, 8, 29, 31, 32 and 34. Unfortunately, for gene modules 30 and 36 the number of genes in these clusters is

[1] http://www.sabiosciences.com/

too small for WebMOTIFS analysis. All these gene modules are predicted to the downstream targets of E2F. For instance, 43 out of 52 genes in gene module 2 have putative *E2F* binding sites in their upstream regions. The detailed information of binding site enrichment analysis results is shown in Table 2. For those gene regulatory networks where two transcription factors are involved in, the downstream gene modules are found to be enriched in both the binding site sequence motifs. For instance, gene module 32 is enriched in both E2F_TDP and MH1 motifs, corresponding to the two transcription factors in the gene regulatory module: *E2F1* and *SP1*. These binding site enrichment analysis results strongly support our inference results.

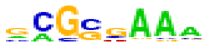
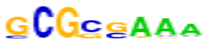
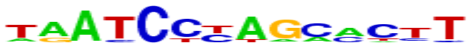
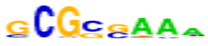
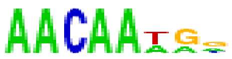
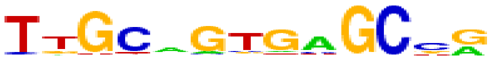
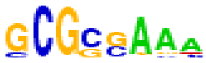
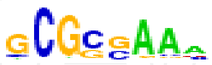
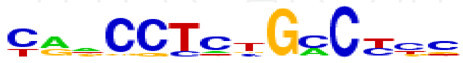
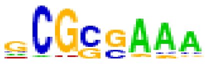
Cluster #	Sequence logo[a]	Binding domain (Pfam ID)	Corresponding transcription factor	Conserved binding motif[b]
Cluster 2		E2F_TDP (PF02319)	E2F1 E2F2	GCGssAAa
Cluster 8		E2F_TDP (PF02319)	E2F1 E2F2	GCGssAAa
Cluster 29		zf-C4 (PF00105)	BRCA1	TGACCTTTG ACCyy
		E2F_TDP (PF02319)	E2F1 E2F2	GCGssAAa
Cluster 31		HMG_box (PF00505)	HMGB2	AACAAwRr
Cluster 32		MH1 (PF03165)	SP1	TGGc.....gCCA
		E2F_TDP (PF02319)	E2F1 E2F2	GCGssAAa
Cluster 34		E2F_TDP (PF02319)	E2F1 E2F2	GCGssAAa
Cluster 38		zf-C4 (PF00105)	BRCA1	TGACCTTTG ACCyy
		E2F_TDP (PF02319)	E2F1 E2F2	GCGssAAa

Table 2. Binding site enrichment analysis for gene modules identified in human Hela cell cycle dataset. [a]Sequence logos represent the motif significantly overrepresented in individual gene cluster associated with their predicted upstream transcription factors, according to the WebMOTIFS discovery algorithm. Individual base letter height indicates level of conservation within each binding site position; [b]Conserved binding motifs are the conserved binding sequences used in the WebMOTIFS discovery algorithm.

5. Summary and discussions

Reverse engineering GRNs is one of the major challenges in the post-genomics era of biology. In this chapter, we focused on two broad issues in GRN inference: (1) development of an analysis method that uses multiple types of data and (2) network analysis at the NM modular level. Based on the information available nowadays, we proposed a data integration approach that effectively infers the gene networks underlying certain patterns of gene co-regulation in yeast cell cycle and human Hela cell cycling. The predictive strength of this strategy is based on the combined constraints arising from multiple biological data sources including time course gene expression data, combined molecular interaction network data, and gene ontology category information.

This computational framework allows us to fully exploit the partial constraints that can be inferred from each data source. First, to reduce the inference dimensionalities, the genes were grouped into clusters by Fuzzy c-means, where the optimal fuzziness value was determined by statistical properties of gene expression data. The optimal cluster number was identified by integrating gene ontology category information. Second, the NM information established from the combined molecular interaction network was used to assign NM(s) to a given transcription factor. Once the NM(s) for a transcription factor was identified, a hybrid GA-PSO algorithm was applied to search for target gene modules that may be regulated by that particular transcription factor. This search was guided by the successful training of a NN model that mimics the regulatory NM(s) assigned to the transcription factor. The effectiveness of this method was illustrated via well-studied cell cycle dependent transcription factors (Figure 3.10 and 3.11). The upstream BINDING SITE ENRICHMENT ANALYSIS indicated that the proposed method has the potential to identify the underlying regulatory relationships between transcription factors and their downstream genes at the modular level. This demonstrates that our approach can serve as a method for analyzing multi-source data at the modular level.

Compared to the approach developed in [148], our proposed method has several advantages. First, our method performs the inference of GRNs from genome-wide expression data together with other biological knowledge. It has been shown that mRNA expression data alone cannot reflect all the activities in one GRN. Additional information will help constrain the search space of causal relationships between transcription factors and their downstream genes. Second, we decompose the GRN into well characterized functional units - NMs. Each transcription factor is assigned to specific NM(s), which is further used to infer the downstream target genes. We not only reduce the search space in the inference process, but also provide experimental biologists the regulatory modules for straightforward validation, instead of one whole GRN containing thousands of genes and connections as is often generated by IPA. Third, we group the genes into functional groups that are potentially regulated by one common transcription factor. The proposed approach reduces the noise in mRNA expression data by incorporating gene functional annotations.

In summary, we demonstrate that our method can accurately infer the underlying relationships between transcription factor and the downstream target genes by integrating multi-sources of biological data. As the first attempt to integrate many different types of data, we believe that the proposed framework will improve data analysis, particularly as more data sets become available. Our method could also be beneficial to biologists by predicting the components of the GRN in which their candidate gene is involved, followed by designing a more streamlined experiment for biological validation.

6. References

Alon, U. (2007). "Network motifs: theory and experimental approaches." Nat Rev Genet 8(6): 450-461.

Ashburner, M., C. A. Ball, et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." Nat Genet 25(1): 25-29.

Bader, G. D., D. Betel, et al. (2003). "BIND: the Biomolecular Interaction Network Database." Nucleic Acids Res 31(1): 248-250.

Berriz, G. F., O. D. King, et al. (2003). "Characterizing gene sets with FuncAssociate." Bioinformatics 19(18): 2502-2504.

Birge, B. (2003). PSOt - a particle swarm optimization toolbox for use with Matlab. Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE.

Blais, A. and B. D. Dynlacht (2005). "Constructing transcriptional regulatory networks." Genes Dev 19(13): 1499-1511.

Boyer, L. A., T. I. Lee, et al. (2005). "Core transcriptional regulatory circuitry in human embryonic stem cells." Cell 122(6): 947-956.

Bracken, A. P., M. Ciro, et al. (2004). "E2F target genes: unraveling the biology." Trends Biochem Sci 29(8): 409-417.

Bubitzky, W., M. Granzow, et al. (2007). Fundamentals of Data Mining in Genomics and Proteomics. New York, Springer.

Chen, T., H. L. He, et al. (1999). "Modeling gene expression with differential equations." Pac Symp Biocomput: 29-40.

Cherry, J. M., C. Ball, et al. (1997). "Genetic and physical maps of Saccharomyces cerevisiae." Nature 387(6632 Suppl): 67-73.

Chiang, J. H. and S. Y. Chao (2007). "Modeling human cancer-related regulatory modules by GA-RNN hybrid algorithms." BMC Bioinformatics 8: 91.

Clarke, R., H. W. Ressom, et al. (2008). "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data." Nat Rev Cancer 8(1): 37-49.

Costanzo, M. C., M. E. Crawford, et al. (2001). "YPD, PombePD and WormPD: model organism volumes of the BioKnowledge library, an integrated resource for protein information." Nucleic Acids Res 29(1): 75-79.

D'Haeseleer, P., X. Wen, et al. (1999). "Linear modeling of mRNA expression levels during CNS development and injury." Pac Symp Biocomput: 41-52.

De Hoon, M. J., S. Imoto, et al. (2002). "Statistical analysis of a small set of time-ordered gene expression data using linear splines." Bioinformatics 18(11): 1477-1485.

Essers, J., A. F. Theil, et al. (2005). "Nuclear dynamics of PCNA in DNA replication and repair." Mol Cell Biol 25(21): 9350-9359.

Friedman, N., M. Linial, et al. (2000). "Using Bayesian networks to analyze expression data." J Comput Biol. 7: 601-620.

Harbison, C. T., D. B. Gordon, et al. (2004). "Transcriptional regulatory code of a eukaryotic genome." Nature 431(7004): 99-104.

Hartemink, A. J., D. K. Gifford, et al. (2002). "Combining location and expression data for principled discovery of genetic regulatory network models." Pac Symp Biocomput: 437-449.

He, F., R. Balling, et al. (2009). "Reverse engineering and verification of gene networks: principles, assumptions, and limitations of present methods and future perspectives." J Biotechnol 144(3): 190-203.

Holmes, I. and W. J. Bruno (2001). "Evolutionary HMMs: a Bayesian approach to multiple alignment." Bioinformatics 17(9): 803-820.

Hong, R. and D. Chakravarti (2003). "The human proliferating Cell nuclear antigen regulates transcriptional coactivator p300 activity and promotes transcriptional repression." J Biol Chem 278(45): 44505-44513.

Iranfar, N., D. Fuller, et al. (2006). "Transcriptional regulation of post-aggregation genes in Dictyostelium by a feed-forward loop involving GBF and LagC." Dev Biol 290(2): 460-469.

Ishida, S., E. Huang, et al. (2001). "Role for E2F in control of both DNA replication and mitotic functions as revealed from DNA microarray analysis." Mol Cell Biol 21(14): 4684-4699.

Iyer, V. R., C. E. Horak, et al. (2001). "Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF." Nature 409(6819): 533-538.

Kennedy, J. and R. C. Eberhart (1995). "Particle swarm optimization." Proceedings of the 1995 IEEE International Conference on Neural Networks (Perth, Australia) IV: 1942-1948.

Lee, T. I., N. J. Rinaldi, et al. (2002). "Transcriptional regulatory networks in Saccharomyces cerevisiae." Science 298(5594): 799-804.

Mangan, S. and U. Alon (2003). "Structure and function of the feed-forward loop network motif." Proc Natl Acad Sci U S A 100(21): 11980-11985.

Mangan, S., A. Zaslaver, et al. (2003). "The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks." J Mol Biol 334(2): 197-204.

Maraziotis, I., A. Dragomir, et al. (2005). "Gene networks inference from expression data using a recurrent neuro-fuzzy approach." Conf Proc IEEE Eng Med Biol Soc 5: 4834-4837.

Martin, P. J., V. Lardeux, et al. (2005). "The proliferating cell nuclear antigen regulates retinoic acid receptor transcriptional activity through direct protein-protein interaction." Nucleic Acids Res 33(13): 4311-4321.

Matys, V., O. V. Kel-Margoulis, et al. (2006). "TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes." Nucleic Acids Res(34 Database): D108 - 110.

Mewes, H. W., D. Frishman, et al. (2002). "MIPS: a database for genomes and protein sequences." Nucleic Acids Res 30(1): 31-34.

Milo, R., S. Itzkovitz, et al. (2004). "Superfamilies of evolved and designed networks." Science 303(5663): 1538-1542.

Milo, R., S. Shen-Orr, et al. (2002). "Network motifs: simple building blocks of complex networks." Science 298(5594): 824-827.

Mitchell, M. (1998). An introduction to genetic algorithm, MIT Press.

Naraghi, M. and E. Neher (1997). "Linearized buffered Ca2+ diffusion in microdomains and its implications for calculation of [Ca2+] at the mouth of a calcium channel." J Neurosci 17(18): 6961-6973.

Odom, D. T., N. Zizlsperger, et al. (2004). "Control of pancreas and liver gene expression by HNF transcription factors." Science 303(5662): 1378-1381.

Przulj, N., D. A. Wigle, et al. (2004). "Functional topology in a network of protein interactions." Bioinformatics 20(3): 340-348.

Ren, B., H. Cam, et al. (2002). "E2F integrates cell cycle progression with DNA repair, replication, and G(2)/M checkpoints." Genes Dev 16(2): 245-256.

Ren, B., F. Robert, et al. (2000). "Genome-wide location and function of DNA binding proteins." Science 290(5500): 2306-2309.

Ressom, H., R. Reynolds, et al. (2003). "Increasing the efficiency of fuzzy logic-based gene expression data analysis." Physiol Genomics 13(2): 107-117.

Ressom, H. W., Y. Zhang, et al. (2006). "Inference of gene regulatory networks from time course gene expression data using neural networks and swarm intelligence." Proceedings of the 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, Toronto, ON: 435-442.

Ressom, H. W., Y. Zhang, et al. (2006). Inferring network interactions using recurrent neural networks and swarm intelligence. Conf Proc IEEE Eng Med Biol Soc (EMBC 2006), New York City, New York, USA.

Romer, K. A., G. R. Kayombya, et al. (2007). "WebMOTIFS: automated discovery, filtering and scoring of DNA sequence motifs using multiple programs and Bayesian approaches." Nucleic Acids Res 35(Web Server issue): W217-220.

Rual, J. F., K. Venkatesan, et al. (2005). "Towards a proteome-scale map of the human protein-protein interaction network." Nature 437(7062): 1173-1178.

Saddic, L. A., B. Huvermann, et al. (2006). "The LEAFY target LMI1 is a meristem identity regulator and acts together with LEAFY to regulate expression of CAULIFLOWER." Development 133(9): 1673-1682.

Segal, E., B. Taskar, et al. (2001). "Rich probabilistic models for gene expression." Bioinformatics 17 Suppl 1: S243-252.

Shen-Orr, S. S., R. Milo, et al. (2002). "Network motifs in the transcriptional regulation network of Escherichia coli." Nat Genet 31(1): 64-68.

Shibutani, S. T., A. F. de la Cruz, et al. (2008). "Intrinsic negative cell cycle regulation provided by PIP box- and Cul4Cdt2-mediated destruction of E2f1 during S phase." Dev Cell 15(6): 890-900.

Shmulevich, I., E. R. Dougherty, et al. (2002). "Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks." Bioinformatics 18(2): 261-274.

Simon, I., J. Barnett, et al. (2001). "Serial regulation of transcriptional regulators in the yeast cell cycle." Cell 106(6): 697-708.

Spellman, P. T., G. Sherlock, et al. (1998). "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization." Mol Biol Cell 9(12): 3273-3297.

Swiers, G., R. Patient, et al. (2006). "Genetic regulatory networks programming hematopoietic stem cells and erythroid lineage specification." Dev Biol 294(2): 525-540.

Takahashi, Y., J. B. Rayman, et al. (2000). "Analysis of promoter binding by the E2F and pRB families in vivo: distinct E2F proteins mediate activation and repression." Genes Dev 14(7): 804-816.

Tavazoie, S., J. D. Hughes, et al. (1999). "Systematic determination of genetic network architecture." Nat Genet 22(3): 281-285.

Tegner, J., M. K. Yeung, et al. (2003). "Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling." Proc Natl Acad Sci U S A 100(10): 5944-5949.

Troyanskaya, O., M. Cantor, et al. (2001). "Missing value estimation methods for DNA microarrays." Bioinformatics 17(6): 520-525.

Tuncay, K., L. Ensman, et al. (2007). "Transcriptional regulatory networks via gene ontology and expression data." In Silico Biol 7(1): 21-34.

Wang, E., A. Lenferink, et al. (2007). "Cancer systems biology: exploring cancer-associated genes on cellular networks." Cell Mol Life Sci 64(14): 1752-1762.

Weigel, D. and H. Jackle (1990). "The fork head domain: a novel DNA binding motif of eukaryotic transcription factors?" Cell 63(3): 455-456.

Wernicke, S. and F. Rasche (2006). "FANMOD: a tool for fast network motif detection." Bioinformatics 22(9): 1152-1153.

Whitfield, M. L., G. Sherlock, et al. (2002). "Identification of genes periodically expressed in the human cell cycle and their expression in tumors." Mol Biol Cell 13(6): 1977-2000.

Wingender, E., X. Chen, et al. (2001). "The TRANSFAC system on gene expression regulation." Nucleic Acids Res 29(1): 281-283.

Wit, E. and J. McClure (2006). Statistics for Microarrays: Design, Analysis and Inference, John Wiley & Sons.

Woolf, P. J. and Y. Wang (2000). "A fuzzy logic approach to analyzing gene expression data." Physiol Genomics 3(1): 9-15.

Yeger-Lotem, E., S. Sattath, et al. (2004). "Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction." Proc Natl Acad Sci U S A 101(16): 5934-5939.

Yeung, K. Y., M. Medvedovic, et al. (2004). "From co-expression to co-regulation: how many microarray experiments do we need?" Genome Biol 5(7): R48.

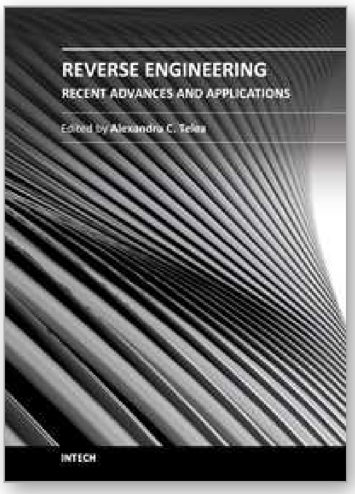
Zhang, Y., J. Xuan, et al. (2008). "Network motif-based identification of transcription factor-target gene relationships by integrating multi-source biological data." BMC Bioinformatics 9: 203.

Zhang, Y., J. Xuan, et al. (2009). "Reverse engineering module networks by PSO-RNN hybrid modeling." BMC Genomics: In Press.

Zhang, Y., J. Xuan, et al. (2010). "Reconstruction of gene regulatory modules in cancer cell cycle by multi-source data integration." PLoS One 5(4): e10268.

Zhu, J. and M. Q. Zhang (1999). "SCPD: a promoter database of the yeast Saccharomyces cerevisiae." Bioinformatics 15(7-8): 607-611.

Zhu, J. W., S. J. Field, et al. (2001). "E2F1 and E2F2 determine thresholds for antigen-induced T-cell proliferation and suppress tumorigenesis." Mol Cell Biol 21(24): 8547-8564.


REVERSE ENGINEERING
RECENT ADVANCES AND APPLICATIONS
Edited by Alexandru C. Telea
INTECH

Reverse Engineering - Recent Advances and Applications
Edited by Dr. A.C. Telea

ISBN 978-953-51-0158-1
Hard cover, 276 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

Reverse engineering encompasses a wide spectrum of activities aimed at extracting information on the function, structure, and behavior of man-made or natural artifacts. Increases in data sources, processing power, and improved data mining and processing algorithms have opened new fields of application for reverse engineering. In this book, we present twelve applications of reverse engineering in the software engineering, shape engineering, and medical and life sciences application domains. The book can serve as a guideline to practitioners in the above fields to the state-of-the-art in reverse engineering techniques, tools, and use-cases, as well as an overview of open challenges for reverse engineering researchers.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821