Analysis of Gene Expression Data **Spring Semester, 2002**

Lecture 11: June 21, 2002

Lecturer: Ron Shamir *Scribe: Tamir Tuller and Koby Lindzen*[1]

11.1 Identification of Gene Regulatory Networks by Gene Disruptions and Overexpressions

11.1.1 Preface

This section is based on the article of Akutsu et al. [2]. Almost all proofs and all figures were take from this paper. In this section we show how to identify a gene regulatory network from data obtained by multiple gene perturbations (disruptions and overexpressions) taking into account the number of experiments and the complexity of experiments. An experiment consists of parallel gene perturbations and their total number is the complexity of an experiment.

11.1.2 Model Description and Definitions

We define the gene regulatory network as in Lecture 9. We further assume that it satisfies the following conditions:

1. When the boolean function f_v assigned to v has k inputs, k input lines (directed edges) come from k distinct nodes $u_1, ..., u_k$ other then v.
2. For each $i = 1, ..., k$ there exists an input $(a_1, .., a_k) \in \{0,1\}^k$ with $f_v(a_1, ..., a_k) \neq f_v(a_1, .., \bar{a}_i, ..., a_k)$ where $\bar{a}_i$ is a complement bit of a_i.
3. A node v with no inputs has a constant value (0 or 1).

Definition The *state* of a gene v is active (inactive) if the value of v is 1 (0).

Definition The node v is called $AND(OR)$ node if the value of $f_v(a_1, ..., a_k)$ is determined by the formula $\ell(u_1) \wedge \ell(u_2) \wedge ... \wedge \ell(u_k)$ $(\ell(u_1) \vee \ell(u_2) \vee ... \vee \ell(u_k))$, where $\ell(u_i)$ is either u_i or $\neg u_i$.

[1]Based on scribes by Igor Bogudlov and Vladimir Koushnir, February 11, 2000 and Amos Tanay and Eyal Zach, January 17, 2002.

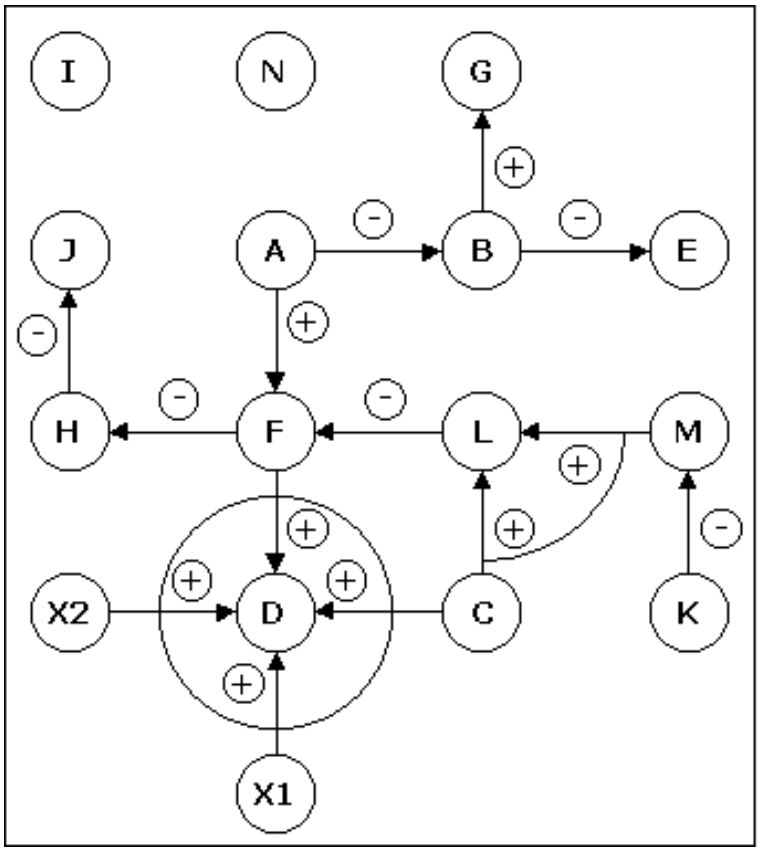

Figure 11.1: Source: [2]. Example of a gene regulatory network with 16 genes ($\oplus$ means "activation" and $\ominus$ means "deactivation" of the gene). Gene F is *activated* by gene A and is also *inactivated* by gene L ($f_F(A, L) = l(A) \wedge \neg l(L)$). Gene D is expressed if all its predecessors $C, F, X1, X2$ are expressed (AND - node).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	X1	X2
Normal Condition	1	0	1	1	1	1	0	0	1	1	1	0	0	1	1	1
Disruption of A	0	1	1	0	0	0	1	1	1	0	1	0	0	1	1	1
Overexpression of B	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1

Figure 11.2: Source: [2]. Gene expressions by disruption and overexpression from the gene regulatory network of Figure 11.1 (0 - the gene is not expressed , 1 - the gene is expressed).

Definition An edge (u, v_i) is called an *activation edge (inactivation edge)* if $\ell(u_i)$ is a positive literal (negative literal).

For a gene v, *a disruption* of v forces v to be inactive and *overexpression* of v forces v to be active. Let $x_1, ..., x_p, y_1, ..., y_q$ be mutually distinct genes of G. An *experiment* with gene overexpressions $x_1, ..., x_p$ and gene disruptions $y_1, ..., y_q$ is denoted by $e = \langle x_1, ..., x_p\ , \neg y_1, ..., \neg y_q \rangle$. The *cost* of e is defined as $p + q$. Three cases of gene expression conditions (normal, disruption of gene A, overexpression of gene B) are presented in figure 11.2.

Let us define the nodes with fixed values given *experiment* e:

Definition The node v is said to be *invariant* if it satisfies one of the following conditions:

- v belongs to e, i.e., v is disrupted or overexpressed in e.
- v has in-degree 0.
- v depends only on invariant nodes.

We now define different types of states of gene regulatory network G:

1. A *global state* of G is a mapping $\psi : V \to \{0, 1\}$. The global states of the genes need not be consistent with the gene regulation rules.
2. The global state ψ of G is *stable* under experiment $e = \langle x_1, ..., x_p\ , \neg y_1, ..., \neg y_q \rangle$ if $\psi(x_i) = 1\ (i = 1, ..., p)\ ,\ \psi(y_j) = 0\ (j = 1, ..., q)$ and it is consistent with all gene regulation rules, i.e., for each node v with inputs $u_1, ..., u_k\ , \psi(v) = f_v(\psi(u_1), ..., \psi(u_k))$. Otherwise, it is called *unstable*.
3. The global state ψ of G is an *observed global state* under experiment $e = \langle x_1, ..., x_p\ , \neg y_1, ..., \neg y_q \rangle$ if it satisfies all gene regulation rules for invariant nodes.
4. The observed global state ψ of G is a *native global state* when no perturbations are made ($e = \langle\rangle$).

We shall now prove upper and lower bounds for the number of experiments required for identifying a gene regulatory network with n genes, depending on the in-degree constraint and acyclicity. Table 11.1 summarizes the results. Computationally the running time of all algorithms when the in-degree is bounded is polynomial.

Constraints	**Lower bounds**	**Upper bounds**
None	$\Omega(2^{n-1})$	$O(2^{n-1})$
In-degree $\leq D$	$\Omega(n^D)$	$O(n^{2D})$
In-degree $\leq D$ All genes are AND-nodes (OR-nodes)	$\Omega(n^D)$	$O(n^{D+1})$
In-degree $\leq D$ Acyclic	$\Omega(n^D)$	$O(n^D)$
In-degree ≤ 2 All genes are AND-nodes (OR-nodes). No inactivation edges.	$\Omega(n^2)$	$O(n^2)$

Table 11.1: Source: [2]. Bounds on the number of experiments needed for reconstruction (n - number of genes, D - maximum in-degree). As seen from the table, forcing more constrains on the possible network topologies can improve experimental complexity significantly. The cases of acyclic topologies and restricted monotone logic (AND/OR gates only) are simpler mathematically but have no biological motivation.

11.1.3 Upper and Lower Bounds on the Number of Experiments

We first show that an exponential number of experiments are required in the worst case.

Proposition 11.1 $\Theta(2^{n-1})$ experiments must be performed in order to identify a gene regulatory network in the worst case.

Proof Consider a boolean function of $(n-1)$ variables $f(x_1, x_2, .., x_{n-1})$ which is assigned to the node x_n. There are $2^{2^{n-1}}$ possible boolean functions of $(n-1)$ variables. Hence we can identify this function by examining 2^{n-1} assignments and less examinations will not suffice (we get one output bit per experiment). ■

Proposition 11.2 $n2^{n-1}$ experiments always suffice in order to identify a gene regulatory network.

Proof For each node 2^{n-1} experiments are sufficient to identify its Boolean function by Proposition 11.1. Hence $n2^{n-1}$ experiments suffice in order to identify the whole network. ■

Theorem 11.3 *An exponential number of experiments are necessary and sufficient for the identification of a gene regulatory network.*

11.1.4 Bounded In-degree Case With Bounded Cost

Since an exponential lower bound was proven in the general case, we consider a special but practical case, in which the maximal in-degree is bounded by a constant D. First, we consider the case $D = 2$.

Proposition 11.4 $\Omega(n^2)$ experiments are necessary for identification even if the maximum in-degree is 2 and all nodes are AND nodes, where we assume that the maximum cost is bounded by a fixed constant C.

Proof First, consider the case of $C = 2$. Assume that $\neg x \wedge \neg y \rightarrow z$ is assigned to z and all other nodes have in-degree 0. Among all experiments only $(\neg x, \neg y)$ can activate z. Therefore, we must test $\Omega(n^2)$ pairs of nodes in order to find (x, y).

Next, we consider the case of $C = 3$ with the same function $\neg x \wedge \neg y \rightarrow z$. If we disrupt or overexpress u, v, w such that $x \notin \{u, v, w\}$ or $y \notin \{u, v, w\}$, we can only learn that $(u, v), (u, w), (v, w)$ are different from (x, y). Since there are $\Theta(n^3)$ triplets and only $\Theta(n)$ triplets can include $\{x, y\}$, $\Theta(n^2)$ triplets must be examined in the worst case (each experiment removes at most a constant number of pairs out of the $\Theta(n^2)$ possible ones).

For cases of $C > 3$, similar arguments work: suppose $C = k > 3$, if we disrupt and/or everexpress $u_1, ..., u_k$ such that $x \notin \{u_1, ..., u_k\}$ or $y \notin \{u_1, ..., u_k\}$, we can only know that $\frac{k!}{2! \cdot (k-2)!}$ pairs are different from (x, y). Since there are $\Theta(n^k)$ k-mers and only $\Theta(n^{k-2})$ k-mers can include $\{x, y\}$, $\Theta(n^2)$ triplets must be examined in the worst case (each experiment removes at most a constant number of pairs out of the $\Theta(n^2)$ possible ones). ■

If C is not bounded, the above proposition does not hold. It is possible to identify the above pair (x, y) by $O(\log(n))$ experiments of maximum cost n, using a strategy based on binary search. Although this strategy might be generalized for other cases, we do not investigate it because experiments with high cost are not realistic. (The cells simply die if they are heavily mutated.)

Next, we consider the upper bound.

Proposition 11.5 $O(n^4)$ experiments with maximum cost 4 are sufficient for identification if the maximum in-degree is 2.

Proof We assume (w.l.o.g.) that all nodes are of in-degree 2 since identification of nodes of in-degree 1 or 0 is easier. Let c be any node of V. We examine all assignments to all quadruplets $\{a, b, x, y\}$ with $c \notin \{a, b, x, y\}$. The boolean function $g(a, b)$ is assigned to c (i.e., $f_c \equiv g$) if and only if $c \equiv g(a, b)$ for any assignment to $\{a, b, x, y\}$, where $c \equiv g(a, b)$ means that the *state* of c equals to $g(a, b)$. The 'only if' part is trivial. We shall prove the 'if'

part. Suppose that $g(a,b)$ is not assigned to c, i.e., $f_c = h(a,b)$ and $h(a,b) \neq g(a,b)$. Clearly, $c \equiv g(a,b)$ does not hold. Next, consider the case where $h(p,q)$ is assigned to c where h may be equal to g and $\{p,q\} \cap \{a,b\} = \emptyset$. In this case, c takes both 1 and 0 by changing assignments to $\{p,q\}$ even if the assignment to $\{a,b\}$ is fixed. Therefore, $c \equiv g(a,b)$ does not hold. In the case remaining $\{p,q\} \cap \{a,b\} \neq \emptyset$. Suppose $f_c \equiv h(p,b)$ and $a \neq p$. Then there is a value of b so that $h(0,b) \neq h(1,b)$, but then $f_c(a,b,p=0,y) \neq f_c(a,b,p=1,y)$ and $c \equiv g(a,b)$ does not hold again. Since all assignments to all quadruplets are examined, in total $0(n^4)$ experiments are sufficient. ■

The above property holds even for an *unstable* graph because c is consistent under any experiment on $\{a,b,x,y\}$ if $f_c \equiv g(a,b)$.

Theorem 11.6 *$O(n^{2D})$ experiments with maximal cost $2D$ are sufficient for the identification of a gene regulatory network of bounded in-degree D. On the other hand, $\Omega(n^D)$ experiments are necessarily in the worst case if the cost of each experiment is bounded by a constant.*

11.1.5 Efficient Strategies for Special Cases

In this section we consider the case where the network consists of AND and/or OR nodes. In this case we assume that any AND (resp. OR) node c is *inactive* (resp. *active*) if at least one literal appearing in the boolean function assigned to c is forced to 0 (resp. 1) by disruption or overexpression of the gene corresponding to the literal. The above assumption is biologically reasonable even when the network contains inconsistent nodes.

Theorem 11.7 *A gene regulatory network which consists of AND and/or OR nodes and has maximum in-degree D can be identified by $O(n^{D+1})$ experiments.*

Proof Here we only show strategy for a network that consists of AND nodes of in-degree 2. It can be generalized though, to the other cases. We examine all assignments to all triplets $\{a,b,x\}$ with $c \notin \{a,b,x\}$. The function $g(a,b)$ is assigned to c (i.e., $f_c = g$) if and only if $c \equiv g(a,b)$ for any assignment to $\{a,b,x\}$. Following the proof in Proposition 11.5, we only have to consider the case that $h(p,q)$ is assigned to c and $\{p,q\} \cap \{a,b\} = \emptyset$. Consider an assignment to $\{a,b,p\}$ for which $g(a,b) = 1$. If c is not *active* we can conclude that $c \equiv g(a,b)$ does not hold. If c is *active*, we can inactivate c by changing the assignment to p since only one assignment to $\{p,q\}$ can activate c. Thus , $c \equiv g(a,b)$ does not hold. Therefore, the above property holds and $O(n^3)$ experiments are sufficient in total. ■

Next, we consider the acyclic case for which we obtain an optimal bound.

Definition A set of nodes $\{x_1, x_2, ..., x_k\}$ has *influence* on y if there exist two experiments e_1 and e_2 on $\{x_1, x_2, ..., x_k\}$ such that e_1 activates y and e_2 inactivates y.

Definition A set of nodes $\{x_1, x_2, ..., x_k\}$ has *influence* on $\{y_1, y_2, ..., y_p\}$ if $\{x_1, x_2, ..., x_k\}$ has influence on at least one of $\{y_1, y_2, ..., y_p\}$.

Definition A set of nodes $\{x_1, x_2, ..., x_k\}$ has strong *influence* on y if there exist two experiments e_1 and e_2 on $\{x_1, x_2, ..., x_k\}$ such that e_1 activates y and e_2 inactivates y, and e_1 differs from e_2 only on a single x_i.

The above definitions are invalid if the network is unstable (i.e., has an inconsistent node) or has multiple stable states. Henceforth , we assume that the network cannot have inconsistent nodes except ones that are disrupted or overexpressed. Moreover, for stable networks, we make a biologically reasonable assumption that a set of nodes $\{x_1, x_2, ..., x_k\}$ does not have influence on a node to which there is no direct path from any of $\{x_1, x_2, ..., x_k\}$.

Theorem 11.8 *An acyclic gene regulatory network of maximum in-degree D can be identified by $\Theta(n^D)$ experiments.*

Proof The lower bound directly follows from Proposition 11.4 and Theorem 11.6. We prove the upper bound only for $D = 2$. Other cases can be proved in similar way. Moreover, we only show the strategy for a node with $a \wedge b \rightarrow c$ although it can be generalized to other types of nodes. We assume (w.l.o.g.) that all nodes are of in-degree 2 as in Proposition 11.5. Let P be a set of pairs (x, y) satisfying the following conditions: c is *active* under $\langle x, y\rangle$, and c is *inactive* under the other assignments to (x, y). Then $a \wedge b \rightarrow c$ if and only if $(a, b) \in P$ and (a, b) does not have influence on any other pair $(x, y) \in P$. If $a \wedge b \rightarrow c$, then $(a, b) \in P$ must hold. Moreover, (a, b) does not have influence on any other pair in P since the network is acyclic. Conversely, if $a \wedge b \rightarrow c$ does not hold, then $(a, b) \notin P$ or (a, b) has influence on at least one node x, such that there is an edge from x to c. Therefore, we can identify the network by $O(n^2)$ experiments with maximum cost 2. ■

For cyclic networks with in-degree, 2 experiments of cost 2 do not suffice. It is possible to identify such network in some cases in $O(n^D)$ experiments. The strategy is based on detection of strongly connected components.

Proposition 11.9 Suppose each set U of nodes does not have influence on any node to which there is no path from U. Then strongly connected components can be identified by $O(n^D)$ experiments of maximum cost D for a gene regulatory network of bounded in-degree D.

Proof We prove the proposition only for $D = 2$. Other cases can be proved in a similar way. As in the previous propositions we assume (w.l.o.g.) that all nodes are of in-degree 2. Let $G(V, E)$ be the underlying directed graph of a gene regulatory network. Examining all assignments to all pairs, we can identify a set of pairs $InfTo(x)$ such that each pair has *strong influence* on x. Let $E' = \{(y, x) | (y, z) \in InfTo(x)$ for some $z\}$. Clearly, $E \subseteq E'$. Moreover, for any edge $(x, y) \in E'$, there is a directed path from x to y in $G(E, V)$. Formally, $G'(E', V)$

is the transitive closure of $G(E,V)$. Therefore, strongly connected components of $G'(E',V)$ coincide with those of $G(E,V)$, and thus we can identify strongly connected components of $G(E,V)$ by computing those of $G'(E',V)$. ■

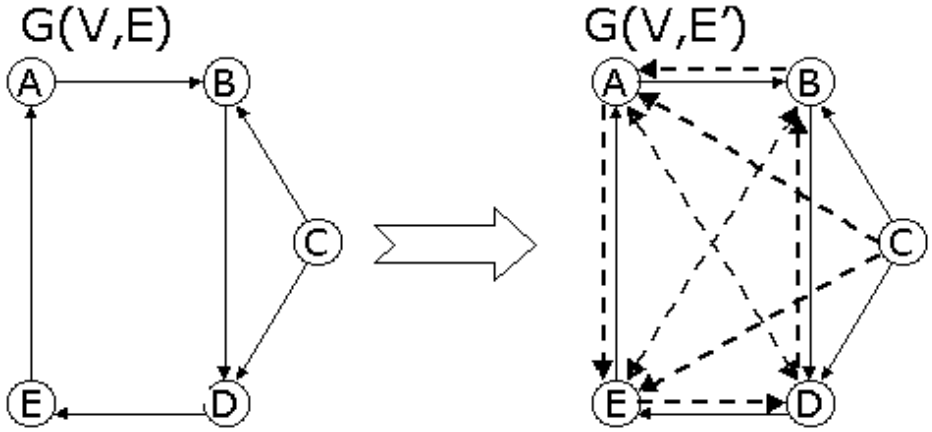

Figure 11.3: *Example of constructing E' from E (Proposition 11.9)*
InfTo(A)={(B,C),(B,D),(B,E),(C,D),(C,E),(D,E),
InfTo(B)={(A,C),(A,D),(A,E),(C,D),(C,E),(D,E)},
InfTo(C)={},
InfTo(D)={(B,C),(B,A),(B,E),(C,A),(C,E),(A,E)},
InfTo(E)={(B,C),(B,D),(B,A),(C,D),(C,A),(D,A)}.

Theorem 11.10 *If all nodes of indegree at most* 2 *and all edges are activation edges , a gene regulatory network can be identified by* $O(n^2)$ *experiments .*

Proof First we identify strongly connected components $C_1, ..., C_S$ such that $|C_i| > 2$, using proposition 11.9. Note that, treating a node with no incoming edge as a component, we can assume that there is no such node. Moreover, we can assume (w.l.o.g.) that all nodes are of indegree 2. For each C_k, we can identify a set of edges E_k as in proposition 11.9 which consist of edges in C_k and edges on the paths from C_k to the other components (because the network satisfies the condition in proposition 11.9). Let W be the set of end-points of such edges. Let p be an arbitrary fixed nod in C_k. We examine assignments to $\langle a, b, \neg p\rangle$ to all $\{a, b\} \subseteq W$. Let $P(a,b) = \{x|\ x$ is active under$\langle a, b, \neg p\rangle\}$. Then, the following property holds: $a \wedge b \rightarrow c$ if and only if $c \in P(a,b)$ and $(\forall (x,y) \neq (a,b))((c \in P(a,b)) \Rightarrow (x \notin P(a,b) \vee y \notin P(a,b)))$. Here, we only prove 'only if' part of this property because 'if' part can be proved in a similar way. Since $a \wedge b \rightarrow c$ holds, $c \in P(a,b)$ must hold. We assume that there exists another pair (x,y) such that $c \in P(x,y)$. Since all nodes are AND nodes and all edges are activation

edges, we can assume (w.l.o.g.) that there must exist a path from x to a or b. Since there must exist a path from p to x, the following two cases should be considered (see figure 11.4)

i There exists a simple path from p to x not including a or b,

ii Every simple path from p to x includes a or b.

Then, $x \notin P(a,b)$ must hold in case (i), and $c \notin P(x,y)$ must hold in case (ii). Therefore, $(\forall(x,y) \neq (a,b))((c \in P(a,b)) \Rightarrow (x \notin P(a,b) \vee y \notin P(a,b)))$ must hold. Using this property, we can identify edges in E_k for all k. Let n_k be the number of edges appearing in E_k. Since each node is counted by at most three components (by means of indegree bound), $\sum n_k = O(n)$ holds. Since $O(n_k^2)$ experiments are done for each C_k, $O(n^2)$ experiments are done in total. ■

11.1.6 Heuristic Strategy for the Practical Case

In previous sections an $\Omega(n^2)$ lower bound on the number of experiments even for the case of in-degree 2 was shown. Since $O(n^2)$ experiments are impossible for *yeast* ($n = 6200$), this suggests that we cannot expect a single strategy identification method for the gene regulatory network of *yeast* and that the methods from the previous section should be employed only for determining the local network structure. This leads us to develop a strategy by which we can identify as many parts as possible using $O(n)$ experiments. In such a case, we should find a set of edges E' such that $E' \subseteq E$. That is , we should find a set of edges without *false positive edges*.

Let $InfFrom(x)$ denote the set of nodes influenced by x (excluding x itself). Note that $InfFrom(x)$ is not necessarily determined uniquely because of inconsistent nodes. Only such nodes x for which $InfFrom(x)$ is determined uniquely will be considered.

Proposition 11.11 If $InfFrom(b) \cup \{b\} = InfFrom(a)$ holds and there is no cycle including node b, then the edge (a,b) appears in the gene regulatory network.

Proof Suppose that both conditions hold but the edge (a,b) does not appear in the gene regulatory network. Since $b \in InfFrom(a)$ holds from $InfFrom(b) \cup \{b\} = InfFrom(a)$, there must exist simple path from a to b which includes at least another node x such that $x \in InfFrom(a)$. Since there is no cycle including b and there is a path from x to b, $x \notin InfFrom(b)$ holds, a contradiction. ■

Note that the condition that there is no cycle including b cannot be removed from the above proposition. For example, $InfFrom(b) \cup \{b\} = InfFrom(a)$ holds in both networks of Figure 11.5, but in each case the edge (a,b) does not necessarily exist. Note, that three nodes satisfy $InfFrom(b) \cup \{b\} = InfFrom(a)$ in case (i) while only one node satisfies this condition in case (ii). Although testing the existence of a cycle may require an exponential

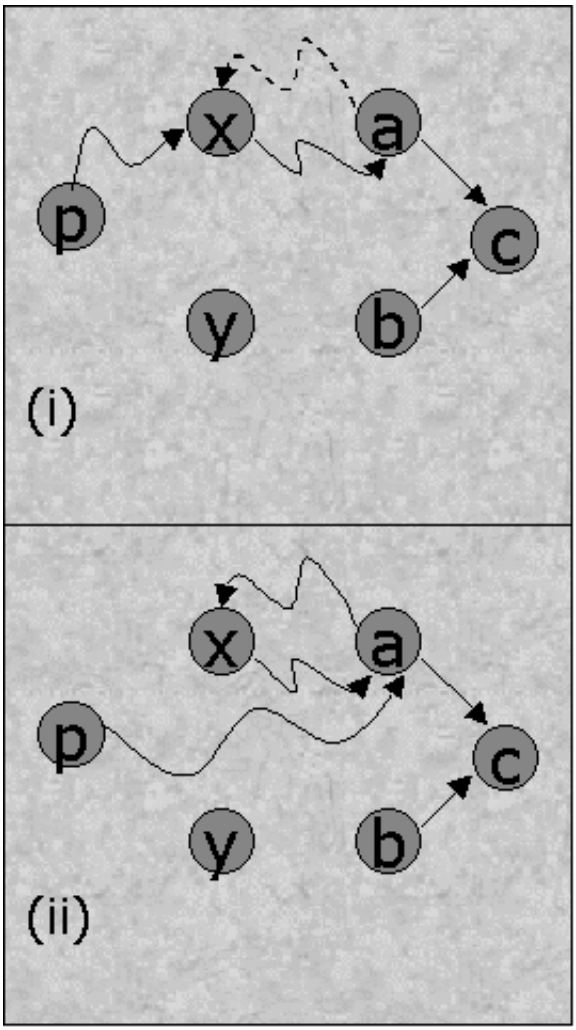

Figure 11.4: Two cases considered in Theorem 11.10.

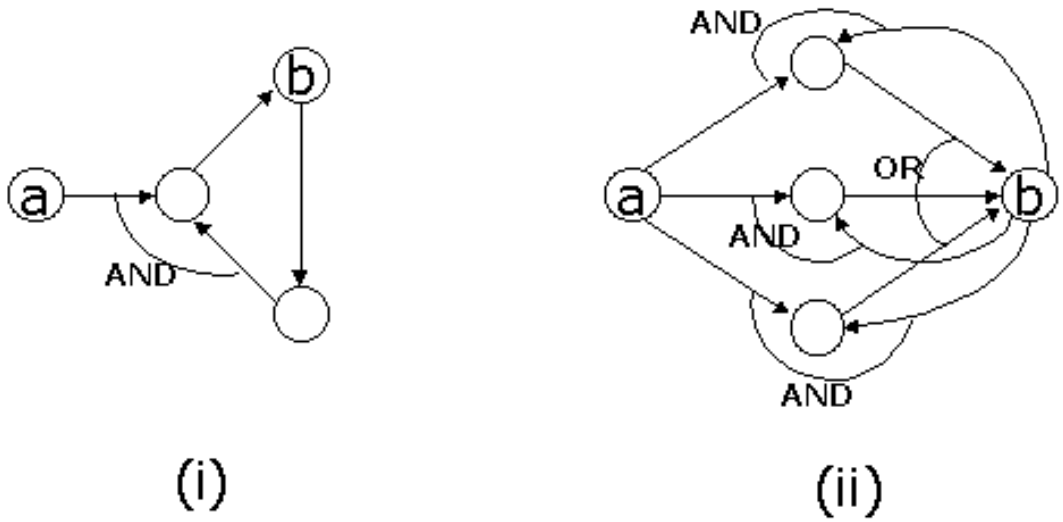

Figure 11.5: Source [2]. These cases satisfy $InfFrom(b) \cup \{b\} = InfFrom(a)$ but do include cycles passing through node b.

number of experiments as in Proposition 11.1, it is expected that such cases as in Figure 11.5 seldom occur. Therefore, if only one node b satisfies $InfFrom(b) \cup \{b\} = InfFrom(a)$, we may conclude that edge (a, b) appears in the network. Moreover, if we can identify the set of edges incoming to b (by such a method as above), we can identify the boolean function assigned to b by examining assignments only to incoming nodes.

11.1.7 Related Problems: Consistency and Stability of Networks

Along with the identification of the gene regulatory network, there exist several important problems. Here we observe two of them.

1. The consistency problem: given a network $G'(V', F')$, check whether or not this network coincides with an underlying gene regulatory network $G(V, F)$, that is not given explicitly.

   Theorem 11.12 *Exponential number of experiments are necessary and sufficient to check the consistency of a given gene regulatory network.*

2. The stability problem: given a network $G(V, F)$, check whether or not it is stable (in a native state), i.e., there is a global state consistent with all gene regulation rules.

   Theorem 11.13 *Testing the stability of a given gene regulatory network under an experiment is* NP-complete.

11.2 Interactive Inference and Experimental Design

This section is based on a paper Iddeker, Thorsson and Karp [6]. 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. Moreover, we shall restrict ourselves only to acyclic networks. In the case of acyclic networks there is no need for assumptions about the time delays of the components. Moreover, 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.

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. Accordingly, 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 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.

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 [6] 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. The following figures 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.

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.

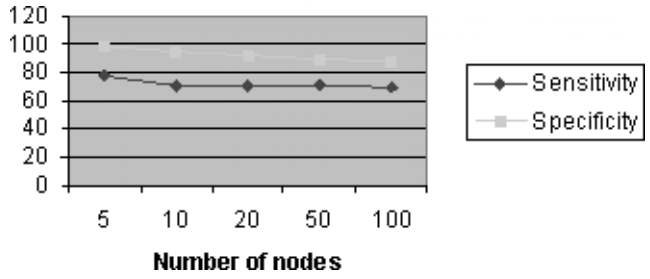

Figure 11.6: Sensitivity and specificity in percents vs. number of nodes.

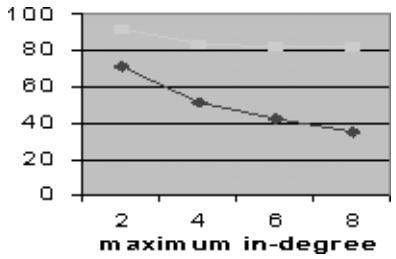

Figure 11.7: Sensitivity and specificity vs. maximum in-degree.

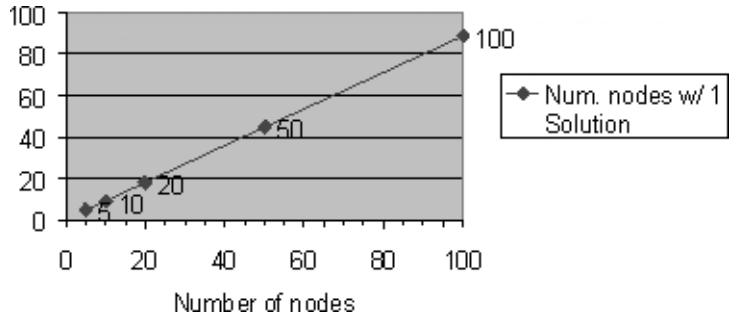

Figure 11.8: Percentage of networks with one solution.

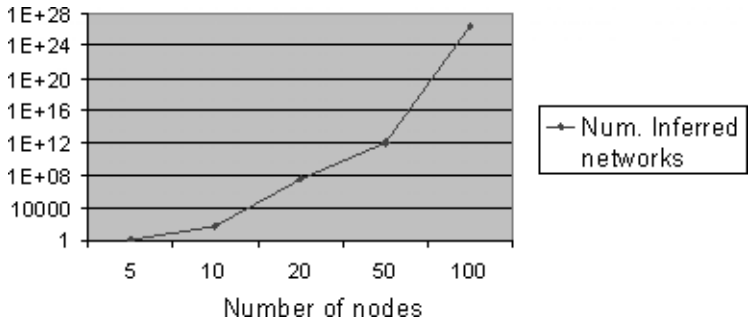

Figure 11.9: Number of inferred networks vs. number of nodes.

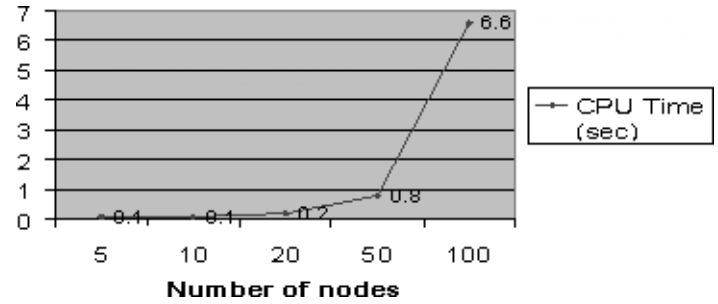

Figure 11.10: CPU time vs. number of nodes.

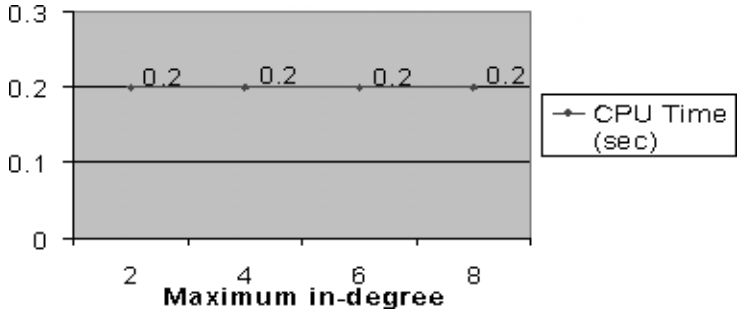

Figure 11.11: CPU time vs. maximum in-degree.

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 11.12: Summary of predictor evaluations.

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 11.13. They show a pattern of jumps and decays in the number of network solutions, correlated with an increase in the number of inferred edges.

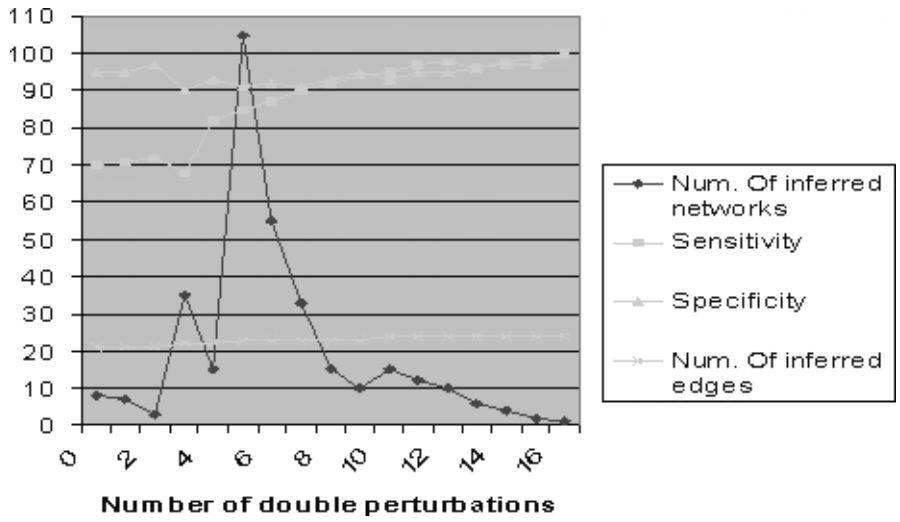

Figure 11.13: Source: [6]. Summary of chooser evaluation: progress through experimental design.

Lower Bound Comparison

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

Theorem 11.14 ([5]) *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 11.14, indicate a logarithmic behavior.

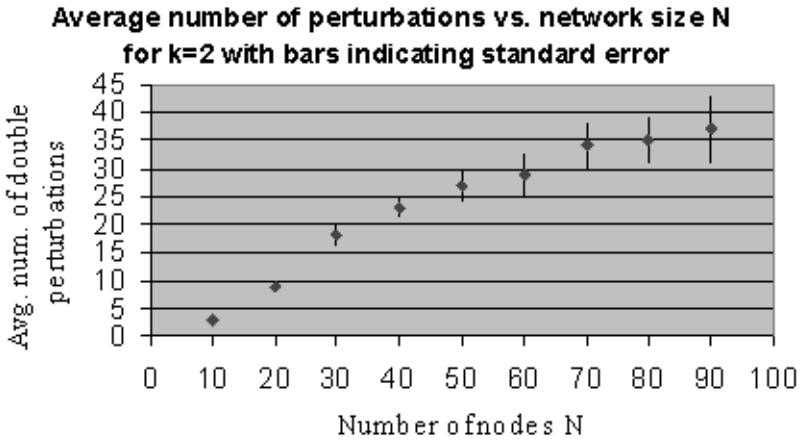

Figure 11.14: Source: [6]. Average number of perturbations vs. network size.

11.3 Computational Expansion of Genetic Networks

This section presets a work of Tanay and Shamir [1]. In this paper the authors suggest a new methodology for computational analysis of gene and protein networks. The aim is to generate new educated hypotheses on gene functions and on the logic of the biological network circuity, based on gene expression profiles. The framework supports the incorporation of biologically motivated network constrains and rules to improve specificity. Since current data is insufficient for de-novo reconstruction, the method receives as input a known pathway core and suggest likely *expansions* to it. Network modeling is combinatorial, yet data can be probabilistic. At the heart of the approach are a fitness function which estimates the quality of suggested network expansions given the core and the data, and the specificity measure of the expansions. The approach has been implemented in an interactive software tool called GENESYS. Figure 11.15 demonstrates some screens from the software.

11.3.1 The Main Idea

Since the goal is to choose the best network, we will define the space of all valid networks and a fitness function which tells us how well does a network fit all data sources. Then we will apply a heuristic search algorithm on the space, guided by the fitness function. As we saw earlier it is impractical to search over the whole space.

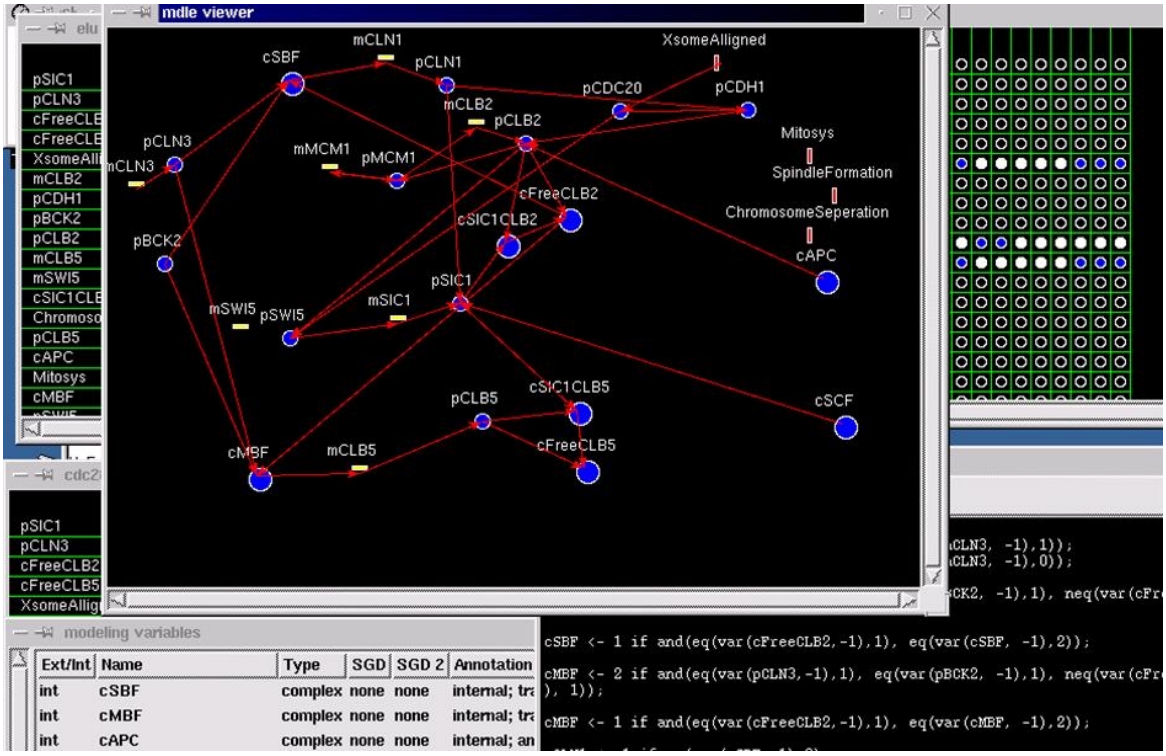

Figure 11.15: Demonstration of GENESYS' screens. The upper window displays the wiring diagram. The right-bottom window contains the functions between the genes and the left-bottom screen displays some data about the proteins/mRNAs.

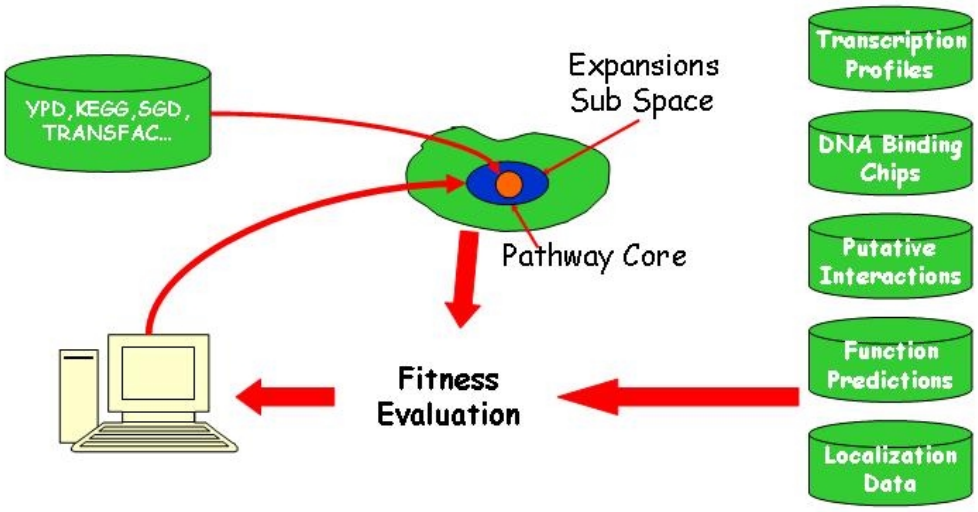

Figure 11.16: A known pathway core is being taken from Database such as KEGG [7]. Valid expansions are being evaluated according to known experiments.

The Practical Approach

The starting point of the process is a pathway core, which represents prior knowledge on a particular biological sub-system. A combinatorial search algorithm suggests the most promising core expansions, in light of their level of fitness to a given experimental dataset. Currently there is a restriction of choosing pathways driven mostly by transcription, since this is the experiments data that was available to the authors. Through studying sub-systems (Cell-cycle, Mating, etc.) and the connections between them we hope to infer eventually, the whole network. Figure 11.16 presents the general scheme.

11.3.2 Model Assumption

The model is synchronous (as in previous works [2, 6]) and has three (instead of boolean) discrete states $(-1, 0, 1)$. A biological network N is defined by a set U of variables (genes, proteins, mRNAs, etc.), a set C of values or states that the variables may attain and a set of functions F where $F \equiv f^v : C^{|U|} \rightarrow C \ \forall v \in U$. We denote by $arg(f^v)$ the set of non trivial arguments of f^v (u is a trivial argument of f^v if changing the value of u alone never alters the value of the function). The dependency graph (wiring diagram) is $G(N) \equiv (U, A)$ where $(u, v) \in A \Leftrightarrow u \in arg(f^v)$.

Since we will be searching for the "best" network, we need to describe the search space: A *model space* is defined by the four-touple (U, C, F_{bio}, G_{bio}) where U and C are as above, $F_{bio} \subseteq \{f : C^{|U|} \rightarrow C\}$ is the class of candidate functions and G_{bio} is a class of dependency

graphs on U. The space consists of all networks with functions from F_{bio} and dependency graphs from G_{bio}. F_{bio} and G_{bio} are used to limit the model space, by incorporating biological knowledge and realistic constraints. F_{bio} constrains the properties of each particular function. For example, $MONO_d$ is the set of monotone functions with at most d inputs. G_{bio} constrains the overall architecture of the network. For example, the diameter of the graph.

Modeling Experimental Data

One of the problems that was ignored in the previous works we have presented, is that expression profiles are very noisy. The authors are coping with this issue by translating each reading into a distribution over the discrete value space as can be seen in Figure 11.17.

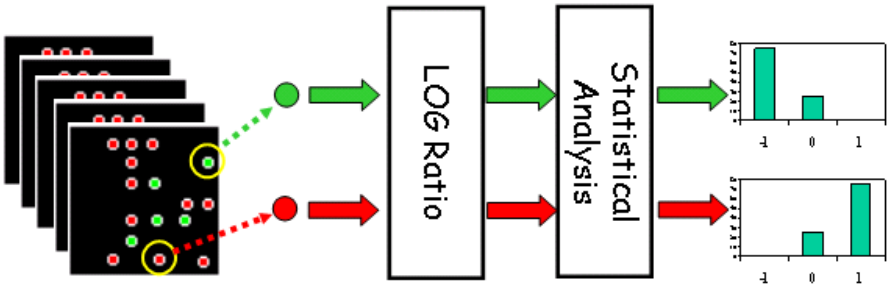

Figure 11.17: The chip's experiment data is being translated into logarithmic scale and undergoes a statistical analysis phase. For example, the upper gene has value -1 with probability 0.7 and value 0 with probability 0.3.

By an *experiment* we mean a triplet $(INP, OUT, PERT)$ where INP and OUT are the input and output vectors, assigning values to each variable in U. $PERT \in U$ is the set of perturbed variables, i.e., those genes that were knockedout or overexpressed. Hence, a knockout or overexpressive experiment will produce one triplet.

A perfect experiment is where $INP, OUT : U \rightarrow C^{|U|}$, while in noisy experiment $INP, OUT : U \rightarrow (2^C)^{|U|}$. Timeseries data, providing expression levels at a series of n time points, yields $n-1$ experiment triplets, where the vectors at time points i and $i+1$ form INP and OUT of the ith experiment. If in an experiment $INP = OUT$, we say it is a *steady state experiment*. We also use the notation $INP_v(c)$ which is the probability that v attains the value $c \in C$ in the input.

As mentioned before our goal is to infer biological pathways by finding an expansion network or digraph that fit the experimental data best. This must be preceded by developing a good fitness function. One of the key ingredients in our computing fitness is the *Consistency* with experimental data.

11.3.3 Consistency

Under the assumption of our model we expect to see a consistency in the data, meaning that identical input values should yield the same output value. Figure 11.18 demonstrates this idea.

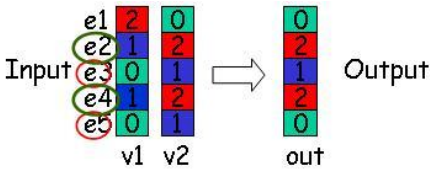

Figure 11.18: Experiments $e2$ and $e4$ are consistent while experiments $e3$ and $e5$ are inconsistent.

According to previous works such [2], the inconsistent experiments will indicate that the experiment's argument $e \notin arg(f^v)$. But since results are noisy the *consistency of dependency structure* is defined as the maximal number of consistent experiments:

$Consist(v, \varphi, E) = |\{e \in E, v \in PERT_e | \varphi(INP^e) = OUT_v^e)\}|$

where $E = (INP_e, OUT_e, PERT_e)_{e<n}$ is an experimental dataset and

φ is a putative regulation function.

When seeking to infer dependencies only, we define $Consist(v, S, E)$, the consistency of a set S of arguments for node v, as the maximum consistency obtained by any $f^v \in F_{bio}$ whose arguments all belong to S: $Consist(v, S, E) = max_{\varphi, arg(\varphi) \subseteq S}[consist(v, \varphi, E)]$.

Those equation can be naturally generalized to probabilistic data.

Proposition 11.15 If $F_{bio} = F$, we can calculate dependency-structure consistency by optimizing the output for each input value separately, this would take $O(n + m^{d+1})$ steps for noiseless experiments. For noisy experiments we assume statistical independence to obtain similar complexity.

Though simple and easy to compute, the consistency function gives no information regarding the specificity of a speculated regulation pattern and thus it is very sensitive to over fitting. To address this problem, we shall describe how to calculate a *p-value* of the measured consistency.

11.3.4 Regulation Specificity

An example for the problem: Assume we had only fifty experiments and each gene is a combination of three other gene and each combination of this triplet values appeared once. The consistency is perfect (no conflicts in the data), but this high consistency can be random (other function could achieved the same consistency). Therefore, we would like to have a

measure of significance to the consistency score we got. As our null hypothesis, we assume independence of the measured values of the variable v and the variables in $arg(v)$. We wish to estimate the probability of observing consistency k or higher in the data under the null hypothesis (see Figure 11.19).

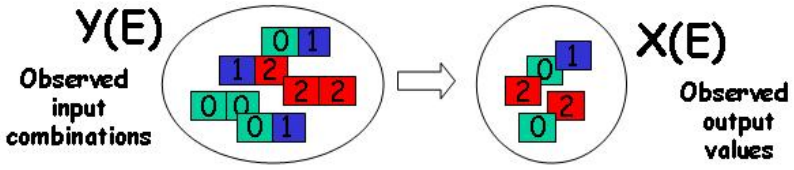

Figure 11.19: The null assumption: Y and X are independent random variables.

Thus, $rSpec(S, v, E) = Pr(Consist(v, S, (Y * X)^n) \geq Consist(v, S, E))$. In practice the computational approach to $rSpec$ is heuristic.

11.3.5 Expansion

In Figure 11.20 we can see the known biological network core and a set of candidates for the core-expansion. For each candidate we will calculate the improvement of the consistency and the specificity (rSpec) score on the base network. Eventually we will choose for expansion the candidate that achieved the best score.

11.3.6 Simulations

Simulation Setup

The structure of the simulated network was semi-hierarchical as can be seen in Figure 11.21 since we tend to believe that this is the model of gene networks. The results are shown in Figures 11.22 - 11.25.

11.3.7 Real Data - Ergosterol Metabolism

The data that the authors worked with was the Ergosterol pathways which is the yeast analogue to human cholesterol and some of the pathway enzymes function as important drug targets. The relevant data was taken from large experiments such as Hughes et al. [4] and Gasch et al. [3] ($\sim$ 360 conditions in total). Out of the $\sim$ 6200 yeast ORFs, they identified 130 putative transcription factors (TFs). For this they used SGD annotations, as well as typical structural motifs (e.g., zinc fingers). The authors then applied the single node expansion algorithm, limiting the candidates for node expansions to these putative TFs. In the first test, they ranked the fitness gain of each of the putative TFs against a 'naked' core

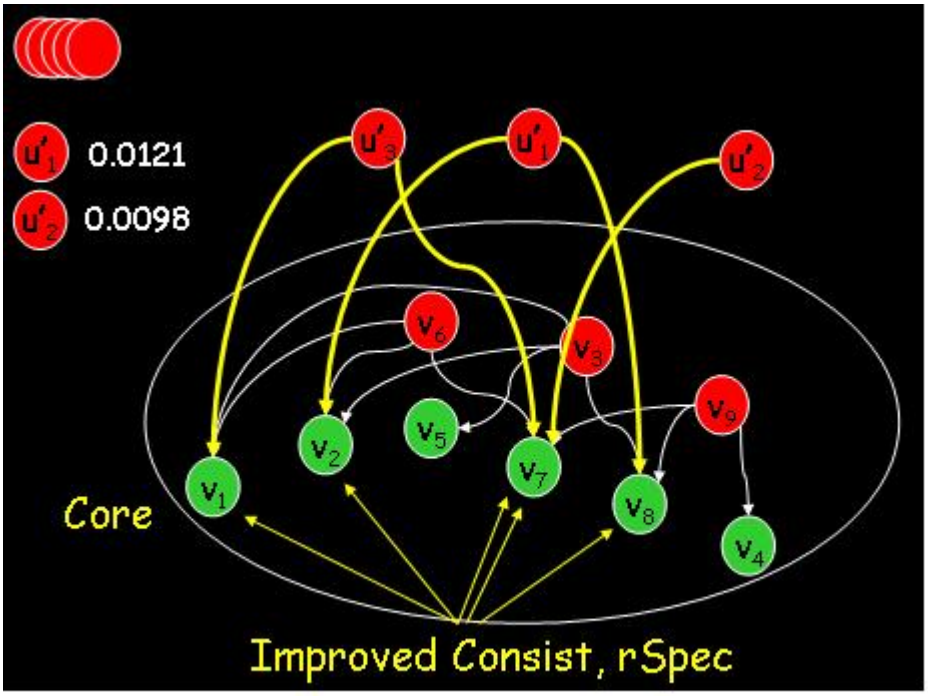

Figure 11.20: The core network contains the set $\{v_1, \ldots, v_9\}$. The improvement scores of the candidates are $u'_1 = 0.0121$ and $u'_2 = 0.0098$. u'_3 achieved the best score and therefor was added to the core as an argument of v_1 and v_7.

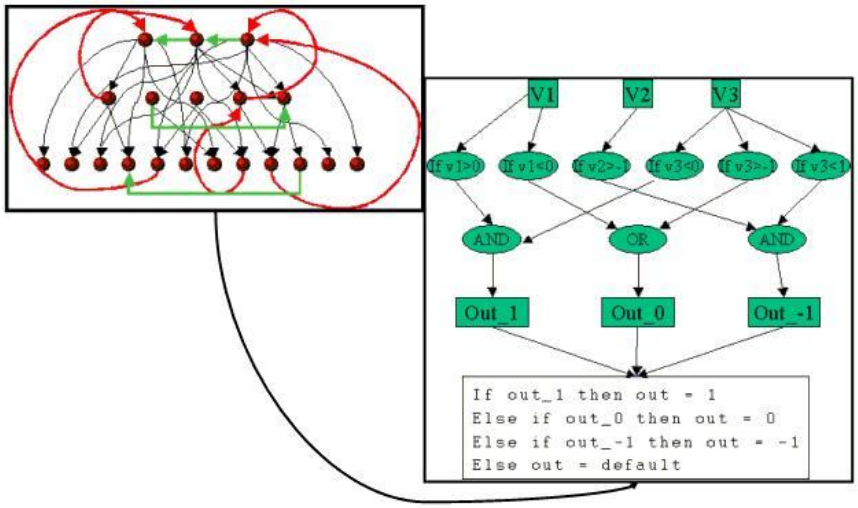

Figure 11.21: The setup of the simulated network. Few genes function as master-regulators in different layers while other genes barely affect the network.

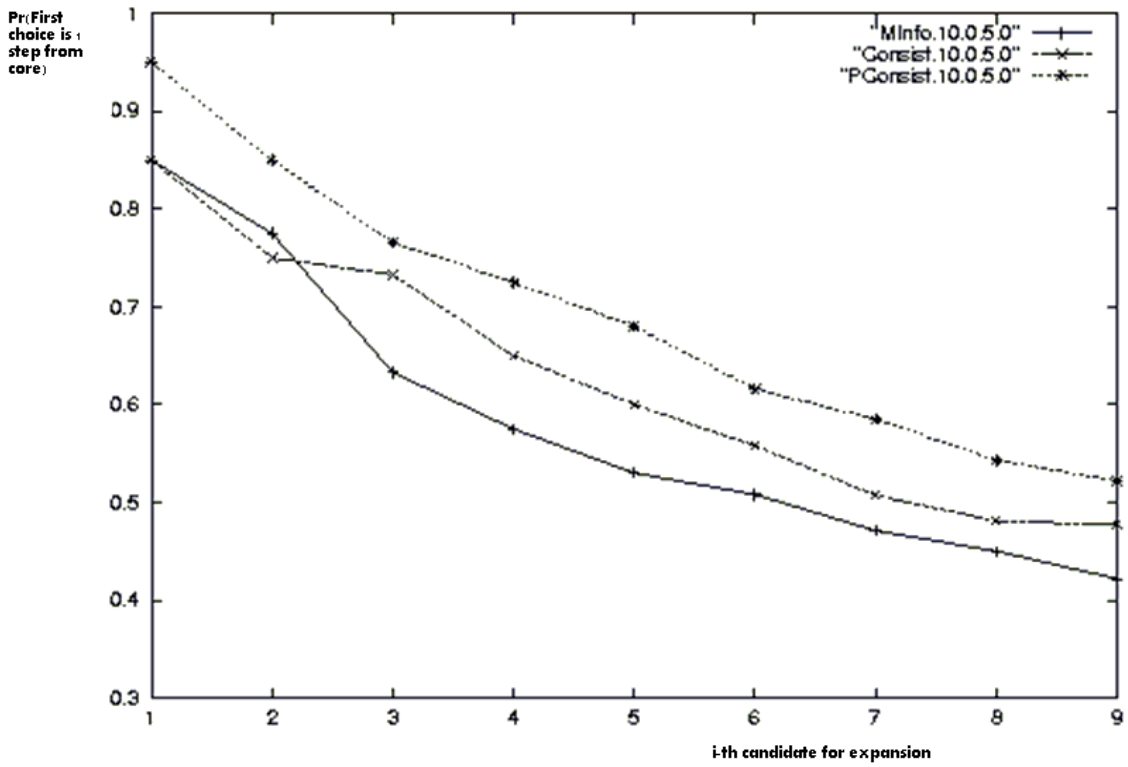

Figure 11.22: Simulation without noise: For each candidate there are three scores (consistency, specificity, mutual). The specificity (rSpec) gave better results than the consistency score.

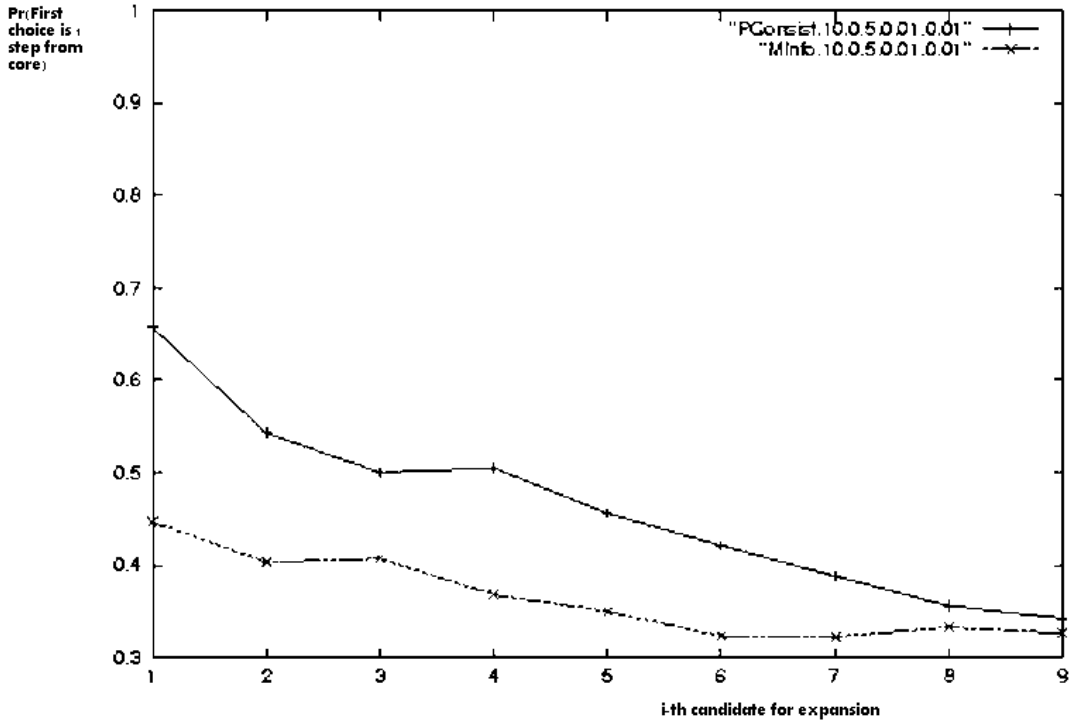

Figure 11.23: Simulation with noisy data: After adding noise to the data we can see the influence on the scores.

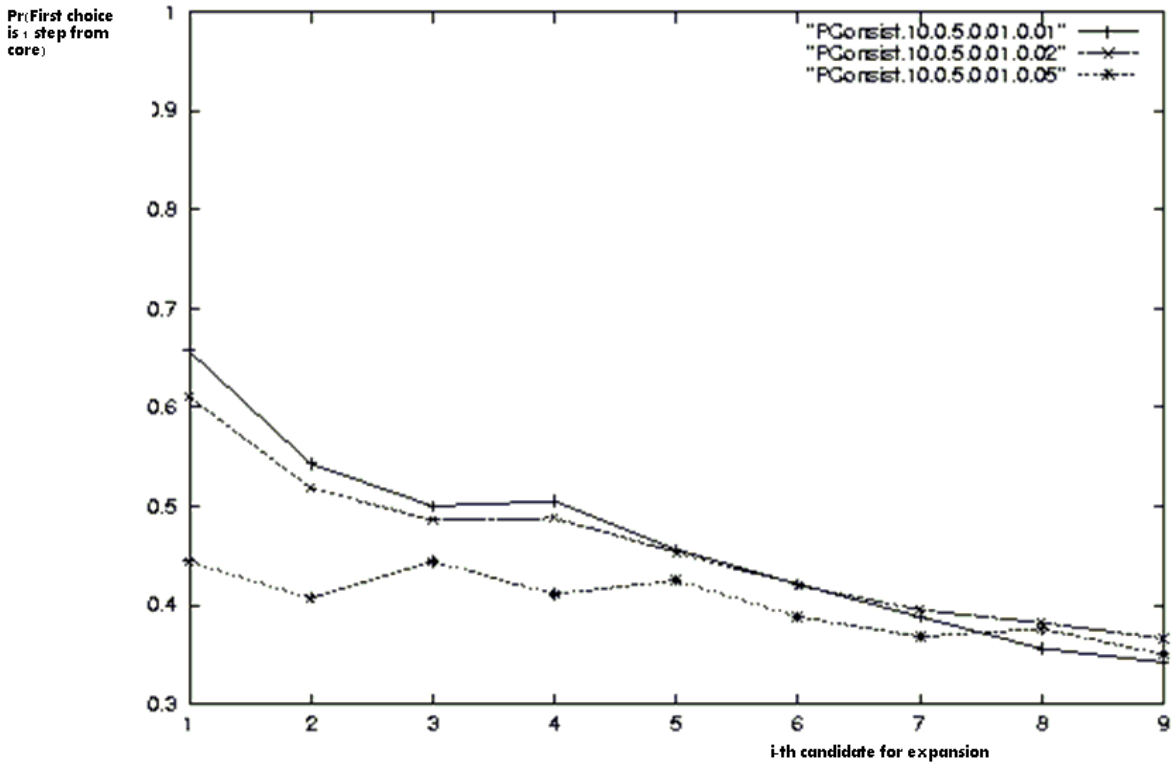

Figure 11.24: False positive impact: After changing the false positive ratio in the simulation we can see it considerably affect the results.

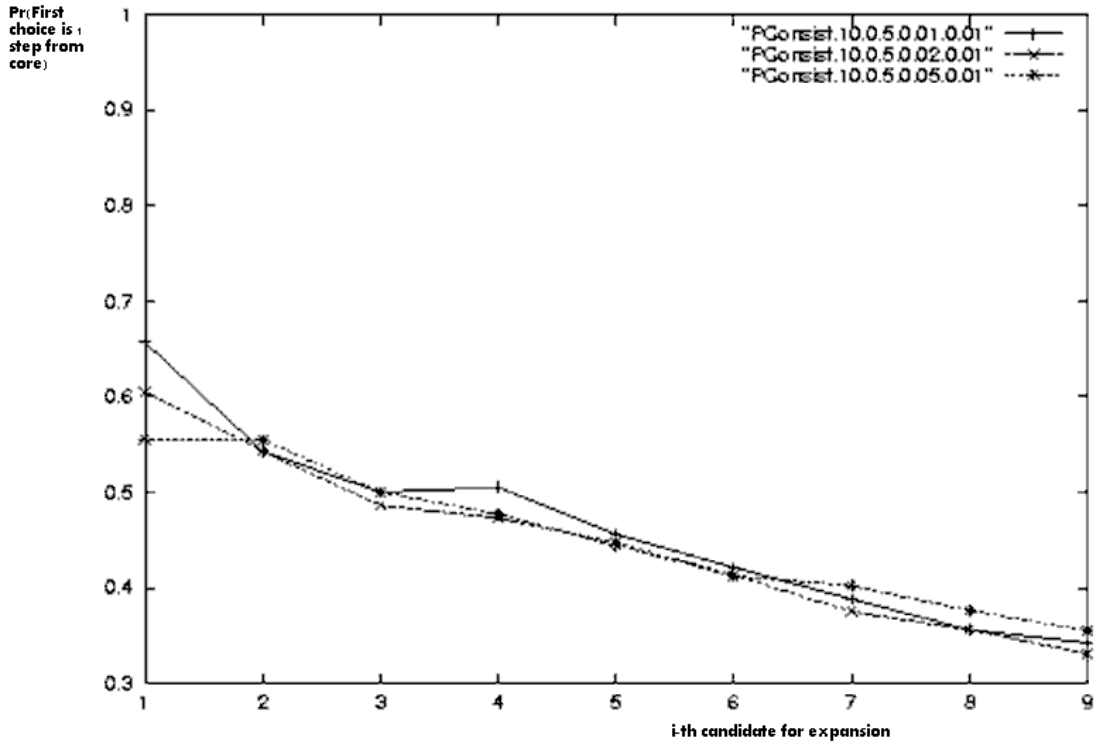

Figure 11.25: False Negative impact: The False negative impact on the system was rather low.

consisting of the eleven ERG enzymes with no dependencies among them (see Figure 11.26). HAP1 was ranked second out of 130, in agreement with the known role of HAP1 in ERG11 regulation. The next test was trying on improving the understanding of ERG11 regulation (finding the third unknown activator of ERG11). Turi and Loper [8] analyzed the promoter region of ERG11 with results that are summarized in Figure 11.27. This time they applied the single node expansion to a core consisting of the eleven ERG enzymes as well as HAP1 and ROX1 as regulators of ERG11. The algorithm measured the improvement in fitness contributed by each of the 130 TFs, and an uncharacterized gene was ranked first. That gene improves the fitness of ERG11 (and others). Remarkably, it also has a good homology to HAP1. Moreover, analyzing ERG11 logic as a function of HAP1, ROX1, TUP1 and the novel TF shows that the effect of the new putative TF on ERG11 is inductive (as expected from a UAS2 binding gene).

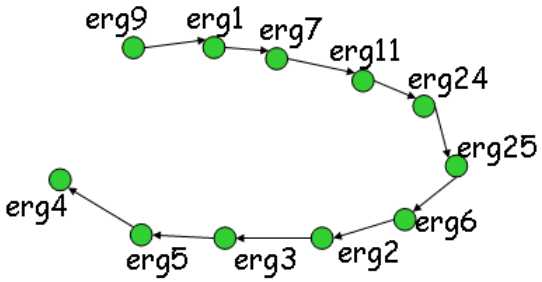

Figure 11.26: Source: [1]. Metabolical pathway of Ergosterol: Enzyme pathway, each enzyme's product is the substrate of the next one.

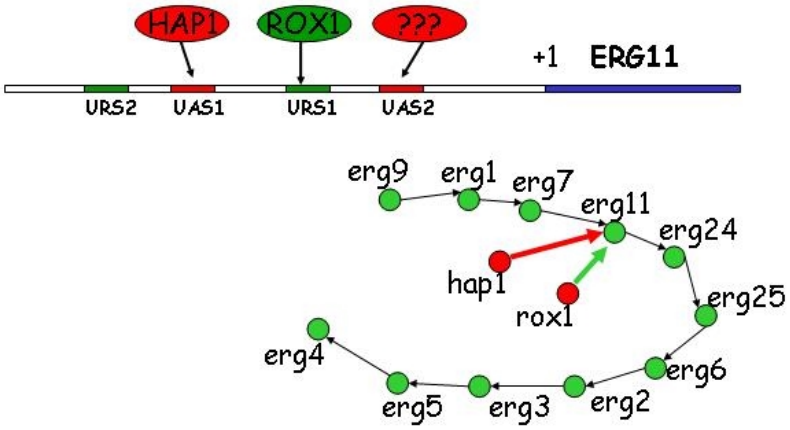

Figure 11.27: Source: [1]. ERG11 promoter region according to Turi and Loper [8]. UAS/URS: upstream activation/repression site. The transcription factors is HAP1 and ROX1 induce and repress, respectively, ERG11 transcription via the binding sites UAS1 and URS1. UAS2 was identified as a likely binding site of an unknown activator. One of the goals in this study was to demonstrate that the authors can suggest the identity of the missing activator.

Bibliography

[1] Computational expansion of genetic networks. In *Ninth Annual Conference on Intelligent Systems for Molecular Biology (ISMB 01), pp. S270–S278*, pages S270–S278, 2001.

[2] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In *Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 695–702, San Francisco, California, 25–27 January 1998.

[3] Camilla M. Kao Orna Carmel-Harel Michael B. Eisen Gisela Storz David Botstein Audrey P. Gasch, Paul T. Spellman and Patrick O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. pages 4241–57, 2000.

[4] T. Hughes et al. Functional discovery via a compendium of expression profiles. pages 109–26, 2000.

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

[6] 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.

[7] M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28:27–30, 2000.

[8] T. Turi and J. Loper. Multiple regulatory elements control expression of the gene encoding the saccharomuces cerevisiae cytochromoe p450, lanosterol 14ademethylase (erg11). pages 2046–56, 1992.