# Lecture 14: February 11, 2000

*Lecturer: Ron Shamir* *Scribe: Igor Bogudlov and Vladimir Koushnir*

## 14.1 Gene Networks

### 14.1.1 Preface

An ultimate goal of a molecular biologist is to use genetic data in order to reveal fundamental cellular processes, and their impact on complex organisms. In order to achieve this goal one has to study complex systems of several genes and proteins, which carry out a specific function, rather than single genes. Such research can help identify proteins and genes associated with human diseases, which has huge practical importance.

**Definition** A *gene network* is a set of molecular components such as genes and proteins and interactions between them that collectively carry out some cellular function.

Gene networks describe functional pathways in a given cell or tissue, representing processes such as metabolism, gene regulation, transport and signal transduction.

#### Examples

Figure 14.1 depicts gene expression and its role in catalyzing certain chemical reaction in the cell. The proB gene is being expressed into the gamma-glutamyl-kinase protein, which catalyzes a reaction involving glutamate and ATP, which produces gamma-glutamyl-phosphate and ADP compounds.

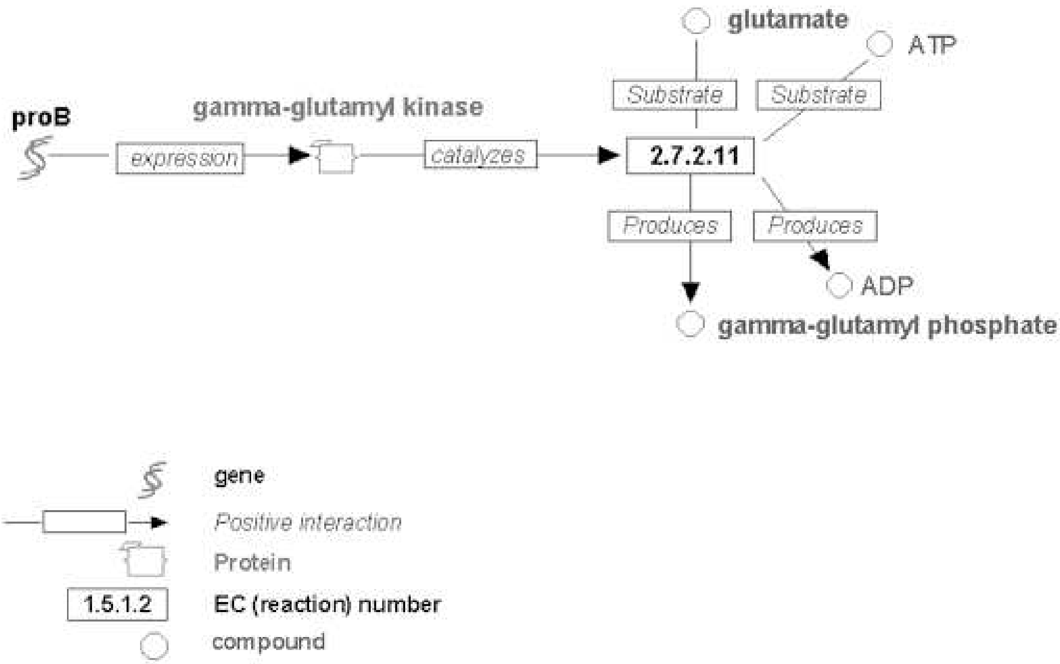



Figure 14.1: An example for the role of gene expression in catalyzing chemical reactions.

This gene expression is a part of a simple metabolic pathway, involving a chain of generated proteins, which is shown on figure 14.2. One of the final products of the chain, proline, inhibits the initial reaction, which has started the whole process. This "feedback inhibition" pattern is highly typical to genetic networks, and serves to regulate the process execution rate.

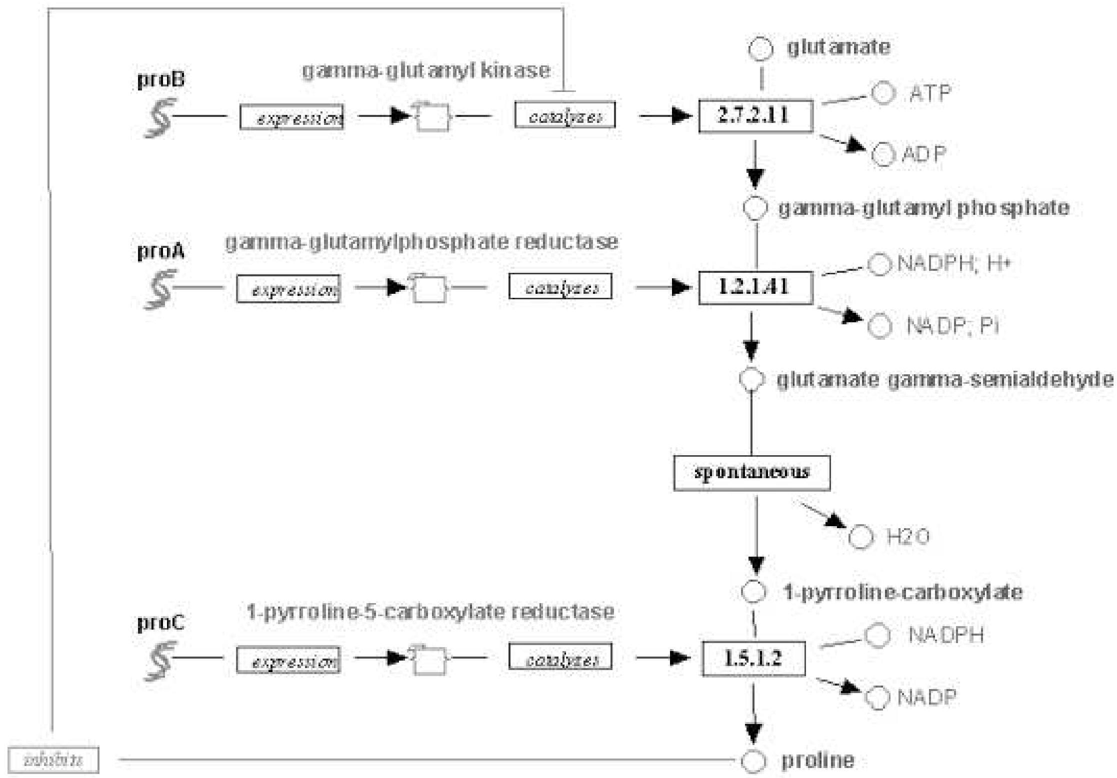



Figure 14.2: An example of an metabolic pathway: Proline biosynthesis.

The following two figures show a more complex gene network, describing Methionine biosynthesis in E-coli. The second figure is a shortcut representation of the pathway, with most nodes omitted, but it can give a better idea on overall topology.

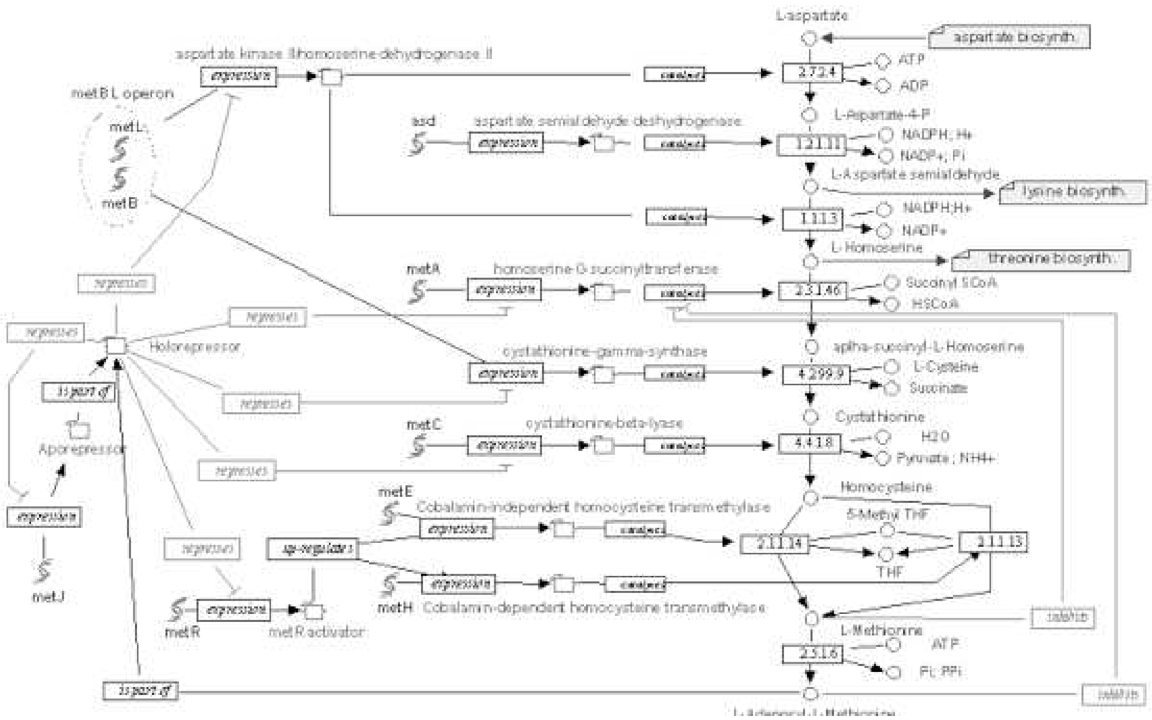



Figure 14.3: Methionine biosynthesis network in E-coli.

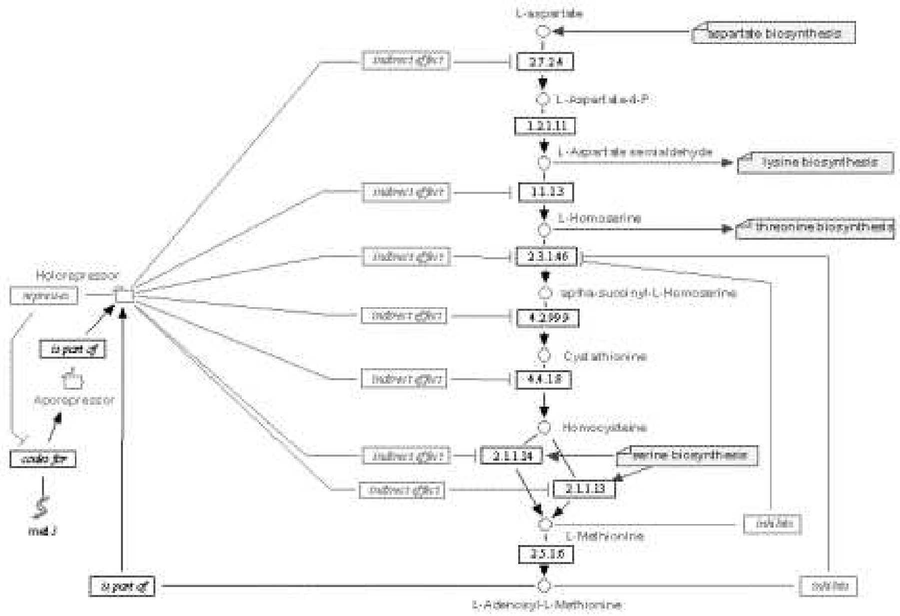



Figure 14.4: Shortcut representation of the biosynthesis pathway presented in Figure 14.3

The last example is that of signal transduction - complex cellular process initiated by signaling protein arrived from outside of a cell. This process eventually affects gene expression in the cytoplasm and inside the nucleus.

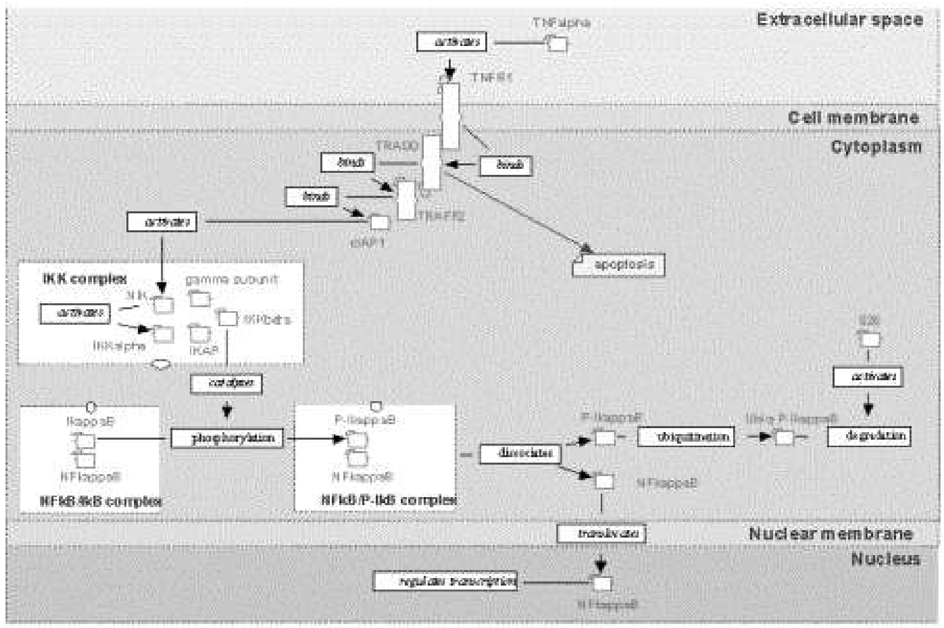



Figure 14.5: A gene network that performs signal transduction from outside the cell into the nucleus.

### 14.1.2 Functional Analysis

Using a known structure of such networks it is sometimes possible to describe behavior of cellular processes, reveal their function and the role of specific genes and proteins in them. That's why one of the most important and challenging problems today in molecular biology is that of *functional analysis* - discovering and modelling gene networks from experimental data.

#### Tools

Addressing this problem has been made possible by recent advances in genetic sequencing and development of a whole new generation of sophisticated biological tools.

The most promising technique to date is based on the view of gene systems as a logical network of nodes that influence each other's expression levels. Consequently, one may obtain some information on gene interactions in the network by measurement of gene expression. The level of gene expression, i.e., the production rate of a given protein changes during execution of a genetic process involving it and can be monitored by several biological experiments.

A variety of experimental tools have been developed recently with the ability to observe the expression of many genes simultaneously. At the forefront of these technologies lies the DNA microarray, commonly used to monitor gene expression at the level of mRNA abundance. Similarly, the rapid identification of proteins and their abundances is becoming possible through methods such as 2D polyacrylamide gel electrophoresis, 2-hybrid systems, protein chips, etc. The main contribution of all of these technologies is that numerous genes can be monitored in the same experiment, making it possible to perform a global expression analysis of the cell.

Additional information about a genetic network may be gleaned experimentally by applying a directed perturbation to the network, and observing expression levels of every gene in the network in the presence of the perturbation. Perturbations may be *genetic*, in which the expression levels of one or more genes are fixed by **knockout** (removal of the gene) or **overexpression** (higher than usual level of gene expression), or *biological*, in which one or more non-genetic factors are altered, such as a change in environment, nutrition, or a temperature increase. Such biological experiments are very costly and very few such perturbations may be performed at one time. Thus, reducing the number and cost of experiments is crucial.

Methods presented above supply biological data in terms of expression levels of many genes at different time points and in different conditions. The functional analysis of the data can be defined as a computational problem, aiming to infer some plausible model of the network from the observations with minimal number or cost of biological experiments. The model should describe how the expression level of each gene in the network depends on external stimuli and expression levels of other genes. Additional goals include construction of a knowledge-base of gene regulatory networks, verification of pathways or gene networks hypotheses.

### 14.1.3 Gene Network Models

Several models has been proposed in the literature to capture the notion of gene networks and allow mathematical solutions of the above computational problem.

- **Linear Model**
  This model, proposed by D'haeseleer et. al. [3], [2], assumes that expression level of a node in a network depends on linear combination of the expression levels of its neighbors
- **Boolean Model**
  Due to Liang, Fuhrman and Somogoyi [7]. It assumes only two distinct levels of expression - 0 and 1. According to this model a value of a node at the next step is boolean function of the values of its neighbors.
- **Bayesian Model**
  Due to Friedman et al. [6]. It attempts to give a more accurate model of network behavior, based on Bayesian probabilities for expression levels

In the rest of this document we shall concentrate only on the Boolean model. The reader is referred to the mentioned papers for more information regarding the other models.

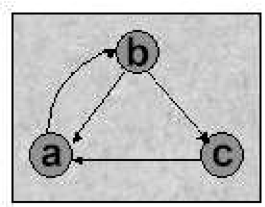
### 14.1.4 Boolean Network Model

According to this model the network is represented as an oriented graph $G = (V, F)$, whose nodes $V$ represent element of the network, and $F$ defines a topology of edges between the nodes and a set of boolean functions.

A node may represent either a gene or a biological stimulus, where a stimulus is any relevant physical or chemical factor which influences the network and is itself not a gene or gene product. A node has an associated steady-state expression level $x_v$ , representing the amount of gene product (in the case of a gene) or the amount of stimulus present in the cell. This level is approximated as high or low and represented by the binary value 1 or 0, respectively.

Network behavior over time is modelled as a sequence of discrete synchronous steps. The set $F = \{f_v | v \in V\}$ of boolean functions assigned to the nodes defines the value of a node on the next step depending on values of other nodes, which influence it. The functions $f_v$ are uniquely defined using truth tables. An edge directed from one node to another represents the influence of the first gene or stimulus on that of the second, so that the expression level of a node $v$ is a Boolean function $f_v$ of the levels of the nodes in the network which connect (have a directed edge) to $v$. We sometimes confuse the network with its undirected underlying graph $G(V, E)$.

The following figures 14.6 give an example of a simple boolean network and associated truth tables. This example shows a network of three nodes - $a$, $b$ and $c$. As one can see, expression of $c$ directly depends on expression of $b$, which directly depends on $a$. However, influence of $b$ and $c$ on $a$ is more complex. For example, high level of expression of both $b$ and $c$ leads to inhibition of $a$.




| a | b |
|---|---|
| 0 | 0 |
| 1 | 1 |

| b | c |
|---|---|
| 0 | 0 |
| 1 | 1 |

| b | c | a |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

Figure 14.6: Sample boolean network – adapted from [8]

The assignment of values to nodes fully describes the *state* of the model at any given time. The change of model state over time is fully defined by the functions in $F$. Initial assignment of values uniquely defines the model state at the next step, and, consequently, on all the future steps. Thus, the network evolution is realized as a sequence of consecutive states, and it is uniquely defined by the initial state. Such a sequence of states is called *trajectory*.

Figure 14.7 shows two such trajectories for our sample network. Since the number of possible states is finite, all trajectories eventually end up in single *steady state*, or a cycle of steady states.

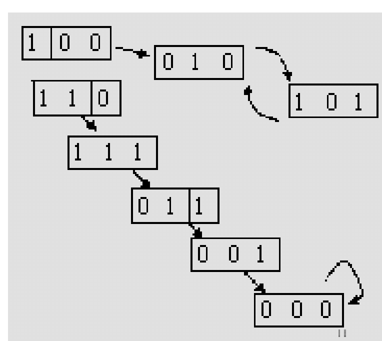



Figure 14.7: States trajectories – adapted from [8]

**Definition**

- An *attractor* of a trajectory is a single steady state, or a cycle at the end of the trajectory.
- *The basin of attraction* for a specific attractor is a set of all trajectories leading to it.

The network in our example has two attractors - one is the steady state $(0, 0, 0)$, and the other is a cycle $(0, 1, 0) \leftrightarrow (1, 0, 1)$.

States in gene networks are often characterized by *stability* - "slight" changes in value of a few nodes do not change the attractor. Biological systems are often *redundant* to ensure that the system stays stable and retains its function even in presence of local anomalies. For example, there may be two proteins, or even two different networks with the same function, which backup each other.

**Expression Matrix**

In order to infer a boolean genetic network, a population of cells containing a target genetic network is monitored in the steady state over a series of $M$ experimental perturbations. In each perturbation $P_m$ $(0 \leq m < M)$ any number of nodes may be forced either to a low or a high level. Genes may be perturbed through laboratory methods for gene deletion or overexpression, while stimuli are perturbed by altering the environment outside the cell. The observed steady-state expression levels for all genes and stimuli over all experimental perturbations or over time are represented by the *expression matrix* $E$.

Figure 14.8 shows an expression matrix generated by several illustrative perturbations. Rows of $E$ represent perturbation conditions while columns represent node values in each steady-state condition, such that matrix entry $E_{ij}$ is the expression level of node $X_i$ in the presence of perturbation $P_j$. The symbols "+" and "-" are used to show that a node has been forced to a high or low value, respectively.

| X0 | X1 | X2 | X3 | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | P0 |
| - | 1 | 0 | 1 | P1 |
| 1 | - | 0 | 0 | P2 |
| 1 | 1 | - | 1 | P3 |
| 1 | 1 | 1 | + | P4 |

Figure 14.8: Expression Matrix - adapted from [5]

## 14.2 Identification of Gene Regulatory Networks by Gene Disruptions and Overexpressions

### 14.2.1 Preface

This section is based on the article of Akutsu et al [1]. 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.

### 14.2.2 Model description and definitions

We define the gene regulatory network as in Section 14.1.4. 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).

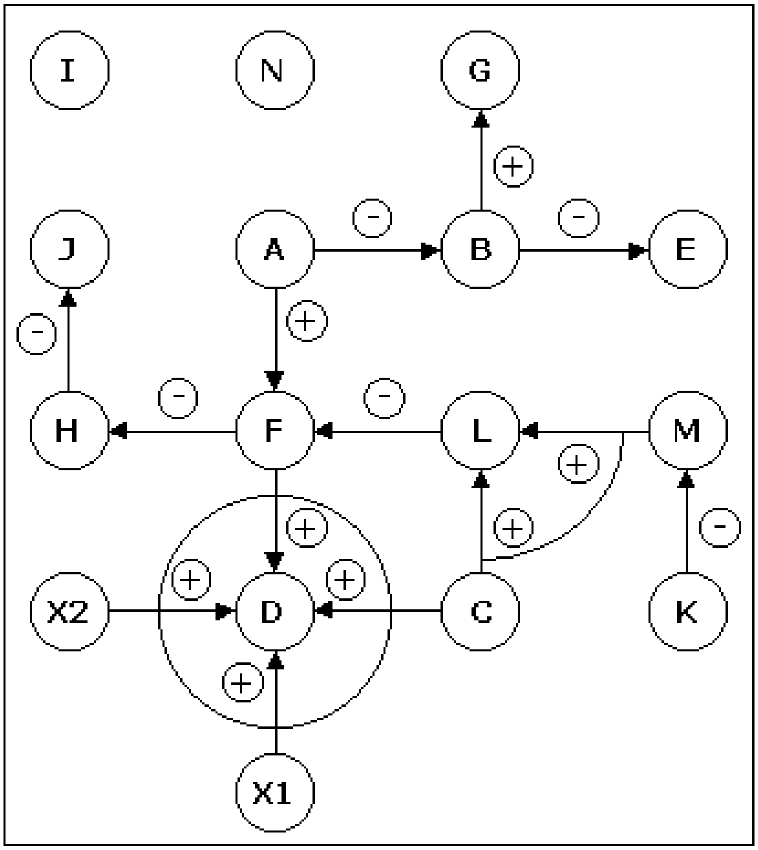



Figure 14.9: Example of gene regulatory network with 16 genes ( $\oplus$ means "activation" and $\ominus$ means "deactivation" of the gene). Gene $F$ is *activated* by gene $A$ and is also *inactivated* by gene $L$ ($f_F(A, L) = l(A) \wedge \neg l(L)$). For gene $D$, it expresses if its all predecessors $C, F, X1, X2$ express ($AND$ - 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 14.10: Gene expressions by disruption and overexpression from the gene regulatory network Fig. 14.9 (0 - the gene is not expressed , 1 - the gene is expressed).

**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 $\neg u_i$.

**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 *a 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 by the number $p + q$. Three cases of gene expression conditions (normal, disruption of A, overexpression of gene B ) are presented in Fig. 14.10.

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

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

- $v$ belongs to $e$.
- $v$ 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 $\psi : V \rightarrow \{0, 1\}$. The global states of the genes need not be consistent with the gene regulation rules.
2. The global state $\psi$ 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 $\psi$ of $G$ is *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 $\psi$ of $G$ is *native global state* when no perturbations are made ($e = \langle\rangle$).

We shall now prove the 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. The Table 14.1 summarizes the results. Computationally, all algorithms for the results with polynomial experiments in Table 14.1 run in polynomial time.

| 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 $\leq D$<br>All genes are $AND$-nodes ($OR$-nodes) | $\Omega(n^D)$ | $O(n^{D+1})$ |
| In-degree $\leq D$<br>Acyclic | $\Omega(n^D)$ | $O(n^D)$ |
| In-degree $\leq 2$<br>All genes are $AND$-nodes ($OR$-nodes). No inactivation edges. | $\Omega(n^2)$ | $O(n^2)$ |

Table 14.1: Bounds on number of experiments needed for reconstruction (n - number of genes, $D$ - maximum in-degree)

### 14.2.3 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 14.1** At least $2^{n-1}$ 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 without suffice (we get one output bit per experiment). ■

**Proposition 14.2** $n2^{n-1}$ 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 Prop. 14.1. Hence $n2^{n-1}$ experiments suffice in order to identify the whole network. ■

**Theorem 14.3** *An exponential number of experiments are necessary and sufficient for the identification of a gene regulatory network.*

### 14.2.4 Bounded In-degree Case With Bounded Cost

Since an exponential lower bound was proven in a 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 14.4** $\Omega(n^2)$ 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 the 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 a 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 $\{x, y\}$, at least $\Theta(n^2)$ 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.

■

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 14.5** $O(n^4)$ 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 of 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 there exists a boolean function $g(a, b)$ such that $c \equiv g(a, b)$ 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, let us 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 a 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 assignment to $\{a, b\}$ is fixed. Therefore, $c \equiv g(a, b)$ does not hold. In case of $\{p, q\} \cap \{a, b\} \neq \emptyset$, 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 14.6** *$O(n^{2D})$ 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 cost of each experiment is bounded by a constant.*

### 14.2.5 More Efficient Strategies for Special Cases

In this section we consider a 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 the network contains inconsistent nodes.

**Theorem 14.7** *A gene regulatory network which consists of AND and/or OR nodes 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 $c \notin \{a,b,x\}$. Then $g(a,b)$ is assigned to $c$ (i.e., $f_c = g$) if and only if there exists a boolean function $g(a,b)$ such that $c \equiv g(a,b)$ for any assignment to $\{a,b,x\}$. Since the other cases of this property can be proved as in Prop. 14.5, we consider a 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 $c$, and 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 an acyclic case for which we can 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 $e_2$ to $\{x_1, x_2, ..., x_k\}$ 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 $y_i$.

**Definition** A set of nodes $\{x_1, x_2, ..., x_k\}$ has *strong influence* on $y$ if there exist 2 experiments $e_1$ and $e_2$ to $\{x_1, x_2, ..., x_k\}$ 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. But, in what follows, the network can not 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 $x_1$ or $x_2$ or ... $x_k$.

**Theorem 14.8** *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 Prop. 14.4 and Thm. 14.6. We prove the upper bound only for $D = 2$, where those for the 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 for other types of nodes. We assume (w.l.o.g) that all nodes are of in-degree 2 as in Prop. 14.5. 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 14.9** Suppose each set $U$ of nodes does not have influence on any node to which there is no path to $U$. 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 $D = 2$, where those for the other cases it can be proved in 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 $G'(E', V)$. ■

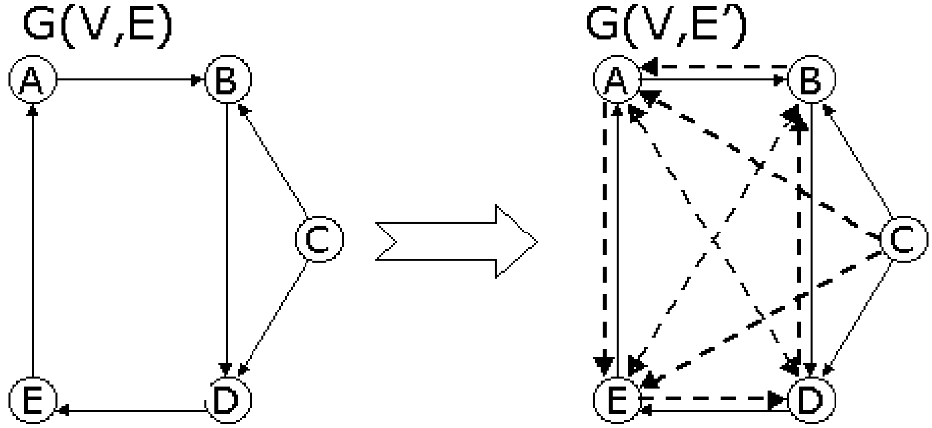



Figure 14.11: *Example of construction E' from E (Prop. 14.9)*
*InfTo(A)={(B,C),(B,D),(B,E),(C,D),(C,E),(D,E),*
*InfTo(B)={(A,C),(A,D),(A,E),(C,D),(C,E),(D,E)},*
*InfTo(C)={},*
*InfTo(D)={(B,C),(B,A),(B,E),(C,A),(C,E),(A,E)},*
*InfTo(E)={(B,C),(B,D),(B,A),(C,D),(C,A),(D,A)}*

### 14.2.6 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 not including *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 $x$ that $InfFrom(x)$ is determined uniquely will be considered.

**Proposition 14.10** 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

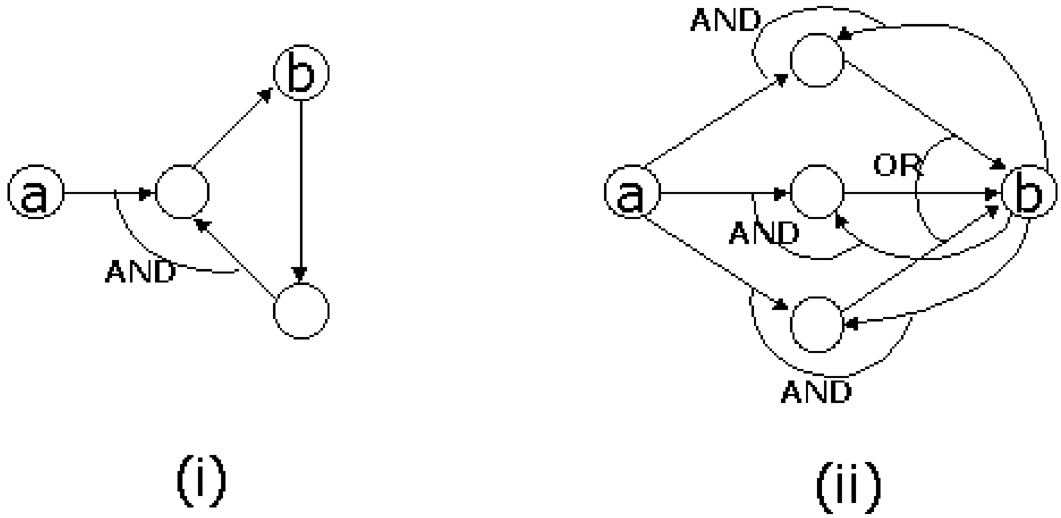



Figure 14.12: These cases satisfy $InfFrom(b) \cup \{b\} = InfFrom(a)$ but do include cycles passing through node $b$.

$x \in InfFrom(a)$. Since there is no cycle including $b$ and there is a path from $x$ to $b$, $x \notin InfFrom(b)$ holds, which is a contradiction.

■

Note that the condition that there is no cycle including $b$ can not be removed from the above proposition. For example, $InfFrom(b) \cup \{b\} = InfFrom(a)$ holds in both networks Fig. 14.12, but in each case, there not necessarily exists an edge $(a, b)$. 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 number of experiments as in Prop. 14.1, it is expected that such cases as in Fig. 14.12 seldom occur. Therefore, if only one node $b$ satisfies $InfFrom(b) \cup \{b\} = InfFrom(a)$, we may conclude 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.

### 14.2.7 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 14.11** *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), that means, there is a global state consistent with all gene regulation rules.

**Theorem 14.12** *Testing the stability of a given gene regulatory network under an experiment is* NP-complete*.*

## 14.3 Interactive Inference and Experimental Design

This section is based on a paper [5] by T. Iddeker, V. Thorsson and R. Karp. Our goal is to infer the underlying genetic networks from a series of steady-state gene expression profile 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. So this model may be able to reconstruct informative networks with few feedback loops.

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.

### 14.3.1 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 level also differs 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 covering* which can be solved by the 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.

### 14.3.2 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:

$$H_p = -\Sigma_{s=1}^{S} \frac{l_s}{L} log_2(\frac{l_s}{L}) \tag{14.1}$$

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 ($H_p = 0$).

### 14.3.3 Evaluation of the Technique

A series of experiments have been performed by the authors of [5] 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 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 evaluation 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

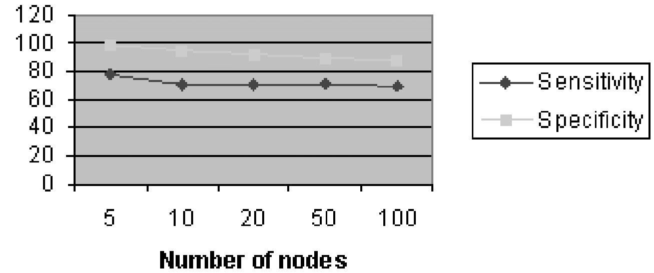



Figure 14.13: Sencitivity and specificity vs. number of nodes

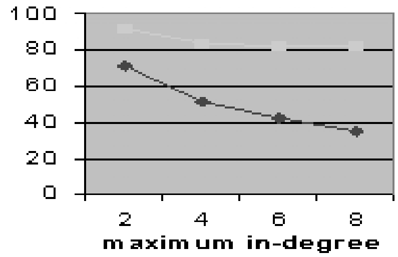



Figure 14.14: Sencitivity and specificity vs. maximum in-degree

grew exponentially with $N$, this number was consistently due to ambiguities at just 10% of the nodes.

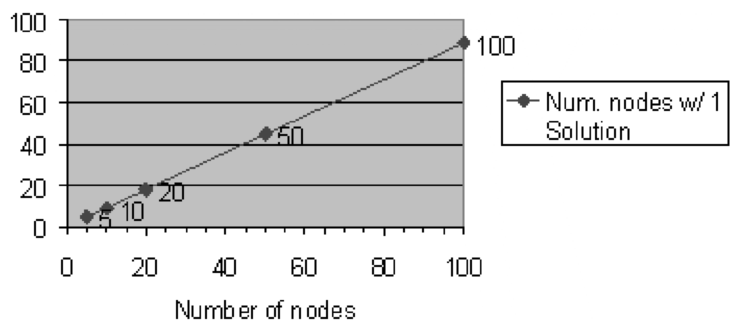



Figure 14.15: Percentage of networks with one solution

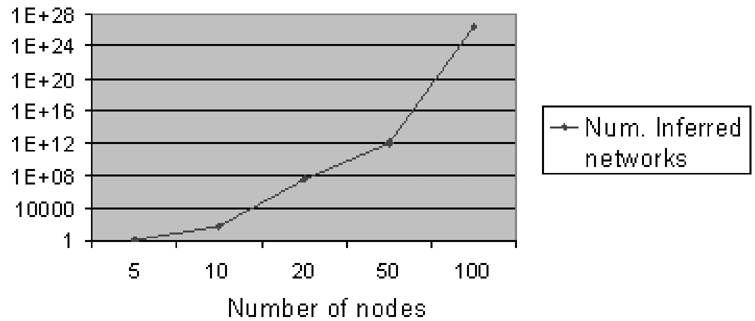



Figure 14.16: Number of inferred networks vs. number of nodes.

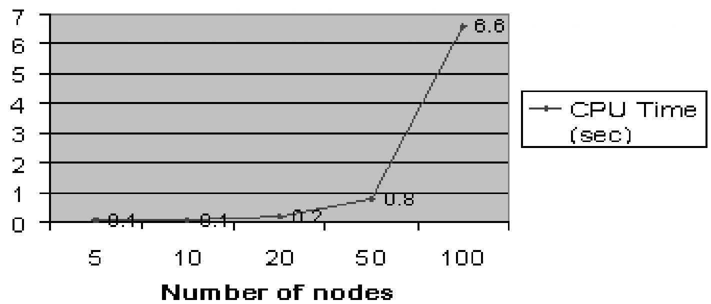



Figure 14.17: CPU time vs. number of nodes.

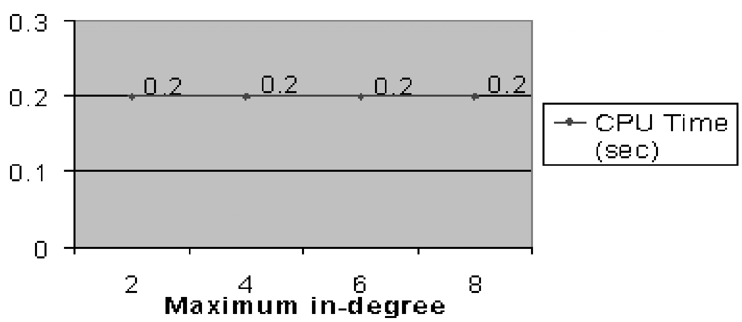



Figure 14.18: CPU time vs. maximum in-degree.

| 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 14.19: Summary of predictor evaluation.

### Chooser Evaluation

In order to evaluate chooser performance 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 14.20. They show a pattern of jumps and decays in the number of network solutions, correlated with an increase of number of inferred edges.

### Lower Bound Comparison

The following theorem, due to J. Hertz, [4], specifies a lower bound on the amount of data needed to specify a network:

**Theorem 14.13** *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 $k$ is, for $N >> k$: $klog_2(\frac{N}{k})$*

It is therefore interesting to characterize the behavior of predictor-chooser strategy in relation to 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

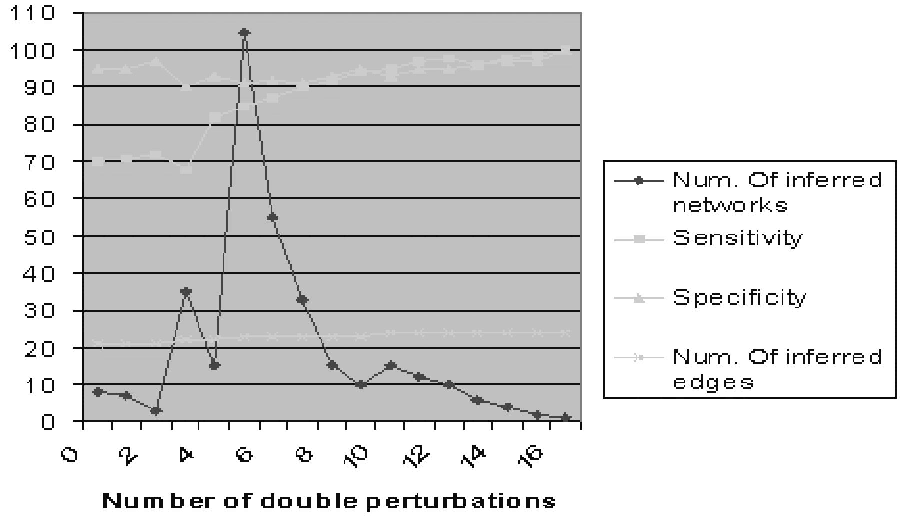



Figure 14.20: Summary of chooser evaluation: progress through experimental design

the network hypotheses. The results, shown in figure 14.21, do indeed indicate logarithmic behavior.

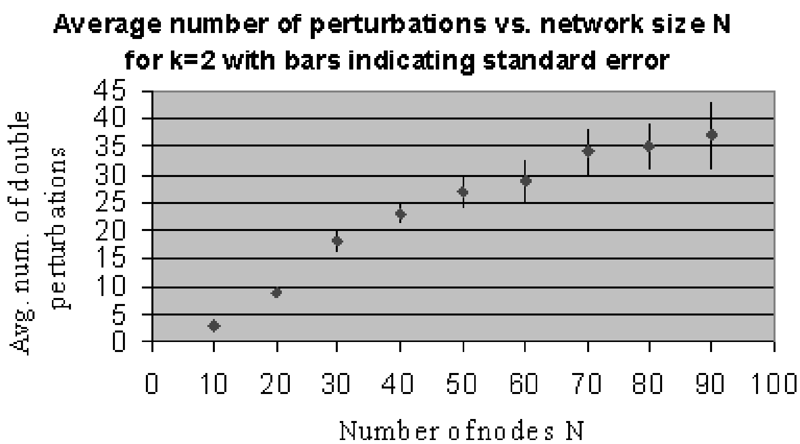



Figure 14.21: Average number of perturbations vs. network size

# Bibliography

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

[2] P. D'haeseleer and S. Fuhrman. Gene network inference using a linear, additive regulation model. unpublished, 2000.

[3] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. In *Pacific Symposium on Biocomputing*, pages 41–52, Hawaii, Hawaii, 1999. World Scientific Publishing Co.

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

[5] 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.

[6] Friedman N., Linial M., Nachman I., and Pe'er D. Using bayesian networks to analyze expression data. *J. Computational Biology*, 7(3):601–620, Nov 1998.

[7] Liang S, Fuhrman S, and Somogyi R. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific Symposium on Biocomputing*, pages 18–29, Maui, Hawaii, 1998. World Scientific Publishing Co.

[8] R. Somogyi and C. Sniegoski. *Complexity*, 1:45–63, 1996.