Analysis of Gene Expression Data **Spring Semester, 2005**

Lecture 11: June 02, 2005

Lecturer: Ron Shamir *Scribe: Uri Avni and Liza Potikha*[1]

11.1 Multi-level Modeling of Transcription Programs

11.1.1 Overview

A transcription program is the process of gene expression regulation by transcription factors. Our goal is to model a transcription program in details, in a way that corresponds to the biological reality. The transcription depends on two factors:

1. Concentration of TFs (dose)
2. DNA binding sites in the promoter (TF-gene affinity)

In this work the dependence on these factors is modeled by a two variables function called DAR function (Data-Affinity-Response) $\delta(d, \alpha)$ where d is the TF dose and α is its affinity. Previous works modeled the transcription using gene networks, where every gene is controlled by a set of other genes (for example the boolean model [1]). This model is depicted in Figure 11.1. The graph was evaluated using genes expression levels.

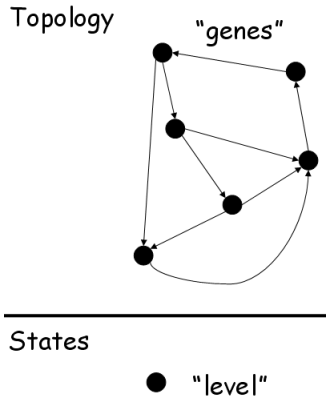

Figure 11.1:

[1]Based partially on scribes by Igor Bogudlov and Vladimir Koushnir, February 2000 , Amos Tanay and Eyal Zach, January 2002 and Tamir Tuller and Koby Lindzen, July 2002

In this work the interesting variable is the amount of active protein, instead of gene expression which measures mRNA level. The suggested model is depicted in Figure 11.2.

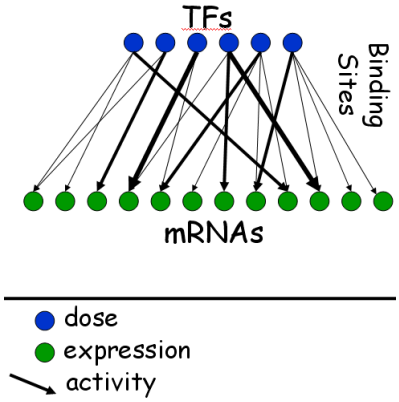

Figure 11.2:

In Figure 11.3 the expression level of gene g is controlled by three sites, s_1 s_2 s_3, via a regulation function $f_g(*,*,*)$. Each site activity is determined through a DAR function, according to TF doses and affinities.

Assumptions

We would like to learn the DAR functions of all the transcription factors, using mRNA expression levels and prior knowledge about TF-gene affinity. To reduce the problem dimensionality several assumptions about DAR functions are made:

1. Monotonicity - DAR functions monotonically increase with increasing dose and with increasing affinity.

2. Affinity and expression levels attain discrete values (they will be treated as sequential numbers, specifying their relative ranking).

3. The same DAR function for one TF applies to all of the genes.

The monotonicity assumption of DAR functions is backed up by experimental data, as seen in the monotonicity of TF-DNA interaction - GCN4 in Figure 11.4 and the monotonicity in TF-DNA interaction - MIG1 in Figure 11.5. Under these assumptions the problem of learning the DAR functions is solvable in polynomial time.

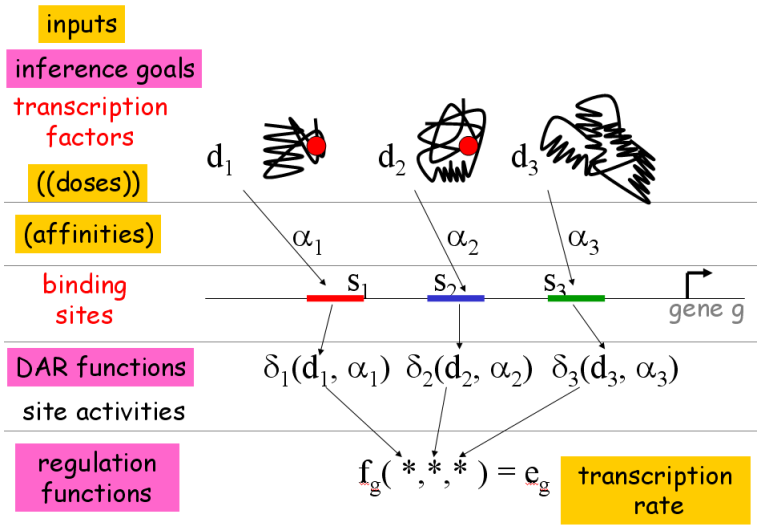

Figure 11.3:

11.1.2 The full picture

We'll examine a regulation network of genes, where expression levels get discrete values. e_g, the expression level of gene g, is usually modeled as a function of g's regulator genes expression levels, $e_g = f_g(e_{g_1}, .., e_{g_k})$, or as a discrete distribution $Pr(e_g \mid e_{g_1}, .., e_{g_k})$. Addition of dose and affinity to the model follows more accurately the biological process, as transcription factors are usually the regulator of the transcription, and not genes themselves. The introduction of these hidden variables adds complexity to the model: the dose and affinity values have to be learned. These values will be initially estimated, and the model will be optimized using this estimation. Then these parameters will be re-estimated using the optimized model, and so on alternately, as in the EM procedure.

Activity level of a TF is evaluated as the sum of expression levels of the genes which are targeted by this TF. A first estimation of TF-Gene affinity levels could be derived from chip-on-chip experiments. It could also be derived from a PSSM matrix of the relevant TF. Precise information about active protein doses is not yet available. The doses could be initially approximated by mRNA levels.

Mean Based Activity

A method of measuring activity level is based on [3]. First the expression matrix is normalized per condition. To estimate the activity of a TF we'll examine a subset G of the genes which are the regulatees of this TF. The mean expression of these genes is distributed normally.

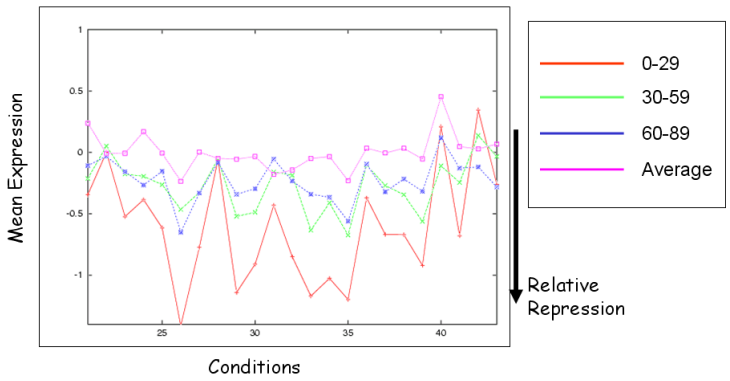

Figure 11.4: Ranked GCN4 targets from ChIP on chip data (Lee et al. 02) Expression mean from Ideker et al.

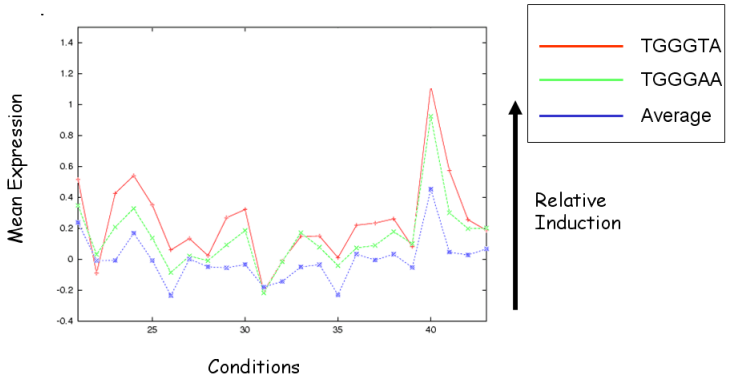

Figure 11.5: MIG1 consensus variants hits. Expression Mean from Ideker et al.

We'll select a subset S of the conditions, whose mean deviates significantly from the total mean. Finally use a z-score on S as the activity of G, and of the TF.

The Activity Score

We would like to give the estimated activity level a log likelihood score. Similarly to SAMBA [2] we'll examine the activity relatively to a random background model, and give the activity a p-value based on a statistical test.

The ASAP Score

The ASAP (Activity Score Approximated P-val) score calculates a p-value on the distribution of weights in a gene set. The weights don't distribute normally, so a p-value will be calculated

heuristically. The weights distribution function is learned by sampling gene sets of the same size. Every condition is examined separately, and the scores are summed over all conditions. This process is depicted in Figure 11.6

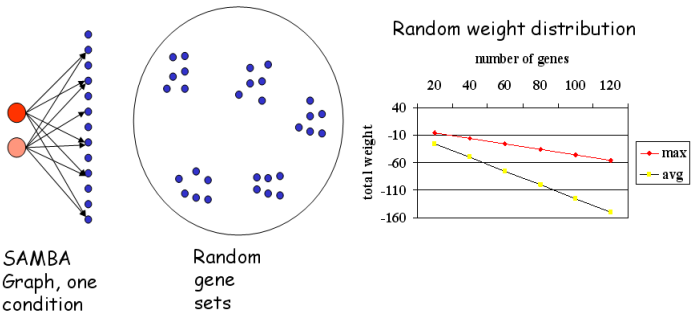

Figure 11.6:

Active TF discovery framework

In order to asses TF initial doses and/or TF-gene affinities, we assume that a functionally important TF is characterized by a typical promoter motif. The following framework discovers active DNA motifs, and assesses their relative activity across experimental conditions.

To measure the activity of a binding site motif, we first identify the set of genes that contain it in their promoters. We then evaluate the level of co-expression of that set in the data using a novel scoring method, which takes into account the individual expression distribution of each gene and condition. Motifs are defined in a more descriptive way than the commonly used PSSMs by taking into consideration their location distribution along the promoter. This method is used in a screening procedure that combines exhaustive search for k-mer seeds and their refinement to high-activity motifs. This procedure may be useful independently of the complete TP model inference algorithm. A position specific substitution matrix (PSSM) is a standard way of representing DNA motifs. A PSSM P is a vector of distributions over ACGT denoted $P[0...l] : ACGT \rightarrow [0,1]$. In practice, many binding sites motifs tend to concentrate in particular regions within the promoter, as in Figure 11.7. To model this phenomenon, we extend the standard PSSM definition by adding to it a distribution of its location: A Localized PSSM (LPSSM) is a PSSM with an additional location distribution P_l. The likelihood of an LPSSM match with a sequence s in location j is simply the product of profile probability and location probability: $Pr(P,s,j) = P_l(j)\prod_{0\leq i\leq l} P(i,s[i+j])$. The matching likelihood of a string s and an LPSSM P is $ML(P,s) = max_j Pr(P,s,j)$.

The LPSSM model is found in an EM-like motif optimization algorithm, which is detailed in [13].

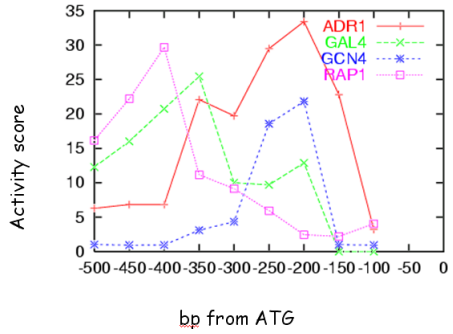

Figure 11.7: Activity score of known yeast PSSMs localized to different offset windows (100 bp)

11.1.3 Learning Dose-Affinity-Response Function

We would like to evaluate a model score using the gene expression levels, to determine how well the model predicts experimental data. For example, the prediction for expression level of gene g which depends on 3 TFs is $e_g = F_g(DAR_{tf1}(D_{tf1}, A_{g:tf1}), DAR_{tf2}(D_{tf2}, A_{g:tf2}), DAR_{tf3}(D_{tf3}, A_{g:tf3}))$, where D is dose and A is affinity (See Figure 11.8).

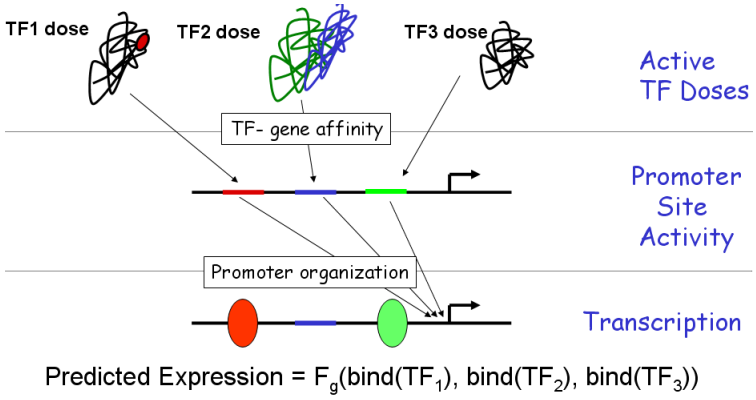

Figure 11.8:

Scoring a topology

We first focus on the dependency between sites and genes. We assume that both site activities and transcription rates attain values in a discrete alphabet $C = \{1..C\}$ and that their regulatory logic is a collection of unconstrained combinatorial functions. Specifically, the

regulation of g is assumed to be via a regulation function $f_g : C^{|N(g)|} \rightarrow C$ which determines the transcription rate of g as a function of the activity levels of its regulator sites, $N(g)$.

A **transcription program topology** is a bipartite graph $M = (T, V, S)$ consisting of a set T of active TFs, a set V of genes and a set S of binding sites (or edges) connecting TFs and genes. Assume we are given a topology M and a set E of experimental conditions, where each condition $u \in U$ has a vector of gene expression levels e^u. $e^u_g \in C$ is the expression level of gene g in condition u and $E = \{e^u | u \in U\}$. Suppose we also have the set A of (measured or predicted) activity levels r^u_s of each site $s \in S$ under each condition $u \in U$. A topology score will be a real valued function in the form of $\phi(M, E, A)$, that assesses the dependencies among site levels and gene expression levels given the topology M. The score is decomposable - it can be expressed as a sum of separate contributions from individual genes, i.e. $\phi(M, E, A) = \sum_{v \in V} \phi(r_{N(v)}, e_v)$.

Denote by n^v_r the number of conditions $u \in U$ in which v's regulator sites $N(v)$ attain the specific combination of activity values r (r is a vector of size $N(g)$). Denote by $n^{v,j}_r$ the number of conditions meeting the previous criterion which also have $e^u_v = j$. Also denote by n the number of conditions $|U|$ and by $n^{v,j}$ the number of times e^u_v equals j. For a given gene g, we'll examine how many conditions in U achieve transcription rate $c \in C$. This is done once using the experimental data, and once under the topology model. The results of this counting are two vectors of length $|C|$, which express the distribution of expression levels under the model and in the data. These vectors will be compared using their mutual information. Mutual information under a combination of regulator sites activity values r is defined as

$$I(r_{N(v)}, e_v) = -\sum_j \frac{n^{v,j}}{n} log(\frac{n^{v,j}}{n}) + \sum_r (\frac{n^v_r}{n} log(\frac{n^v_r}{n}) - \sum_j \frac{n^{v,j}_r}{n} log(\frac{n^{v,j}_r}{n}))$$

Expression fit score of gene g equals χ^2 value of the mutual information between the predictions and the observations. Model Score is then the sum of fit scores for all the genes.

Optimizing DAR functions

We are given a transcription program M, site affinities α_s, and a set of experiments $U = (e^u_v, d^u_t)$. The goal is to find a set of DAR functions, giving rise to site activities $A = r^u_s \mid u \in U, s \in S$, such that the topology score $\phi(M, E, A)$ is optimized.

Suppose we have determined the DAR functions for all transcription factors except for one. We will optimize the DAR of the remaining TF, while fixing all the others. We represent the dose-affinity plain as a matrix with a column for each affinity value and a row for each

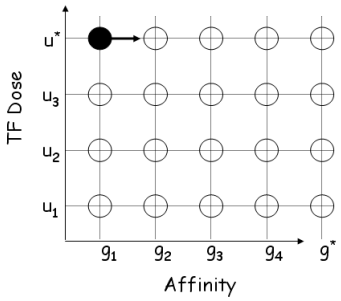

Figure 11.9:

dose value (see Figure 11.9). The following algorithm represent a single DAR optimization as a longest path problem in an appropriate grid graph built over that matrix.

For simplicity we'll assume that the DAR function has two activity levels. Suppose that we have a partial DAR model except for one DAR function δ_t. Let $u_1 ... u_{|U|}$ be the set of experiments sorted by increasing doses, and let $g_1, ...g_n$ be the genes regulated by TF t via sites $s_1, ..., s_n$ respectively, where genes and sites indices are sorted by increasing site affinities. We build a grid graph with horizontal edges between adjacent genes,$((u_j, g_i), (u_j, g_{i+1}))$, and vertical edges between adjacent conditions, $((u_j, g_i), (u_{j-1}, g_i))$. We'll add a new artificial condition $u_{|U|+1}$ and a new gene g_{n+1} to be consistent on the borders of the graph. A horizontal arc $((u_j, g_i), (u_j, g_{i+1}))$ represents a dose threshold between the low and the high values of the function δ_t in the i'th affinity level. Given this dose threshold we can determine the contribution of gene g_i to the total score, $\phi(r_{N(g_i)}, e_{g_i})$. This score will be used as the weight of the horizontal arc $((u_j, g_i), (u_j, g_{i+1}))$. The weight of vertical arcs is set to zero.

A path in the grid from $(u_{|U|+1}, g_1)$ to (u_1, g_{n+1}) defines the monotone DAR function δ_t fully, by determining a dose thresholds for every affinity level (See Figure 11.10).

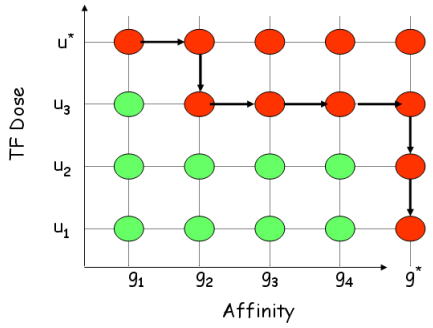

Figure 11.10:

We find the optimal DAR function by solving the longest path problem in the directed acyclic graph. This could be done in $O(E + VlogV)$ [4]. When the DAR function has more

than two activity levels we'll build a similar grid graph in a higher dimension. There is an exponential dependency on the number of activity levels.

To optimize the set of DAR functions we'll start with some arbitrary set of DAR functions and repeatedly select one TF, re-optimize its DAR function, and add it to the current set of functions. The new obtained set must have a score that is equal or higher than that of the previous one. Hence, the whole process is monotonically improving, and convergence to a local optimum is guaranteed. Optimizing simultaneously more than one DAR function is NP-Hard [13].

Dose optimization

We would like to tune the initial evaluation of TF doses. This is done heuristically by repeatedly optimizing the dosage of one TF in one condition at each step. In a given experimental condition the dose of a TF is the same across all genes. Our goal is to find an ordered list of the conditions according to the TF doses. To improve a given conditions list we'll select a pair of conditions u_1, u_2 such that swapping their locations in the list gives the maximal raise to the topology score ϕ (See Figure 11.11). Applying this step repeatedly converges to a local optimum.

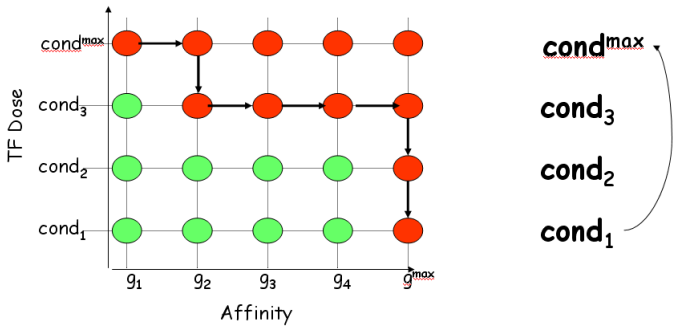

Figure 11.11:

By alternating between model optimization and dose optimization, we finally converge to a locally optimal solution.

11.1.4 Results

The data used in the results section are 61 gene expression profiles on yeast carbohydrate metabolism from [9], [12], [5]. TF binding chip[2] data was taken from [11]. Length of promoters: 600bp upstream. Known PSSMs were taken from TRANSFAC.

Table 11.1: Transcriptional program for the Galactose system

Motif	Activity	Remarks
CGG(11N)CCG	30.8	GAL4
TGACTCAWT	22.57	GCN4
TGGGGTA	22.13	ADR1/MIG1
CANCCCC	26.92	U1
CCG(5N)CCG	26.6	U2
TT(9N)CCCC	22.03	U3

The whole process of finding k-mers, motif creation, and optimizing DARs and doses was applied to the data, using only 23 experiment profiles from Ideker et al. Thereafter we applied active TF screening framework, by checking the contribution of each TF to the final score. If the contribution is neglegable, this factor can be removed. The algorithm detected 6 sites above noise level. 3 of the detected sites are known, and 3 are putative sites. Results are described in Table 11.1

The most significant discovered factor is GAL4. Figure 11.12 describes the genes controlled by this factor. The isolated genes in the figure are controlled solely by GAL4, and the rest of the genes are controlled by GAL4 and by additional factors, as depicted.

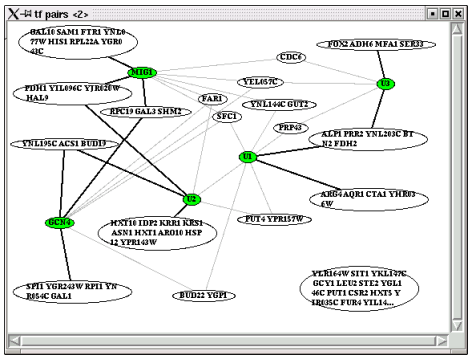

Figure 11.12: The galactose system: GAL4 regulated genes

The DAR functions discovered for these factors as well as the functional role of the factors are shown in Figure 11.13. Gene sets that are activated by the same factor are related to a certain functionality, as follows from a GO score. The contribution of the factors to the total score is shown in Figure 11.14.

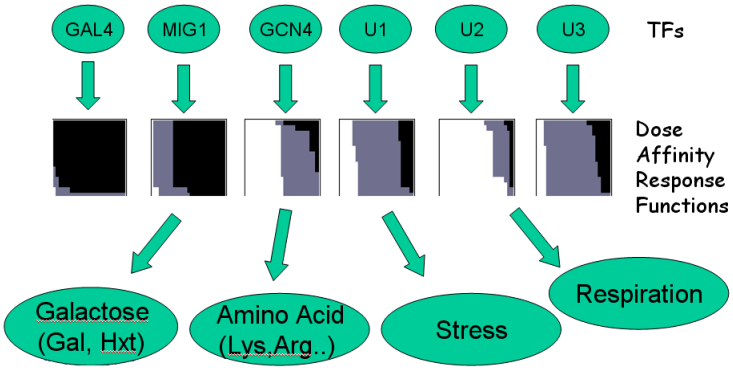

Figure 11.13: The Transcriptional Program Model

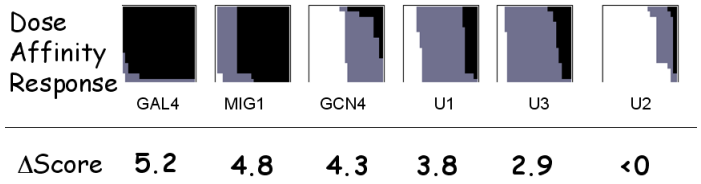

Figure 11.14: Testing TFs contribution

A different method of motif detection was applied in [10] on yeast data independently. This method uses cross-specie DNA comparison to find significant motifs. The results validated the discovery of one of the discovered sequences (U2), as seen in Figure 11.15.

Wrap-Up and future work

Basic network models that we have already mentioned in the overview section make two critical assumptions on the regulatory system: a) the regulators states can be represented using the expression rate of the genes encoding them, and b) the relation between a potential regulator and a regulatee is a binary attribute (either there is an arc or there is none). The described model relaxes both assumptions in an attempt to build a mechanistic model which can follow more faithfully our knowledge on biological regulatory systems.

One of the key points of the model is exposing the hidden variables - activity levels of regulators sites - that are predicted by the model from TF doses.

Future work may include using more biological data sources and applying more biological constraints (for example, constraining the combinatorial functions to a class of biologically reasonable logics (Cf. [6])).

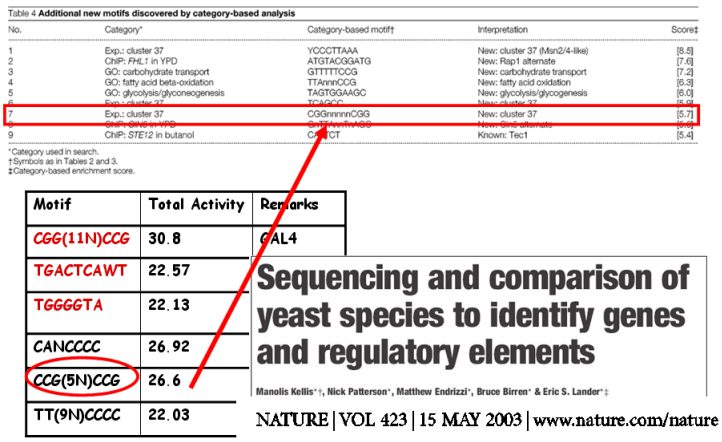

Table 4 **Additional new motifs discovered by category-based analysis**

No.	Category*	Category-based motif†	Interpretation	Score‡
1	Exp.: cluster 37	YCCCTTAAA	New: cluster 37 (Msn2/4-like)	[8.5]
2	ChIP: *FHL1* in YPD	ATGTACGGATG	New: Rap1 alternate	[7.6]
3	GO: carbohydrate transport	GTTTTTCCG	New: carbohydrate transport	[7.2]
4	GO: fatty acid beta-oxidation	TTAnnnCCG	New: fatty acid oxidation	[6.3]
5	GO: glycolysis/glyconeogenesis	TAGTGGAAGC	New: glycolysis/glycogenesis	[6.0]
6	Exp.: cluster 37	TCAGCC	New: cluster 37	[5.8]
7	Exp.: cluster 37	CGGnnnnnCGG	New: cluster 37	[5.7]
8	ChIP: *SWI5* in YPD	G[illegible]	New: Swi5 alternate	[5.6]
9	ChIP: *STE12* in butanol	G[illegible]CT	Known: Tec1	[5.4]

*Category used in search.
†Symbols as in Tables 2 and 3.
‡Category-based enrichment score.

Motif	Total Activity	Remarks
CGG(11N)CCG	30.8	GAL4
TGACTCAWT	22.57	
TGGGGTA	22.13	
CANCCCC	26.92	
CCG(5N)CCG	26.6	
TT(9N)CCCC	22.03	

Sequencing and comparison of yeast species to identify genes and regulatory elements

Manolis Kellis*†, Nick Patterson*, Matthew Endrizzi*, Bruce Birren* & Eric S. Lander*‡

NATURE | VOL 423 | 15 MAY 2003 | www.nature.com/nature

Figure 11.15: Validation of CCG(5N)CCG

11.2 Interactive Inference and Experimental Design

This section is based on a paper by Iddeker, Thorsson and Karp [8]. Our goal is to infer the underlying genetic network from a series of steady-state gene expression profiles for a set of perturbations. We assume the Boolean genetic network model for the gene network. We restrict ourselves only to acyclic networks. This has a technical advantage: in the case of acyclic networks there is no need for assumptions about the time delays of the components. From the other hand acyclic networks description will mostly stay biologically informative: even if most networks do have feedback loops, there are generally few of them, and the main pathway is often acyclic. (The analysis of cyclic networks is complicated by the possibility of oscillatory behavior. For cyclic networks, one may adopt either a *synchronous* model in which each component has a fixed, known delay, or an *asynchronous* model in which the delays are unknown and even nondeterministic.)

The proposed strategy is based on repeated and interactive application of two analytical methods: the *predictor* and the *chooser*. According to this strategy, the underlying network of interest is exposed to an initial series of genetic and/or biological perturbations and a steady-state gene expression profile is generated for each of them.

Next, a method called the *predictor* is used to infer one or more hypothetical Boolean networks consistent with these profiles. When several networks are inferred, the predictor returns only the most parsimonious, as measured by those networks having the fewest number of interactions.

Depending on the complexity of the genetic network and the number of initial perturbations, numerous hypothetical networks may exist. And this is why, a second method called the *chooser* is used to propose an additional perturbation experiment to discriminate among the set of hypothetical networks determined by the predictor.

The two methods may be used iteratively and interactively to refine the genetic network: at each iteration, the perturbation selected by the chooser is experimentally performed to generate a new gene expression profile, and the predictor is used to derive a refined set of hypothetical gene networks using the cumulative expression data.

11.2.1 The Predictor

The predictor is a method for inferring Boolean networks using the expression data given by the matrix E. We seek for a Boolean function f_n independently for each node a_n. To this end, we first pick the input variables to f_n: we determine a minimum set s_n of nodes, whose levels must be input to f_n, in order for s_n to explain the observed data E. Then, we construct a truth table using these nodes as inputs.

Specifically, the function for node a_n is determined according to the following procedure:

1. **Build sets S_{ij} of nodes with different values in rows i and j**
   Consider all pairs of rows (i, j) in E in which the expression level of a_n differs, excluding

rows in which a_n was itself forced to a high or low value. For each such pair, find the set S_{ij} of all other nodes whose expression levels also differ between the two rows (i, j). Because the network is self-contained, a change in at least one of these genes or stimuli must have caused the corresponding difference in a_n. Therefore, at least one node in this set must be included as a variable in f_n.

2. **Find a minimum cover set S_{min} of $\{S_{ij}\}$**
   Identify the smallest set of nodes S_{min} required to explain the observed differences over all pairs of rows (i, j), i.e., S_{min} is such that at least one of its nodes is present in each set S_{ij}. This task is a classic combinatorial problem called *minimum set cover*, which can be solved by a branch and bound technique. More than one smallest set S_{min} may be found, in which case a distinct function f_n is inferred and reported for each such set.

3. **Determine truth table of a_n from S_{min} and E**
   Once S_{min} has been determined for the node a_n, a truth table is determined for f_n in terms of the levels of genes and/or stimuli in S_{min} by taking relevant levels directly from E. If all combinations of input levels are not present in E, the corresponding output level for gene a_n cannot be determined and is represented by the symbol "*" in the truth table.

If a node has more than one minimum cover set, several networks are inferred, each with a distinct function corresponding to each set. If several such nodes exist, a separate network hypothesis is returned for each combination of functions at each node. The minimum set cover ensures that only the most parsimonious networks will be returned.

11.2.2 The Chooser

The chooser procedure takes as its input the L hypothetical equiprobable networks generated by the predictor. Its goal is to choose a new perturbation p, from a set of allowed perturbations P, which best discriminates between the L hypothetical networks. Allowed perturbations are the practical once: only non-lethal perturbations, with just several genes to be forced may be performed.

The following entropy-based algorithm is used for the chooser:

1. For each perturbation $p \in P$ compute the network state resulting from p for each of the L networks. A given perturbation would result in a total of S distinct states over the L networks $(1 \leq S \leq L)$. Evaluate the following entropy score H_p, where l_s is the number of networks giving the state s $(1 \leq s \leq S)$, as follows:

$$H_p = -\sum_{s=1}^{S} \frac{l_s}{L} log_2(\frac{l_s}{L}) \tag{11.1}$$

2. Choose the perturbation p with the maximum score H_p as the next experiment.

The entropy measure H_p describes expected gain in information when performing the perturbation p. The more distinct states the networks produce, the more information is obtained.

According to the predictor procedure, a network may have the "*" symbol in its truth table, meaning that any function value is equally probably for a given node and input. In this case the chooser randomly assigns either 0 or 1 to to replace the "*". In addition, when L is large, it may be infeasible to calculate the entropy for all the hypothetical networks. In this case the entropy is calculate by Monte-Carlo procedure, over a random sample.

The best perturbation returned by the chooser is then performed on the network, and the new measured gene expression values are added to E. A new, narrower set of parsimonious networks is then inferred by the predictor, and so on. This design process proceeds iteratively, choosing a new perturbation experiment in each iteration, until either a single parsimonious network remains ($L = 1$), or no perturbation in P can discriminate between any of the L networks ($H_p = 0$).

11.2.3 Evaluation of the Technique

A series of experiments have been performed by the authors of [8] to evaluate the applicability of the method. The evaluation criteria and results are presented below.

Predictor Evaluation

The predictor procedure was evaluated using both random and non-random simulated networks. In random simulations acyclic genetic networks of size N and maximum in-degree k were randomly generated. The expression matrix E consisted of the wild-type (without any nodes forced to high or low) and all single perturbations. In addition, a number of non-random networks, modelled after known biological networks were simulated. For each such network, the most parsimonious models were created by the predictor.

The similarity between each inferred network and its target was evaluated with regard to *sensitivity*, defined as the percentage of edges in the target network that were also present in the inferred one, and *specificity*, defined as the percentage of edges in the inferred network that were also present in the target network. Figures 11.16 and 11.17 show the results.

Each measurement is an average over 200 simulated target networks. As one can see, the specificity was always significantly higher than sensitivity, and both steadily decreased as N and k were increased.

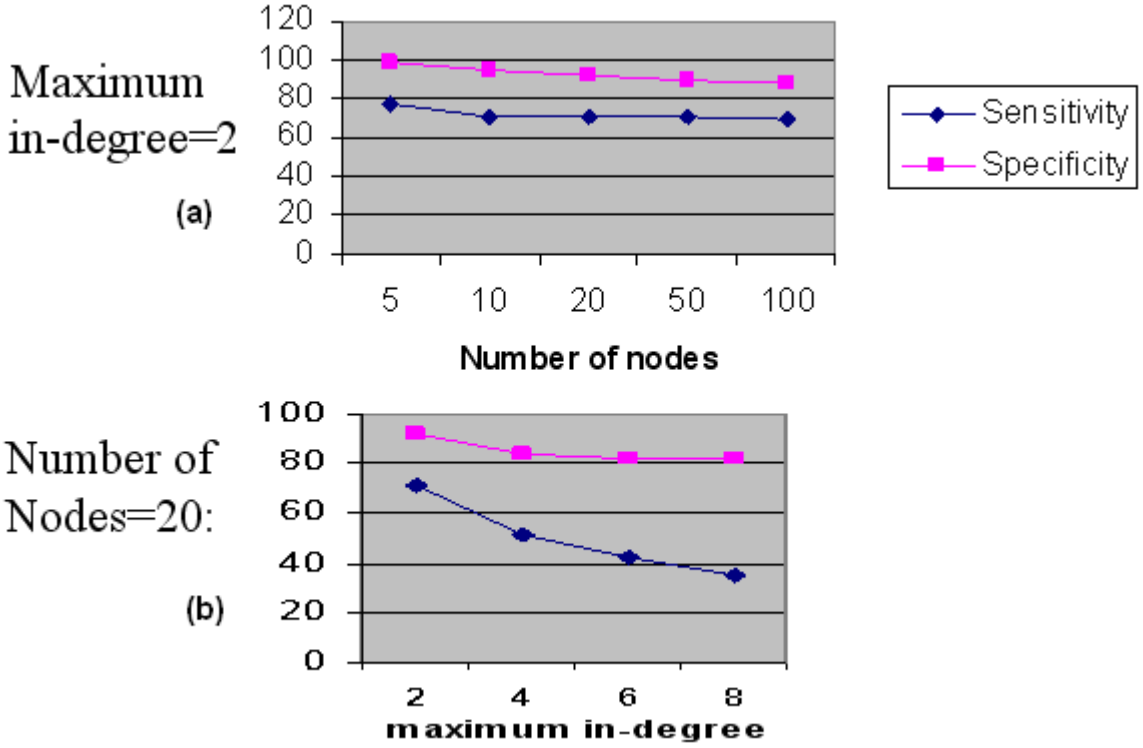

Figure 11.16: (a) Sensitivity and specificity in percents vs. number of nodes (b) Sensitivity and specificity vs. maximum in-degree.

The number of nodes whose functions had only a single minimal solution was approximately 90% for $k = 2$, independent of N. Thus, although the number of inferred networks grew exponentially with N, this number was subjected to ambiguities at just 10% of the nodes.

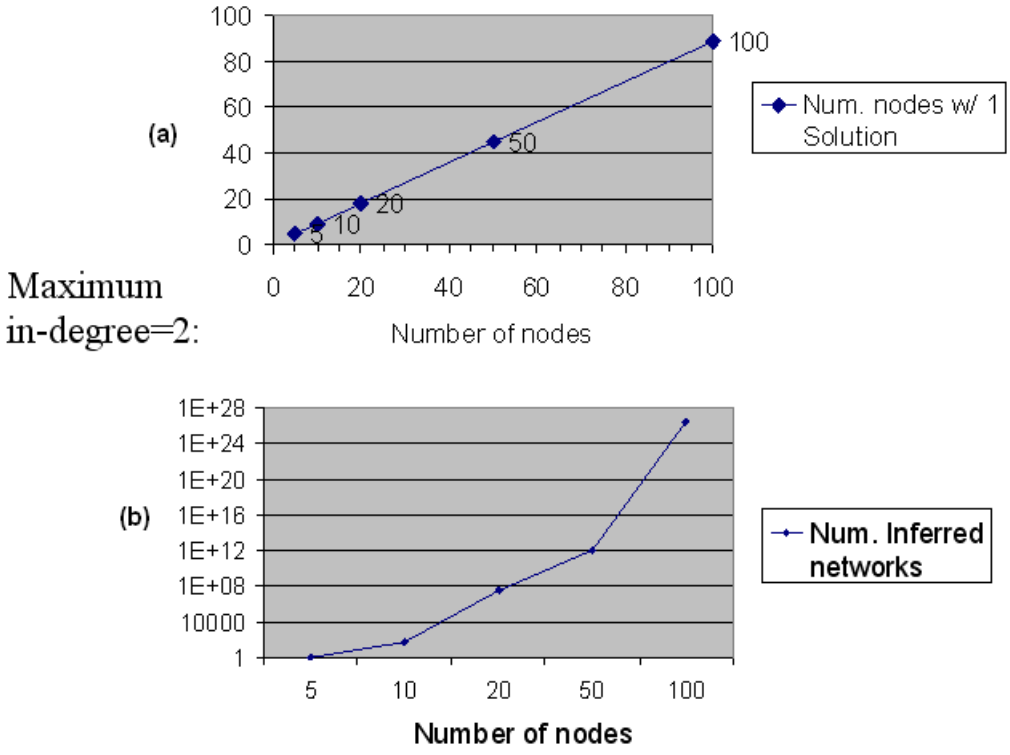

Figure 11.17: (a) Percentage of networks with one solution (b) Number of inferred networks vs. number of nodes.

Chooser Evaluation

In order to evaluate the performance of the chooser the following simulation was performed: A network with 20 nodes, 24 edges and maximum in-degree 4 was generated. The expression matrix E consisted of the wild-type and all single perturbations. Next, 8 parsimonious networks were inferred, all with 21 edges, which were consistent with E. The chooser was used to select a double perturbation which had maximal entropy score over the 8 networks, and the process was repeated iteratively until only a single network was inferred. The results are summarized in the figure 11.20. They show a pattern of jumps and decays in the number

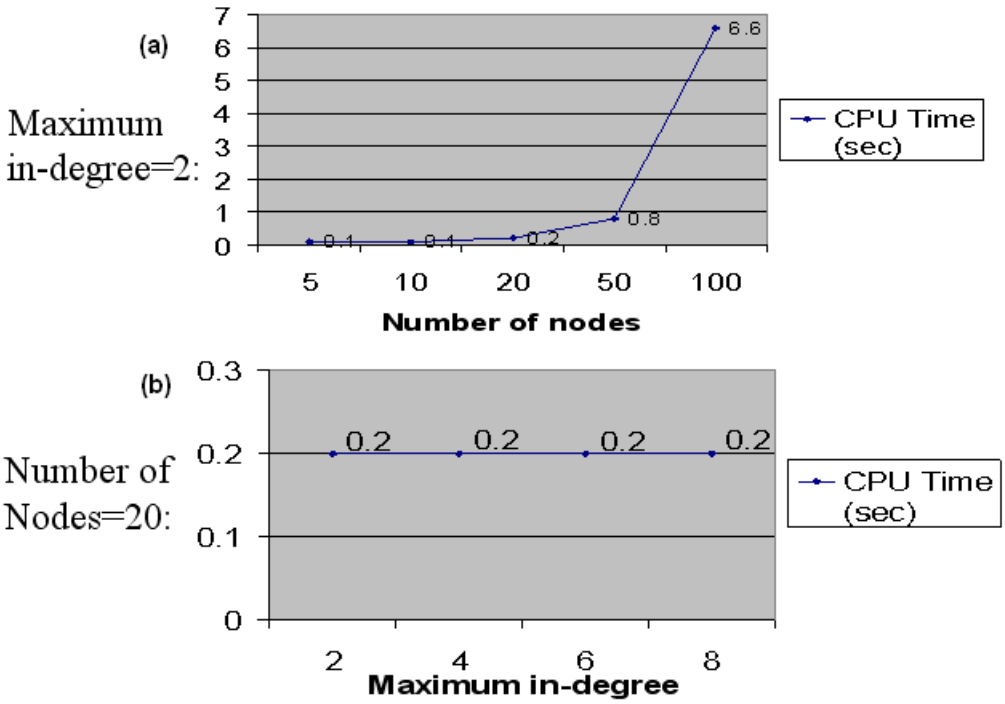

Figure 11.18: (a) CPU time vs. number of nodes. (b)CPU time vs. maximum in-degree. (CPU time was computed using a 500 MHz Pentium III processor)

Number of nodes	Maximum in-degree	Total simulated Edges	Num. Inferred Networks	Total Inferred Edges	Num. Shared Edges	Sensitivity	Specificity	Num. Nodes / 1 Solution	CPU Time (sec)
5	2	4 (0.1)	1 (0.2)	3 (0.1)	3 (0.1)	77%	99%	5 (0.0)	0.1 (0.0)
10	2	12 (0.1)	60 (50)	9 (0.1)	9 (0.1)	71%	95%	9 (0.1)	0.1 (0.0)
20	2	27 (0.2)	3*10^7	21 (0.2)	19 (0.1)	71%	92%	18 (0.1)	0.2 (0.0)
50	2	72 (0.2)	1*10^12	57 (0.3)	51 (0.3)	71%	90%	45 (0.2)	0.8 (0.0)
100	2	146 (0.7)	3*10^26	119 (0.9)	104 (0.7)	70%	88%	89 (0.5)	6.6 (0.3)
20	4	44 (0.3)	2*10^6	28 (0.3)	23 (0.2)	51%	84%	16 (0.1)	0.2 (0.0)
20	6	57 (0.5)	2*10^7	33 (0.3)	27 (0.2)	42%	82%	14 (0.2)	0.2 (0.0)
20	8	69 (0.7)	9*10^7	38 (0.4)	31 (0.3)	35%	82%	13 (0.2)	0.2 (0.0)

Figure 11.19: Summary of predictor evaluations.

of network solutions, correlated with an increase in the number of inferred edges.

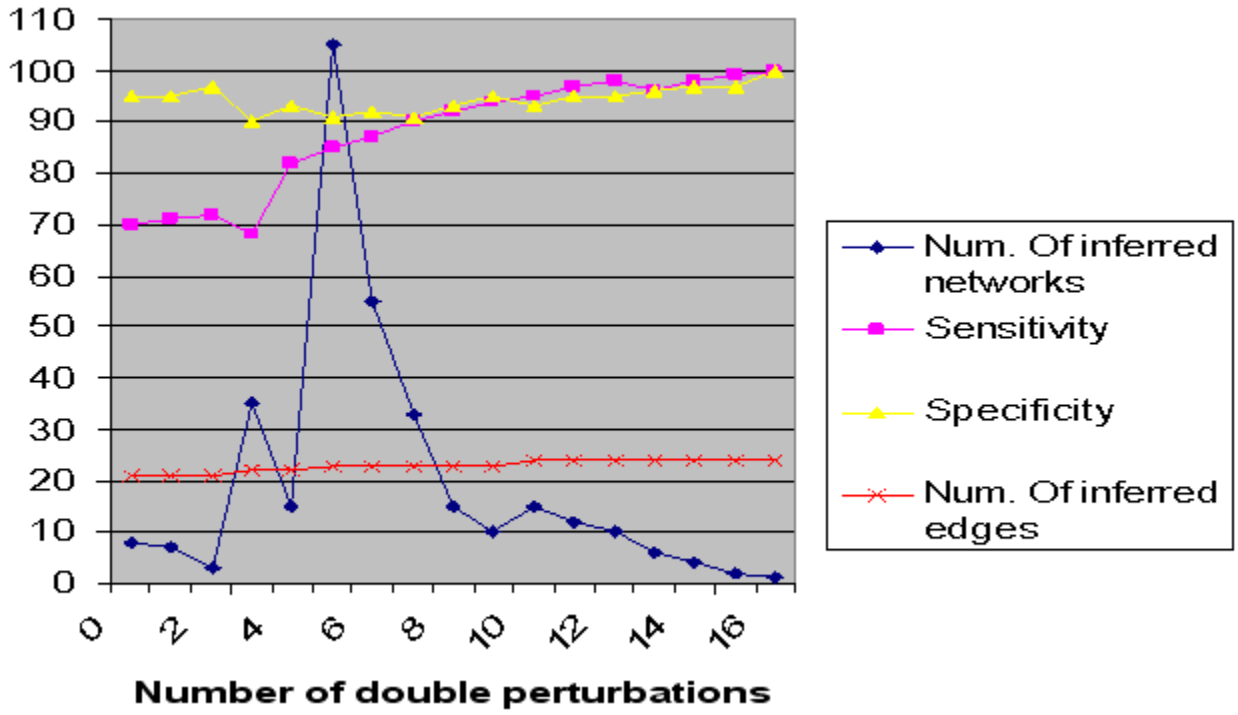

Figure 11.20: Source: [8]. Summary of chooser evaluation: progress through experimental design.

Lower Bound Comparison

The following theorem, due to Hertz, specifies a lower bound on the amount of data needed to specify a network:

Theorem 11.1 ([7]) *A lower bound on the number of gene expression profiles which must be observed in order to uniquely specify a genetic network with N nodes and maximum in-degree k where $N \gg k$ is $klog_2(\frac{N}{k})$.*

It is therefore interesting to characterize the behavior of the predictor-chooser strategy with respect to the lower bound. For this purpose 50 networks for each of several values of N with $k = 2$ were generated. The wild-type perturbation and all single ones were simulated on each network. The chooser was used iteratively in conjunction with the predictor to refine the network hypotheses. The results, shown in Figure 11.21, indicate a logarithmic behavior.

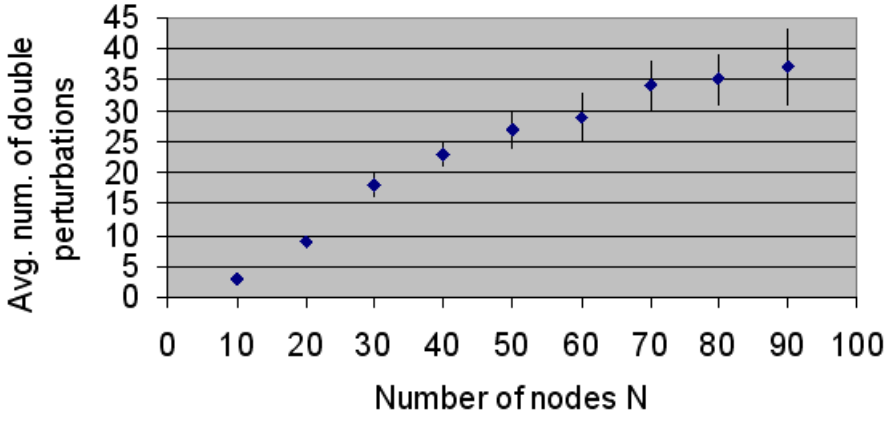

Figure 11.21: Source: [8]. Average number of perturbations vs. network size N for k=2 with bars indicating standard error.

Future work

There are number of extensions that may be done for the described Chooser/Predictor method:
First, in nearly all cases of practical interest some knowledge of network genes and interactions is available. In this regard, pre-existing information about the network may be incorporated during the inference process.
Second, the observed levels of gene products and other macromolecules may be such that a two-level description misses important features of the network. In these cases, a multi-level description (greater than two) may be adequate to describe the data. It may also be possible to extend the method for use with continuous (rather than discrete Boolean or multi-level) gene expression data.
Third, as was already mentioned, only genetic networks which do not contain cycles were considered. This restriction may be sufficient to describe certain biochemical networks, but biological examples of cyclic gene networks are also known. Therefore, another future direction is to allow cyclic solutions in the inference procedure.
Fourth, we currently do not allow for noise or other imperfections in the gene expression data sets used for network inference. Gene expression levels measured with DNA microarrays or other technologies are subject to an appreciable amount of experimental variability, and the impact of this variability on our method should be evaluated. May be inference method could be modified to account for noisy data.
Finally, proposed methods may be used in conjunction with existing software for grouping genes. For instance, a clustering algorithm might be used to reduce the apparent size of the network by grouping genes according to similar expression level over the series of perturbations performed, then one representative from each cluster could be supplied to the network

inference method.

Bibliography

[1] Akutsu, Kuhara, Maruyama, and Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[2] Roded Sharan Amos Tanay and Ron Shamir. Discovering statistically significant biclusters in gene expression data. *In Bioinformatics*, 18(Suppl. 1):S136–S144, 2002.

[3] Harmen J. Bussemaker, Hao Li, and Eric D. Siggia. Regulatory element detection using correlation with expression (abstract only). In *RECOMB '01: Proceedings of the fifth annual international conference on Computational biology*, page 86, New York, NY, USA, 2001. ACM Press.

[4] Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987.

[5] A. Gasch, P. Spellman, C. Kao, O. Carmel-Harel, M. Eisen, G. Storz, D. Botstein, and P. Brown. Genomic expression program in the response of yeast cells to environmental changes, 2000.

[6] I. Gat-Viks and R. Shamir. Canalyzing functions and scoring functions in genetic networks. 2002.

[7] J. Hertz. http://www.nordita.dk/~hertz/projects.html. In *Pacific Symposium on Biocomputing*, Maui, Hawaii, 1998.

[8] T.E. Iddeker, V. Thorsson, and R. M. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. In *Pacific Symposium on Biocomputing 5*, pages 302–313, 2000.

[9] T. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292(5518):929–934, May 2001.

[10] M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E. S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–254, May 2003.

[11] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804, October 2002.

[12] M.C. Lopez and H.V. Baker. Understanding the growth phenotype of the yeast gcr1 mutant in terms of global genomic expression patterns. *J Bacteriol*, 182(17):4970–8, 2002.

[13] Amos Tanay and Ron Shamir. Modeling transcription programs: inferring binding site activity and dose-response model optimization. In *RECOMB '03: Proceedings of the seventh annual international conference on Computational molecular biology*, pages 301–310, New York, NY, USA, 2003. ACM Press.