Analysis of Gene Expression Data **Spring Semester, 2007**

Lecture 11: June 28, 2007

Lecturer: Ron Shamir *Scribe: Yuri Phink and Boris Manilov*[1]

11.1 Metabolic networks

11.1.1 Introduction [7]

The availability of the complete genomes allows many new types of experiments and analysis. The new approaches and questions that the genomes make possible are usually referred to as *functional genomics*. The task is to define the function of a gene (or its protein) in the life processes of the organism, where function refers to the role it plays in a larger context.

But what are these life processes? The most obvious example is the *metabolism of an organism*, the basic chemical system that generates essential components such as amino acids, sugars and lipids, and the energy required to synthesize them and to use them in creating proteins and cellular structures. This system of connected chemical reactions is a *metabolic network*, and this is the one of the oldest biological network of systems that were studied. Today it has target on a new direction in light of the genome research technology rise.

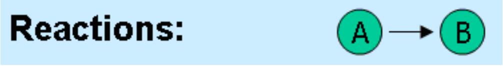
11.1.2 Metabolism

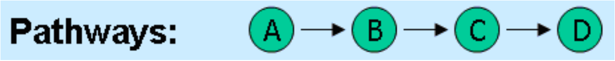
Metabolism is the biochemical modification of chemical compounds in living organisms and cells. The cell metabolism includes all chemical processes in a cell that produce energy and basic materials needed for important life processes. This includes the biosynthesis of complex organic molecules (anabolism) and their breakdown with the release of energy (catabolism). A substance which participates in a biochemical reaction is called a metabolite (cited from [1]).

11.1.3 Metabolic Networks

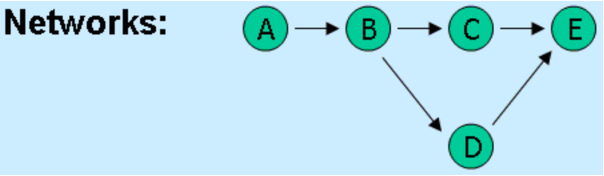
Metabolic networks consist of vertices that represent metabolites and edges that represent *biochemical reactions*. A biochemical reaction is catalyzed by an enzyme, which is a protein and is translated from a corresponding gene. A series of consequent reactions is called a *pathway*. In addition, a metabolic network that is constructed out of reactions is not neces-

[1] Based on scribes by Uri Avni and Liza Potikha, June 02, 2005 and Nir Dil and Liron Levkovitz, January 5, 2006.

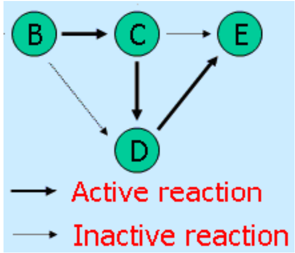




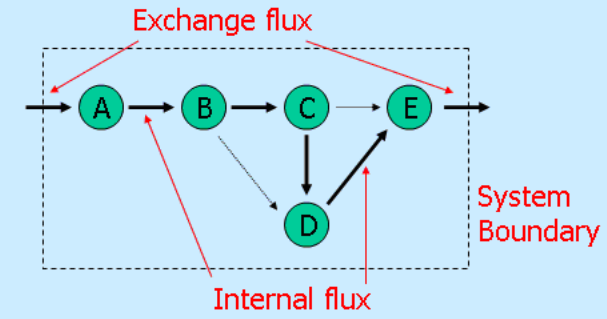
sarily a simple graph, since each enzyme can catalyze more then one reaction. Furthermore, a single reaction can be catalyzed by a complex of enzymes. This can be represented by a bipartite graph. The metabolite that starts the reaction(s) is called a *substrate*, while the resulting one named *product*. All other metabolites are called *intermediates.*



Frequently the metabolic network consists of reactions that are not always active. It is usually mentioned in the graph that displays such a network.



In addition, the intermediates in a metabolite network are not always known. In such a case we are interested in the *flux* of the (sub) network, which is defined as the production or consumption of mass per unit area per unit time. The flux can be measured relative to a system boundary.



The research of metabolic networks/pathways started about a century ago, using mostly chemical reaction analysis. Since then, much of new data became available, revealing structures of some networks. One of the examples is denoted in Figure 11.1.

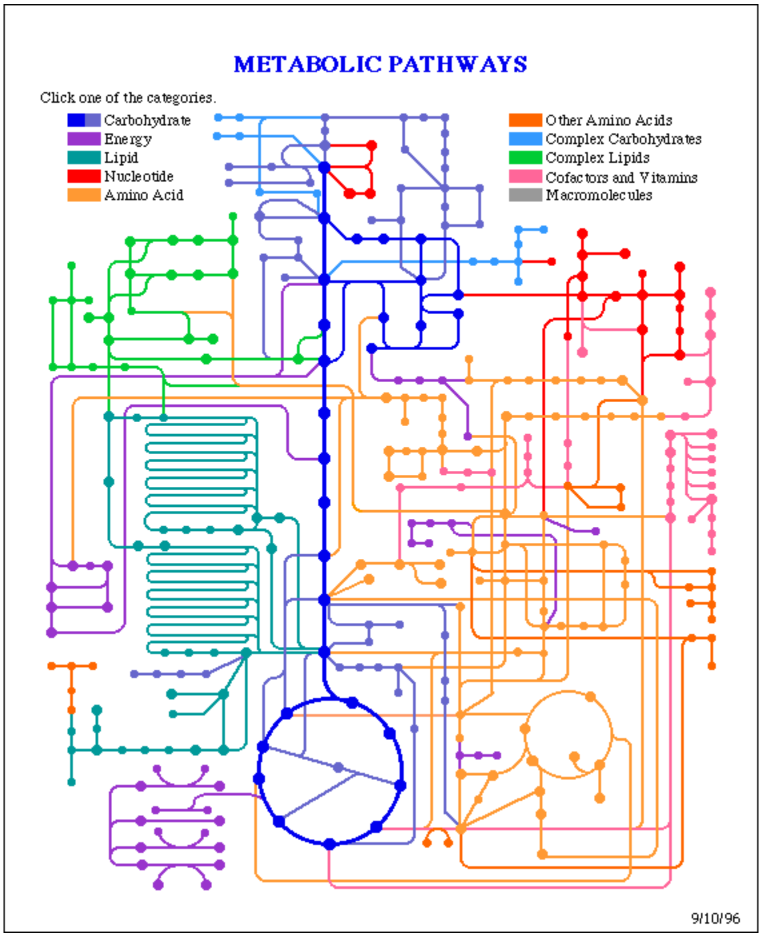


Figure 11.1: This diagram depicts an example of metabolic networks. Each color of the pathway represents a different network. Note that the different networks can cooperate with each other.

11.1.4 Bioinformatics of Metabolic Networks

As was already mentioned, much of the knowledge was gathered in previous years of research on metabolic networks. Currently, the data continues to be collected and stored in metabolic network databases.

There are different ideas of how this data can be helpful. The grand challenge is to use the data in order to reconstruct genome-scale metabolic networks using computational methods. A simpler but non trivial task is to analyze existing networks, which means studying their behavior under different environment/gene perturbations. A slightly different task is simulating of a cell lifecycle. It goes beyond the metabolic networks reconstruction, nevertheless metabolic networks play quite a major role in a lifecycle of a cell. In Figure 11.2 there is an example of how complicated the cell model can be.

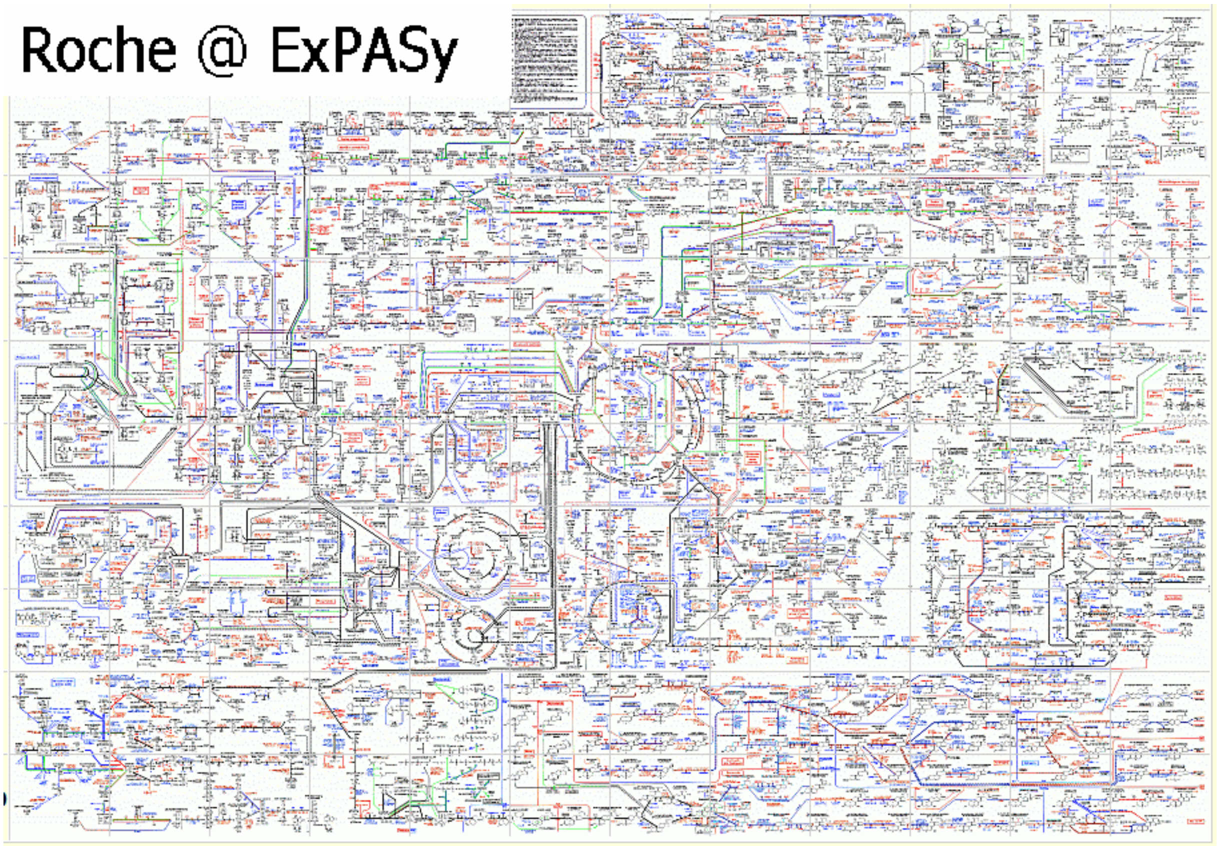


Figure 11.2: A map introducing chemical reactions processing in a cell. Such maps were used widely in the biochemical labs prior to computer databases usage. Nowadays the number of electronics databases of metabolic networks can be found, such as the ExPASy, which contain maps like these.

11.1.5 Genome-scale metabolic network reconstruction

Referring to the grand challenge, the idea is to collect all of the knowledge and information from previous studies (books, papers), genome, pathway databases etc. and build up a metabolic model according to all of these. If the model we build is good enough, it can be

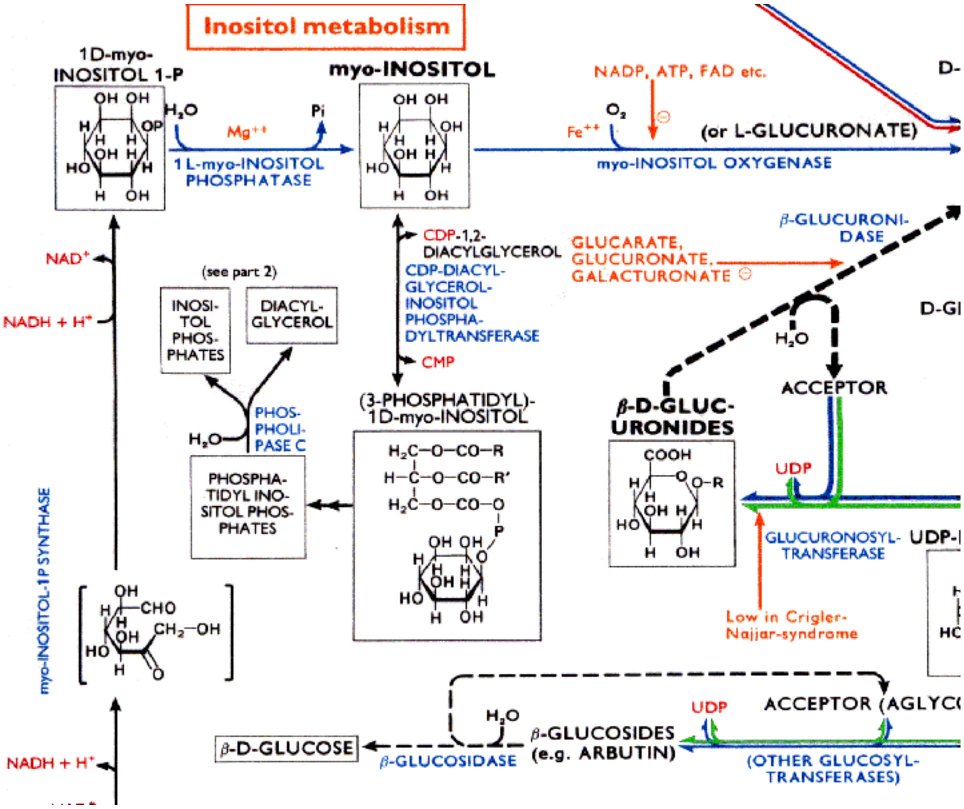


Figure 11.3: Zoom in on Figure 11.2. Observe that the map is detailed enough, describing substrates, products, enzymes etc.

used to predict behavior of that metabolic mechanism subjected to different environmental or genome changes. Otherwise, we must return to the initial point and try to improve the model.

Kinetic Models

Building this kind of model involves a number of steps.

1. *System definition*
   First we choose a number of metabolites and their co-reactions that take a role in the model. The number is commonly limited to tens or at most hundreds. Regulations are generally neglected in primitive models, i.e. only chemical reactions are considered. At this step the system boundary is defined so that flux measurement will be possible. When the latter is chosen, e.g. cell membrane, the environmental considerations need to be taken in to account since the system is not self-sufficient and there may be either diffusion across membranes or active transport systems. The resulting model can resemble the one in Figure 11.4.

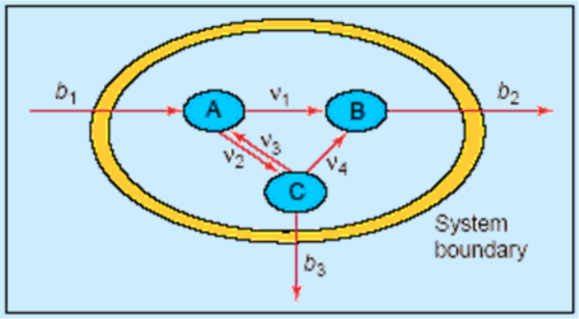


Figure 11.4: A model system comprising three metabolites (A, B and C) with three reactions (internal fluxes, $v_i, i = 1,..,4$ including one reversible reaction) and three exchange fluxes $(b_j, j = 1, 2, 3)$.

2. *Mass balance*
   The edges (reactions) in the graph are signed by numbers, that introduce the amounts of metabolites (molecules) needed in order to cause a reaction. These numbers are provided by *Stoichiometry*, the study of the quantitative relationships between amounts of products and reactants in a chemical reaction.
   Thus, we can look at the resulting weighted graph and compute the delta change rate at each metabolite per unit of time. That is, an example of the delta change rates are given by (refer to Figure 11.4) :

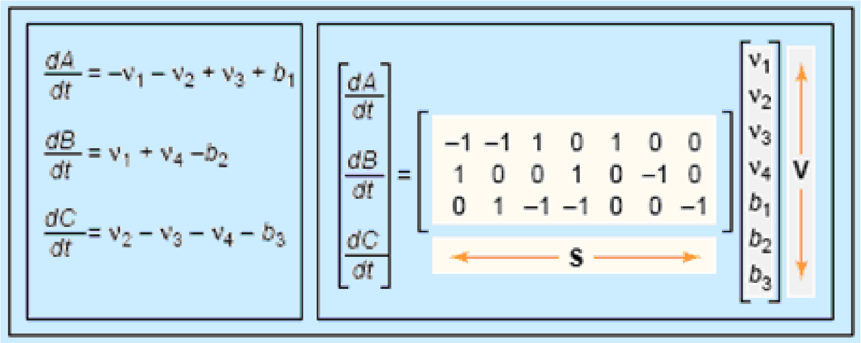
$$\frac{dA}{dt} = v_1 + v_4 - b_2 \quad \frac{dB}{dt} = b_1 + v_3 - v_2 - v_1$$

3. *Evaluation*

   The equations can be presented conveniently as a multiplication of matrix of coefficients (Stoichiometry matrix) with the vector of reactions, e.g. Figure 11.5. In general, for each metabolite $X_i : \frac{d[X_i]}{dt} = \sum_j S_{ij} v_j s$, where S_{ij} is the stoichiometric coefficient of the reactant X_i in the reaction j with the flux v_j. It is negative if X_i is a substrate, and positive if it is a product.

   Finally, these differential equations can be arranged and solved.

   The model introduced here is called a kinetic model. It describes the dynamics of metabolic behavior over time. It incorporates metabolite concentrations, enzyme concentrations and enzyme activity rates (which depend on these concentrations), and is solved using a set of differential equations.



$$\frac{dA}{dt} = -v_1 - v_2 + v_3 + b_1$$

$$\frac{dB}{dt} = v_1 + v_4 - b_2$$

$$\frac{dC}{dt} = v_2 - v_3 - v_4 - b_3$$

$$\begin{bmatrix} \frac{dA}{dt} \\ \frac{dB}{dt} \\ \frac{dC}{dt} \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & -1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & -1 \end{bmatrix}}_{\mathbf{S}} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \mathbf{V}$$

Figure 11.5: Representation of the network kinetics (Figure 11.4) as a mutiplication of stoichiometric matrix by a vector of variables.

Currently such models are not feasible for large scale networks, due to the high complexity level and the requirement of specific enzyme activity rates data, which is difficult to measure. Nevertheless the kinetic models were successfully used to model the activity of specific pathways such as glycolysis and histidine synthesis in yeast (Figure 11.6).

Flux Balance Analysis

The problems mentioned above brought forward another model type that specifies some additional assumptions generally agreed upon by biologists, in a way that simplifies the reconstruction of the metabolic network. The assumption in this case is the *Steady State* assumption, i.e. there is a time invariant flux through each reaction: $S \cdot v = 0$. The biological argument for it comes from the fact that time constants of growth or other processes are much larger than those associated with individual reaction kinetics, therefore steady state can be presumed. This assumption is a type of *balance constraint*. These are associated with

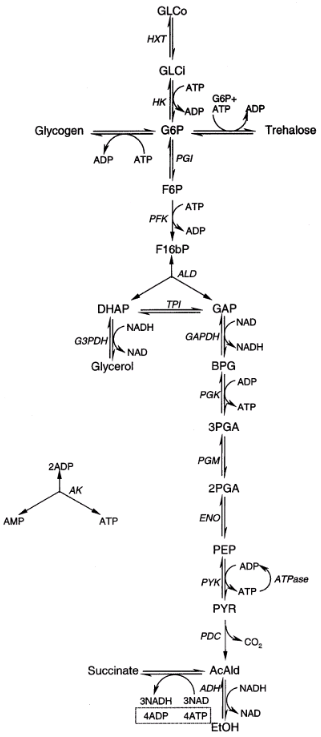


Figure 11.6: Glycolytic pathway in yeast (Pritchard & Kell, 2002).

conserved quantities, such as energy, mass, redox potential and momentum, as well as with phenomena such as solvent capacity, electroneutrality and osmotic pressure.

Hence the null space K of S can be defined by the range of flux distributions under which the system can operate in steady state. The method that uses the steady state assumption in order to solve the problem is called *Flux Balance Analysis*. Figure 11.7 presents an example of the subsystem of a metabolic network and Figure 11.8 represents its equations and the stoichiometric matrix.

A non-unique set of basis vectors spanning the null space may be found. Each vector represents a pathway in the metabolic network by specifying the set of fluxes in the system. However, not every such pathway is biologically feasible. An example is given in Figure 11.9.

Additional constraints are needed for filtering out feasible solutions. Ones that provide it are called *bounds constraints*. They limit numerical ranges of individual variables and parameters such as concentrations, fluxes or kinetic constraints. Allowed ranges of fluxes is an example of bound constraints. Such ranges can be determined in several ways, either experimentally (commonly by isotope labeling), thermodynamically (irreversibility of a reaction) or by capacity constraints (such as maximum uptake rate of a transporter). Math-

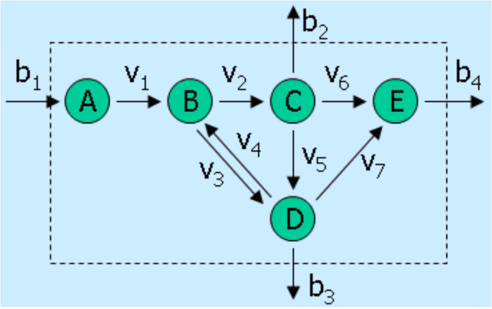


Figure 11.7: A part of metabolic network.

Balance Equations:

A: $-v_1+b_1=0$
B: $v_1+v_4-v_2-v_3=0$
C: $v_2-v_5-v_6-b_2=0$
D: $v_3+v_5-v_4-v_7-b_3=0$
E: $v_6+v_7-b_4=0$

Matrix Notation

$\mathbf{S}\cdot\mathbf{v} = \mathbf{0}$

Stoichiometric Matrix

$$\mathbf{S} = \left[\begin{array}{ccccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \end{array}\right]$$

Internal Fluxes *Exchange Fluxes*

Figure 11.8: The equations and the stoichiometric matrix of the network from Figure 11.7. The internal flux coefficients are the v_is and the exchange flux coefficients are the b_js.

$$\mathcal{B} = \{\mathbf{b}_1,\ldots,\mathbf{b}_6\} = \begin{array}{c} v_1 \\ \\ \\ \\ \\ \\ v_7 \\ b_1 \\ \\ \\ b_4 \end{array} \left[\begin{array}{cccccc} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

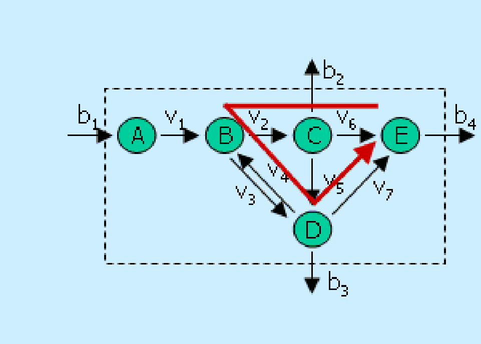


Figure 11.9: Of a possible set of basis vectors each one identifies a flow in the network, however not every one identifies a feasible biological solution.

ematically speaking we define a set of the form: $v_{min} < v_i < v_{max}$. For irreversible reactions, $v_{min} = 0$. Specific upper limits v_{max} that are based on enzyme capacity measurements are generally imposed on reactions. Taken together, the balances and bounds described as linear equations define a biochemically feasible flux distribution space which is a polytope in a high-dimensional space. All allowable network states are contained in this space (Figure 11.10).

Thus, any vector can be represented as a non-negative combination of the generating vectors of the cone, f_k: $F = \{v \in \mathbf{R}^n | v = \sum_k \alpha_k f_k,\ \alpha_k \geq 0\}$.

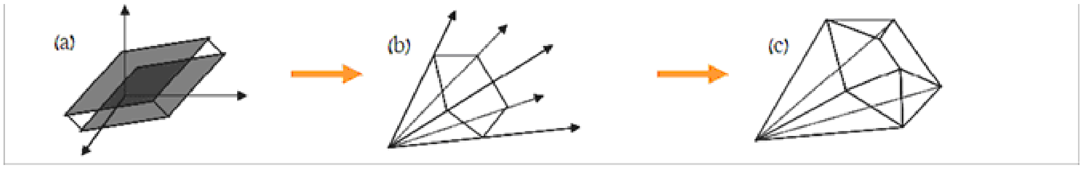


Figure 11.10: Solution space representation. (a) The mass balance constraint $S \cdot v = 0$ limits the solution space to a subspace of $\mathbf{R}^n$. (b) Thermodynamic constraints $v_i > 0$ further limit the solution space to a convex cone. (c) Capacity constrains $v_i < v_{max}$ again further limit the solution space to a bounded convex cone.

Sometimes there are additional constraints that are applied, e.g. we know dependencies among components of some of the flux vectors in the system. Occasionally we also want to allow possible reversible reactions: $v < 0$.

When all the constraints are added to the system an appropriate solution space may be found. Our goal is to find the suitable solution(s) from the given solution space. One of the approaches is to define an objective function and optimize it (find either minimum or maximum) in the given solution space. For instance, one possible goal is to maximize the biomass production. Considering this, we have an optimization problem with linear constraints. Hopefully, the target function is a linear, which results in linear problem with linear constraints and this can be solved through Linear Programming techniques. Observe that an optimal solution can be either unique or there may be multiple solutions (see Figure 11.11).

The choice of an objective function is usually obtained by our understanding (or intuition) of this specific biological mechanism, but we should be careful not to choose too complex mathematical function which will make the task of finding optimal solutions too difficult.

As soon as convex solution space is found we can then analyze the consequences of removing genes (i.e. remove reactions). The different approaches of usefulness of this solution space are mentioned in the Figure 11.12. They all contribute to a better understanding and more precise modeling of metabolic networks.

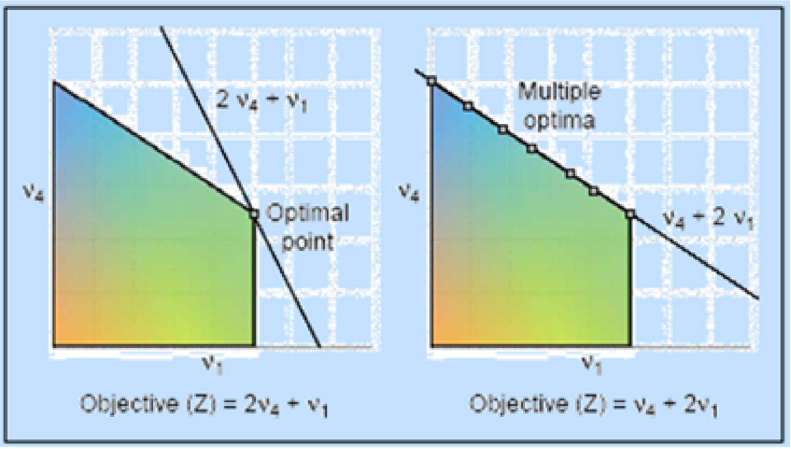


Figure 11.11: Optimization of the system with different objective functions (Z). Case I gives a single optimal point, whereas case II gives multiple optimal points lying along the edge.

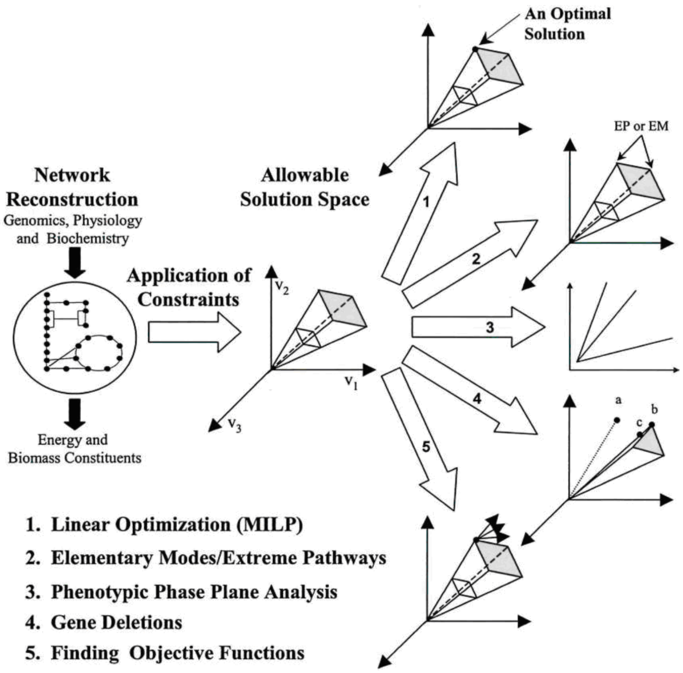

An Optimal Solution
EP or EM
Network Reconstruction
Genomics, Physiology and Biochemistry
Allowable Solution Space
Application of Constraints
v2
v1
v3
Energy and Biomass Constituents
1
2
3
4
5
a
b
c
1. Linear Optimization (MILP)
2. Elementary Modes/Extreme Pathways
3. Phenotypic Phase Plane Analysis
4. Gene Deletions
5. Finding Objective Functions

Figure 11.12: Different approaches to the solution space analysis.

11.2 Interactive Inference and Experimental Design

This section is based on a paper by Ideker, Thorsson and Karp [5]. Our goal is to infer the underlying genetic network from a series of steady-state gene expression profiles, using as few perturbation profiles as possible (where one, two or three genes can be perturbed in one experiment). 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.

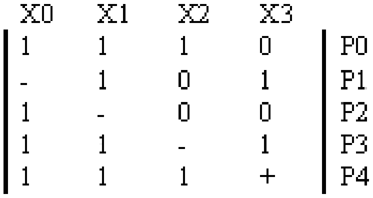
Next, a method called the predictor is used to infer one or more hypothetical Boolean networks consistent with these profiles. When several networks are inferred, the predictor returns only the most parsimonious, as measured by those networks having the fewest number of interactions. Depending on the 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 (Figure 11.13). Every line represents an experiment conditions. First line introduces a steady-state, natural state of a system. The second line represent the experiment where X_0 was knock-out, in the third X_1 was knock-out,... , in the last experiment X_3 had over-expression.

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



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 11.13: The initial expression matrix.

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 (see Figure 11.14).

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 networks generated by the predictor. Its goal is to choose a new perturbation p, from a set of allowed perturbations P,

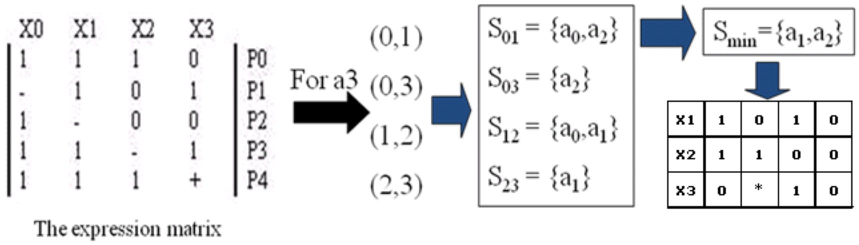


Figure 11.14: The Predictor procedure example.

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:

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

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 [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 to high or low) and all single perturbations. In addition, a number of non-random networks, modeled 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 (figures 11.15 and 11.16). The number of nodes whose functions had only a single minimal solution was approximately 90% for $k = 2$, independent of N.

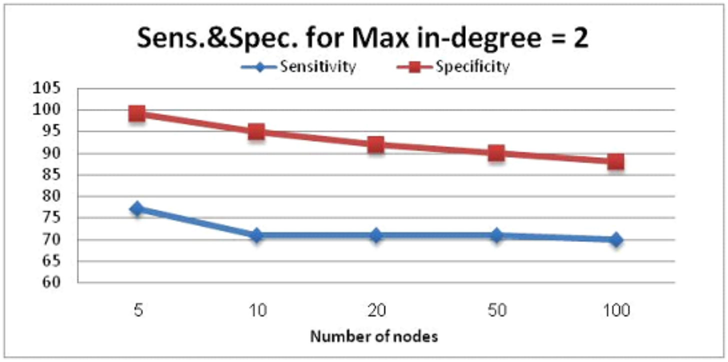


Figure 11.15: Sensitivity and specificity in percents vs. number of nodes.

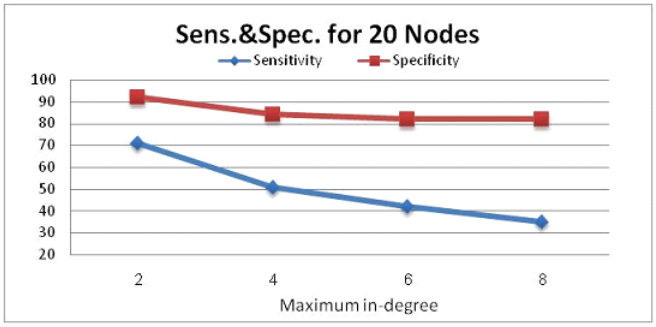


Figure 11.16: Sensitivity and specificity vs. maximum in-degree.

Thus, although the number of inferred networks grew exponentially with N, this number was subjected to ambiguities at just 10% of the nodes (figures 11.17 and 11.18).

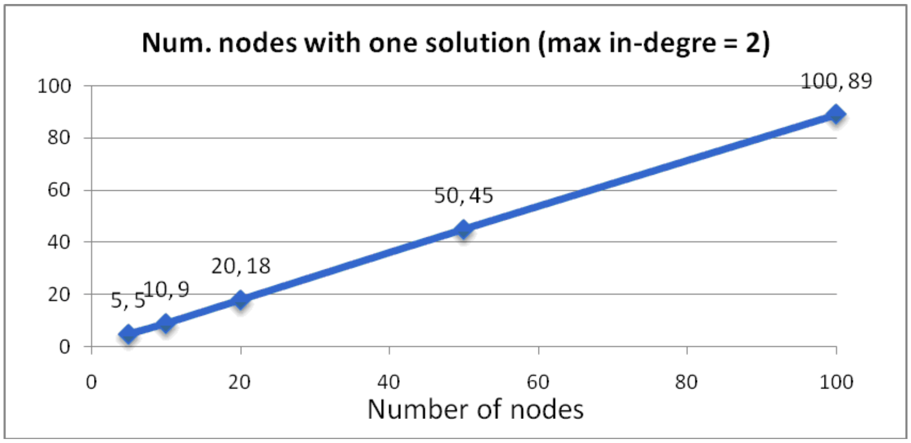


Figure 11.17: The average number of nodes whose functions have only a single minimal solution.

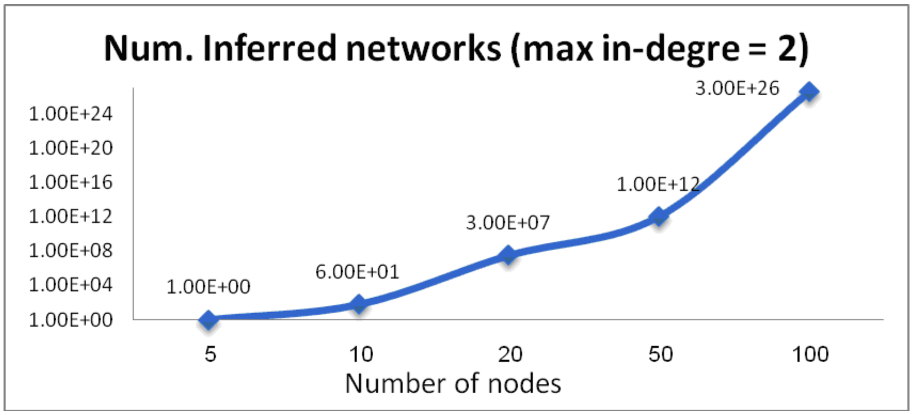


Figure 11.18: The number of inferred networks vs. number of nodes.

Non-Random Simulation

A number of non-random target network topologies modeled after known biological networks were also simulated and inferred. The well-studied networks responsible for bacterial chemotaxis, galactose induction in yeast, and yeast meiosis were encoded using our network representation, and expression levels from the wild-type state and all single perturbations were simulated as before. In all three cases several parsimonious networks were inferred, some of which resembled or corresponded exactly to the original simulated network.

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.

Figure 11.21 tracks changes in the number of inferred networks, the number of edges in each inferred network, average sensitivity, and average specificity as new double perturbations were added over each iteration of the design process. This characteristic pattern of jumps and decays in the number of network solutions, correlated with a monotonic increase in the number of inferred edges, was observed consistently over many other simulations using a wide range of different target networks.

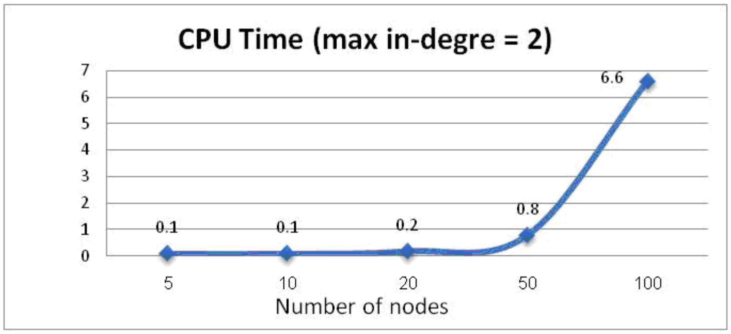


Figure 11.19: CPU time (sec) vs. number of nodes (500MHz PIII).

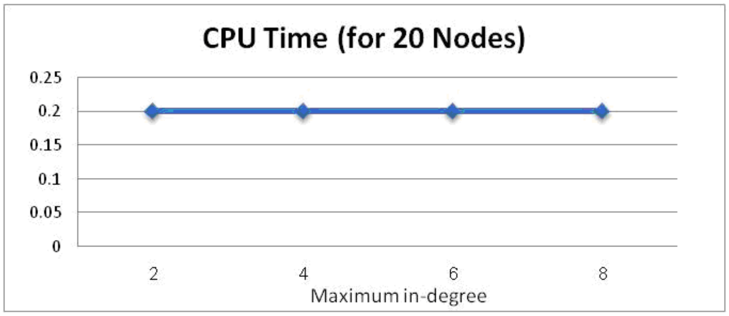


Figure 11.20: CPU time (sec) vs. maximum in-degree (500MHz PIII).

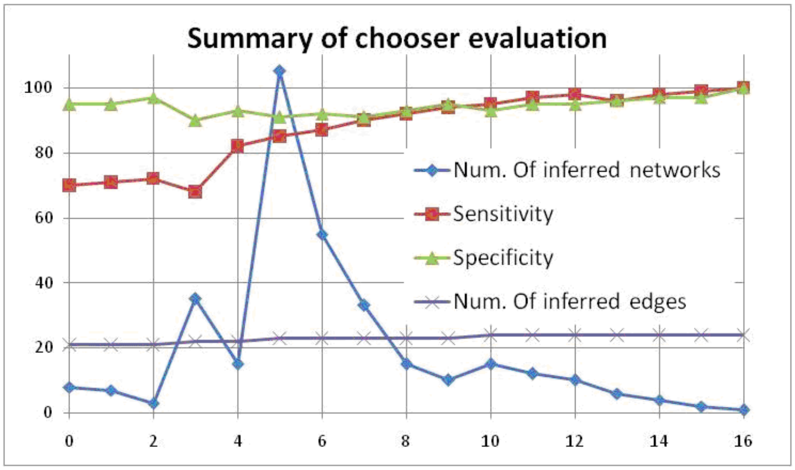


Figure 11.21: Progress through experimental design.

Effect of Network size

A theoretical lower bound on the number of gene expression profiles which must be observed to uniquely specify a genetic network has been reported to be $k \log_2(N/k)$ for $N >> k$ [3]. In order to characterize the performance of this methods in relation to this lower bound, the 50 target networks T for each of several values of N with $k = 2$ was generated. The wild-type state and all single perturbations were simulated on each T, and as before the chooser was used iteratively in conjunction with the predictor to select a series of double perturbation experiments to refine the network hypotheses until the iteration terminated. The average number of double perturbation experiments required for each N is shown in Figure 11.22. These preliminary results show evidence of logarithmic behavior.

Future work

There are number of extensions that may be done for the described Chooser/Predictor method: First, in nearly all cases of practical interest some knowledge of network genes and interactions is available. In this regard, pre-existing information about the network may be incorporated during the inference process.

Second, the observed levels of gene products and other macromolecules may be such that a two-level description misses important features of the network. In these cases, a multi-level description (greater than two) may be adequate to describe the data. It may also be possible to extend the method for use with continuous (rather than discrete Boolean or multi-level) gene expression data.

Third, as was already mentioned, only genetic networks which do not contain cycles were considered. This restriction may be sufficient to describe certain biochemical networks, but

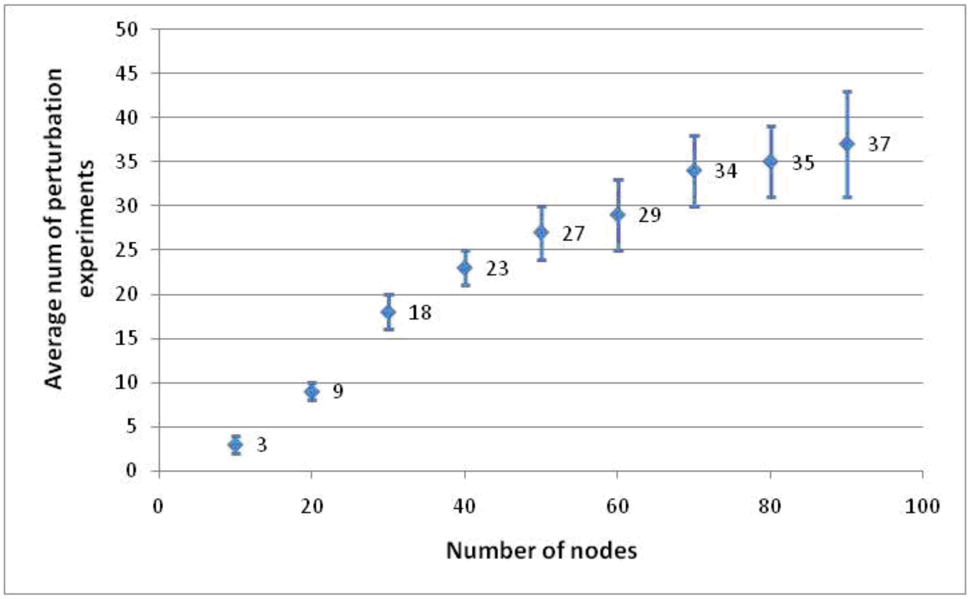


Figure 11.22: Average number of perturbation experiments vs. network size for in-degree = 2, with bars indicating standard error.

biological examples of cyclic gene networks are also known. Therefore, another future direction is to allow cyclic solutions in the inference procedure.

Fourth, we currently do not allow for noise or other imperfections in the gene expression data sets used for network inference. Gene expression levels measured with DNA microarrays or other technologies are subject to an appreciable amount of experimental variability, and the impact of this variability on our method should be evaluated. The inference method could be modified to account for noisy data.

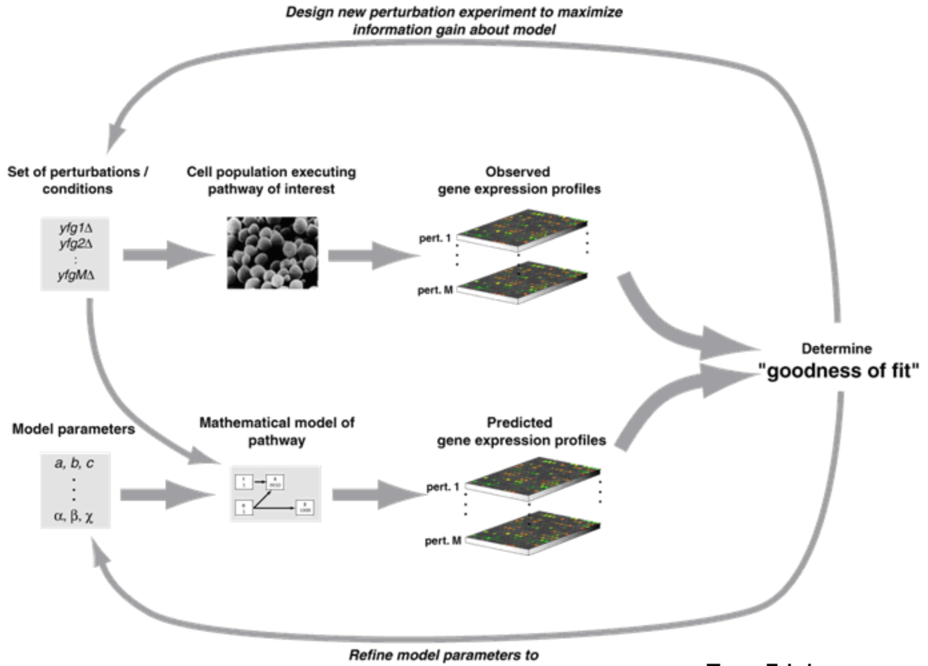
Finally, proposed methods may be used in conjunction with existing software for grouping genes. For instance, a clustering algorithm might be used to reduce the apparent size of the network by grouping genes according to similar expression level over the series of perturbations performed, and then one representative from each cluster could be supplied to the network inference method.

11.3 Modeling and Expansion of Regulatory Pathways

11.3.1 Introduction

There is a common design template of gene expression experiments whose goal is a gene network reconstruction.

We start from the designing of experiment which includes a set of perturbations or different environmental conditions, and perform it by growing cells under these conditions. The results of the experiment are gene profile expressions, which are compared to the expected expressions, provided by an existing presumed model of this network. The estimated 'goodness' of the model is the computed. According to the quality achieved, the directions to improve model can be ascertained and the input for a new experimental design is realized. These stages are depicted on Figure 11.23.


Design new perturbation experiment to maximize information gain about model
Set of perturbations / conditions
yfg1Δ
yfg2Δ
yfgMΔ
Cell population executing pathway of interest
Observed gene expression profiles
pert. 1
pert. M
Determine "goodness of fit"
Model parameters
a, b, c
α, β, χ
Mathematical model of pathway
Predicted gene expression profiles
pert. 1
pert. M
Refine model parameters to improve fit

Trey Ideker

Figure 11.23: Design template for gene network modeling experiments.

As can be observed, standard approaches for gene expression analysis are gene clustering, biclustring or gene network reconstruction. Much of the data is gathered using these techniques, which can be helpful in the analysis of the presumed models. The prevailing paradigm for data analysis is shown in the Figure 11.24.

An ultimate goal is to take into consideration all the information that is known about the biology of the system in order to construct an improved model (see Figure 11.25). However, biological data is generally qualitative and not quantitative, i.e. the great part of it is located in papers and not in databases, therefore quantifying that data becomes a rather complicated issue.

Despite the difficulties mentioned, there is good reason to choose this approach. For example suppose we know about the existence of two TF-s, which are bound to a promoter of the specific gene (Figure 11.26). The result is mRNA which is translated into a protein of

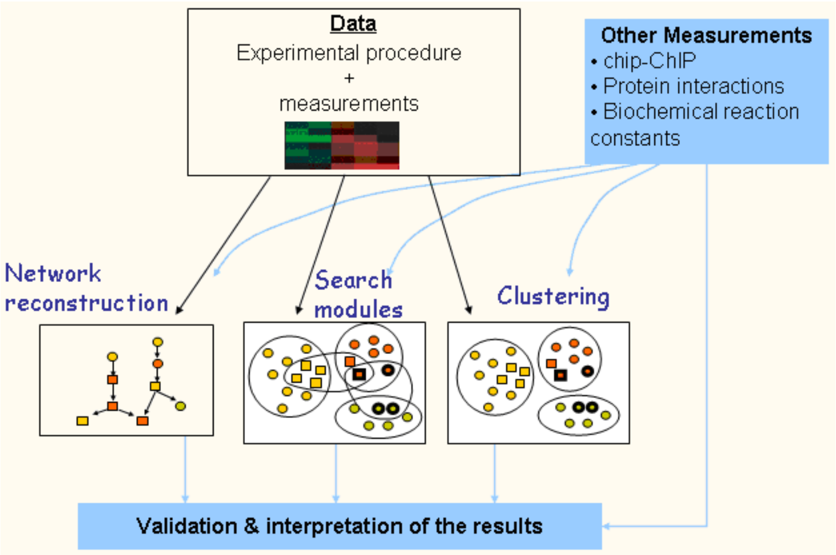


Figure 11.24: The prevailing data analysis paradigm.

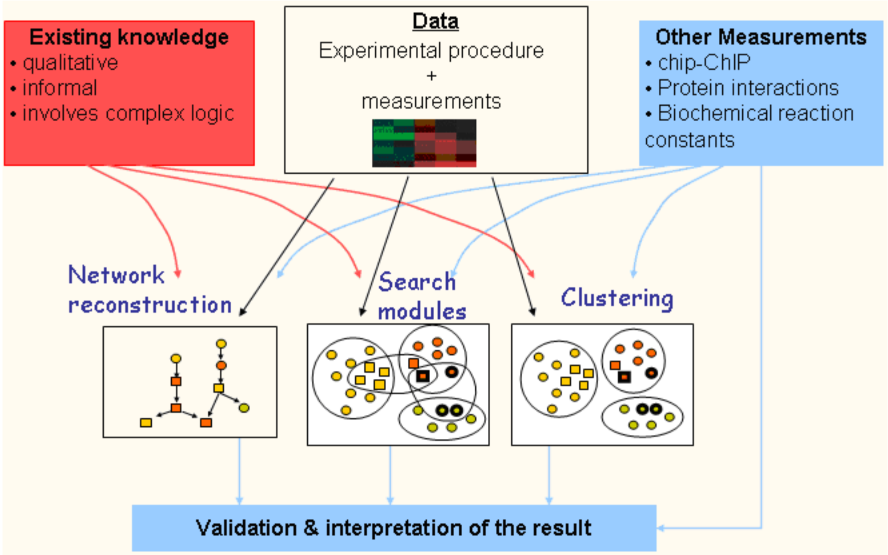


Figure 11.25: An alternative paradigm for gene expression data analysis.

the catalyzing enzyme type, which in its turn participates in a metabolic pathway. Perhaps the loop is closed by some feedback to the TF. Note that only part of whole information about the system is available.

On the other hand we have experimental data of gene expressions of the system given above, and the goal is to improve the model of this system. The improvement comes about by the integration of these two types of data.

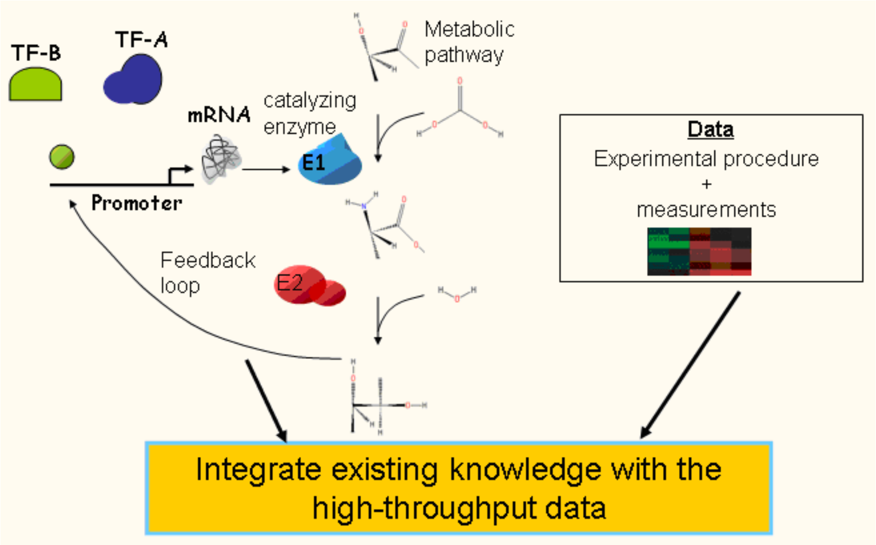


Figure 11.26: The scheme of the integration of experimental data with the existing knowledge.

Once the system we are interested in is defined and the suitable information (e.g. from papers) is found, the translation of this knowledge to the mathematical model is needed. Suppose this task is completed. Then what do we have? What kind of questions would we like to find answers?

In simplistic models we assume that the expression of mRNA exactly identifies the expression of the specific protein. With a more complex approach we can answer the question of whether this is true or not. This reduced to a prediction of the protein variable in the given model when its value is unknown but the value of the mRNA is known. Following the discovery of the expression level of the proteins, we would like to predict the activity level of the TF. This and the former example of prediction variables are called *single variable inference* because the goal is to predict the value of a single variable for a given model.

The model improvement is a difficult challenge, though. It is performed in one of two ways: either the function of a variable in the model is changed, or the model is expanded. The first case implies functional or topological changes in the model, but its framework remains the same. In the second case we change the framework, i.e. new elements are added to the model. The model improvement scheme is depicted below in Figure 11.27.

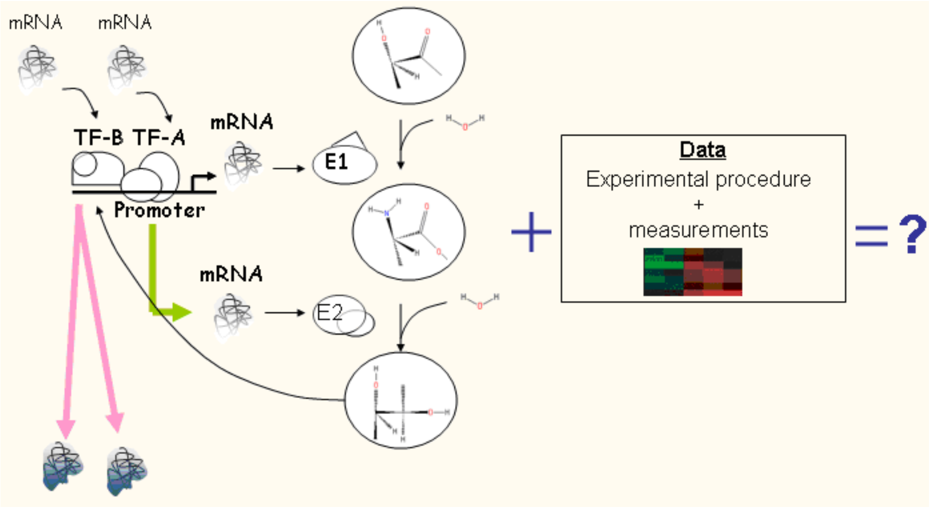


Figure 11.27: Two ways of model improvement. Either model refinement (green arrow) - only the logic of the model is changed (a new connection is added), or expansion (pink arrow) - two new elements are added.

11.3.2 Methodology

Overview

The first step is to build a model according to the available knowledge and it requires a thorough reading of the appropriate literature. Then the relevant data needs to be obtained. These are combined in the final modeling (Figure 11.28).

One of the challenges that exists when creating the initial model is to translate the biological knowledge taken out of papers into a mathematical model, as mentioned above. Throughout this task one is expected to convert biological event into logical functions. However, the logical functions that are available from the literature are usually a significant simplification of reality. Moreover, the relationship between elements is not always stated clearly. These problems complicate the whole procedure.

The Model

Initially a network of variables with discrete states is constructed. Each variable may be a regulator of one or more other variables. The combination of regulator states determines the regulated entity's state via regulation logic. The logic and topology constructs based on prior qualitative knowledge can be expressed as discrete deterministic functions.

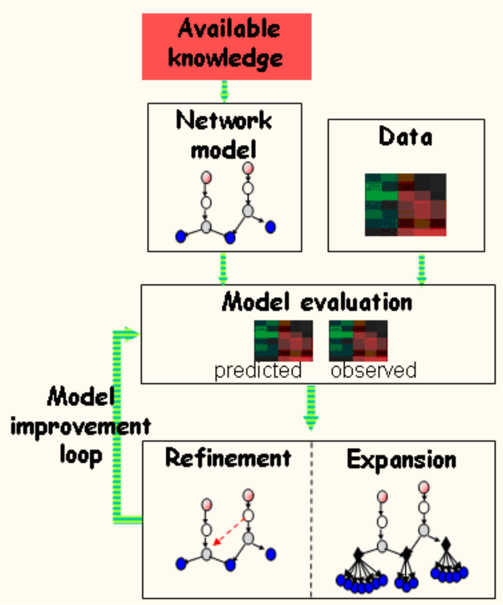
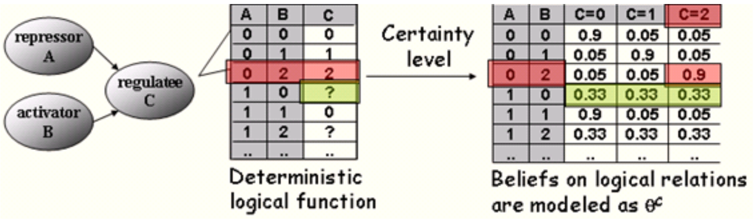


Figure 11.28: Overview of the methodology to gene network modeling.

Usually the logical function is not entirely known and the reliability of the information about the function that is known is limited, therefore a probabilistic model that estimates the variable's values as a function of its regulators is used. An example of such a model that was proposed in [4] will be used throughout the lecture is displayed in Figure 11.29.



A	B	C
0	0	0
0	1	1
0	2	2
1	0	?
1	1	0
1	2	?
..	..	..

Deterministic logical function

A	B	C=0	C=1	C=2
0	0	0.9	0.05	0.05
0	1	0.05	0.9	0.05
0	2	0.05	0.05	0.9
1	0	0.33	0.33	0.33
1	1	0.9	0.05	0.05
1	2	0.33	0.33	0.33
..	..	..	..	..

Beliefs on logical relations are modeled as θ^c

Figure 11.29: The topology and part of the deterministic logic is obtained from prior knowledge. The logical structure is transferred to the probabilistic model when all missing data are supplied with appropriate probability values.

A priori regulation logic is described as $\theta^i(X_i, Pa_i) = Pr(X_i|Pa_i)$, where X_i is a possible value of X (the regulatee - C in Figure 11.29) and Pa_i is a possible regulator values combination (parent's values - A,B).
Note that the resulting function still has deterministic values, but these are not exactly known therefore probabilities for each possible value exist. Also, observe that values range

from 0 to 2 (3 possibilities). This rule was used because more possibilities (4 or 5) do not improve the model significantly.

The probabilistic model we will use is: $Pr_m(X) = \frac{1}{z}\prod_i \theta^i(X_i, Pa_i)$, as proposed by the Bayesian Network chain rule (see lecture 10), with the addition of some normalizing factor z that will not be discussed here in detail. In contrast to the Bayesian model, we will allow loops in the resulting network, so that z needs to be an enhanced normalizer since the Bayesian probability analysis is disrupted in cases of cyclic graphs. This extension of the Bayesian graph model is called a *Factor graph model* [4].

The subsequent step is to gather new useful data by performing appropriate experiments. Gene knockout is one of the experimental options. The outcome is a network that has endured topological transformation in the form of the removal of the input edges of the element that represents the chosen gene, and its expression is forced to have a zero value (see Figure 11.30).

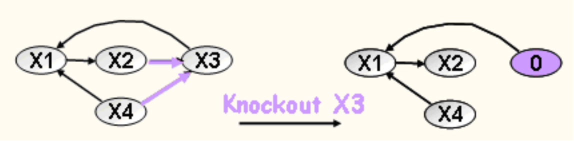


Figure 11.30: The modified network as a consequence of knockout to X_3 gene.

A Microarray analysis gives us noisy estimations of the mRNA levels. This forces us to use some additional probabilistic methods to handle experimental data in a correct way. Discretization of the observed values must be made, and one way to do it is to use Gaussian distributions (assuming their distribution is Gaussian), one per possible value of a variable, because we assume that each discrete value corresponds to a different distribution (see Figure 11.31).

Continuing with the example from Figure 11.29, consider that we measure X_1 and X_4 variables as a result of the knockout of the X_3 gene. The results we get emphasize the probabilities of how close the observed value is to each of the possible discrete states (Fig. 11.32).

This way the likelihood score for comparison between model and the data is assembled, resulting in the following:

$$l_M = Pr(X, Y|M) = \prod_i \theta^i(X_i, Pa_i) \cdot \psi^i(Y_i, x_i)$$

As mentioned in the model overview, the result of this comparison is insight in to model improvement. Let's take for example a possible model refinement, e.g. an addition of some arc (see Figure 11.33). The acceptance of the change depends on a significant likelihood score improvement relative to the original model. The threshold of a decision is estimated

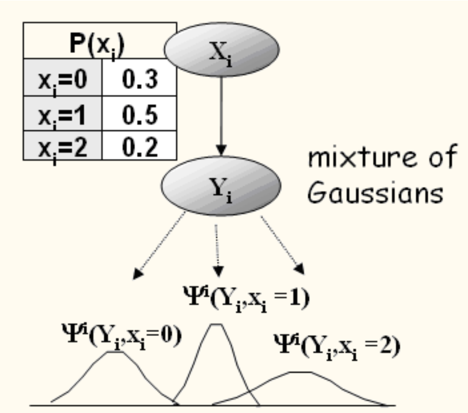


Figure 11.31: A discretization function for the observed gene expression values: $\psi^i(Y_i, x_i)$ denotes the joint distribution of the received data (Y_i) and the logical state (x_i). $Pr(x_i)$ is estimated using an expectation-maximization (EM) algorithm.

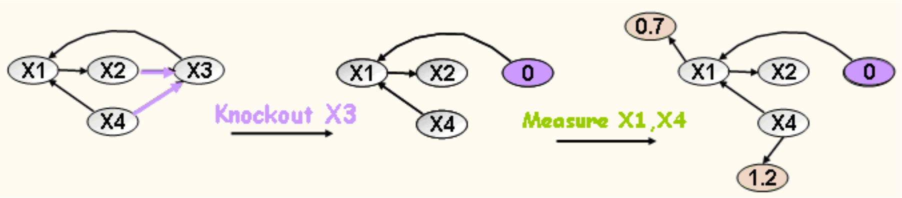


Figure 11.32: The experiment and the measurement process: The observed value of X_1, for example, indicates that there is an 80% probability that the value is 1, 12% that the value is 0 and 8% that the value is 2.

using a non-parametric paired Wilcoxon test on the distribution of likelihood scores. Clearly, the model refinement or expansion is based, first of all, upon a biological intuition/understanding, while of the likelihood function indicates how close this understanding is to reality.

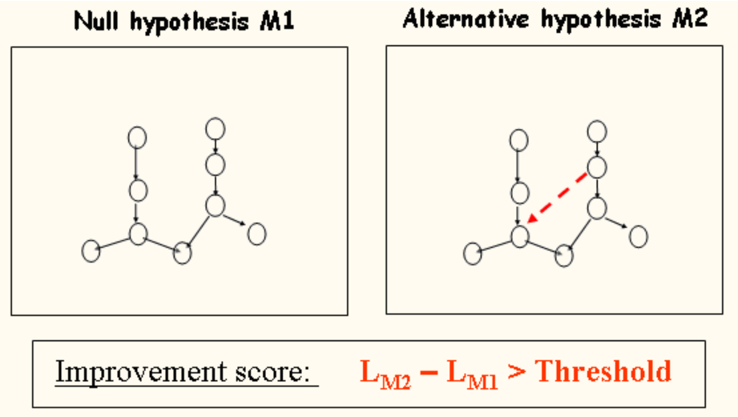


Figure 11.33: Example of model refinement. An alternate model score is computed and accepted only if it is significant enough (the difference of the scores is greater than some threshold).

11.3.3 Results

The osmotic stress response to the yeast was chosen as the target of this model [2]. This organism has been researched for many years and much relevant data are available. The model that was constructed from the literature identifies how the yeast responds to calcium and hyper-osmotic stresses. For each pathway, the model includes the environmental stresses, the signaling cascades, the transcription factors, and their known targets (Figure 11.34).

At first, model refinement was made. As a result three new connections were revealed. They were not included in the original model because the literature did not contain significant evidence for this correlated behavior. However some evidence was still mentioned in the literature, suggesting that the final improved model was actually closer to reality with a high probability (see Figure 11.35).

Model expansion is a more difficult task. As described above, there are lots of genes outside of the model, therefore choosing the right one is not trivial. Nevertheless, much data is available on this gene network, and many perturbation experiments of this network were performed, which provided a division of this network into 12 sub-paths (see Figure 11.36).

Each pathway defines a different logic module. The question is then whether or not there are genes outside of the model (there are 5700 genes outside) that are also regulated by the

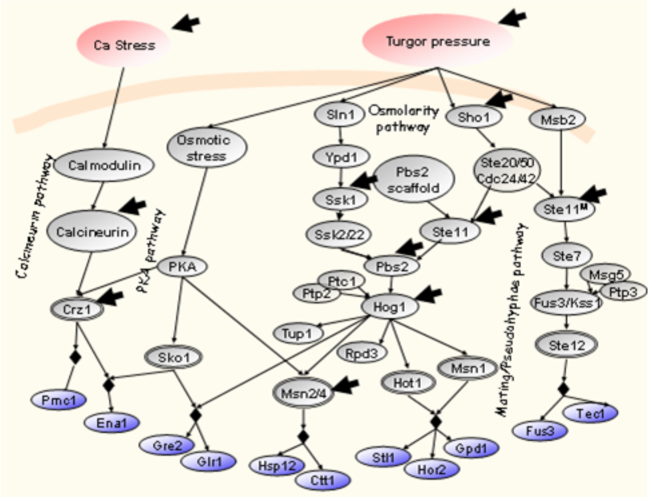


Figure 11.34: A model of the yeast response to osmotic and calcium stress. Arrows in the figure point to knockouts that were performed on the genes of the system during experiments (single or double knockouts were done at each experiment). Double ovals indicates known transcription factors.

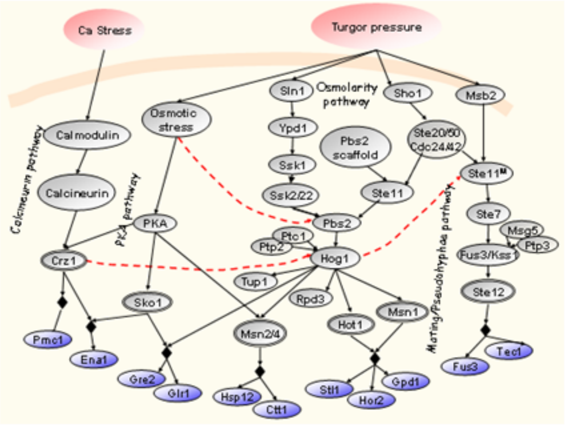


Figure 11.35: The refined model. Three new connections (dotted lines) were added to the original model.

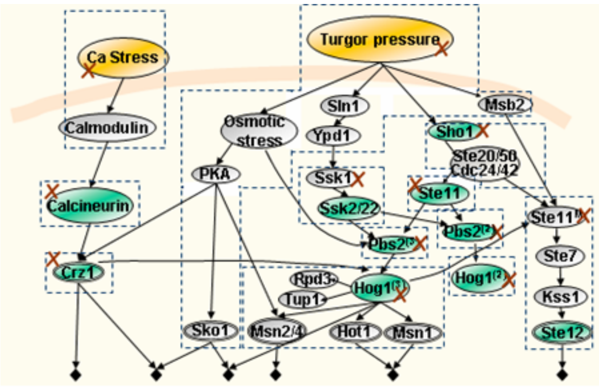


Figure 11.36: Subdivision of the network into 12 pathways. The x-es define the elements that were perturbed in experiments.

same subgroups. Checking all the possibilities of genes vs. all the possible regulators outside of the group is impossible (¿ million possible expansions). To overcome this problem, the search is not for a single gene, but for a group of genes that are regulated by one of the given subgroups. Such a group is called a *Regulatory module* - a set of target genes regulated by the same regulators via the same logic.

Indeed, seeking for such regulatory modules turned out to be successful, as groups with relatively large sizes were found (more on the algorithm can be found at [2]). The statistical significance of these results had to be proved, therefore the comparison with randomized model results was carried out. The randomized model was build by permuting the model logic and seeking for the regulatory modules again. The comparison is introduced in Figure 11.37.

The decision was to choose module sizes of 20 in order to make statistically significant choices. Then the results of the analysis can be seen in Figure 11.38.

11.3.4 Summary

(Cited from Gat-Viks I. and Shamir R. [2]).A large amount of curated qualitative knowledge on biological systems is available today. The formulation of such knowledge is shown here to be surprisingly instrumental in improving biological understanding. This computational framework enables modeling of the existing knowledge in the presence of feedback loops in the network, formalization of the uncertainty in this knowledge, and integration of high throughput data. In addition, the model can accommodate partial noisy measurements of diverse biological entities ([4]). We make major modeling simplifications: The regulatory relations are discrete logical functions, and the model describes the steady state of the system. As expected, the prediction and improvement processes that we propose here also

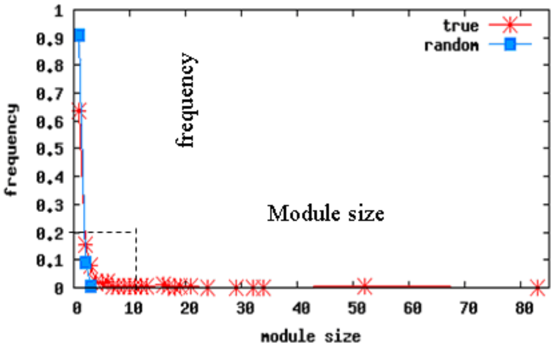


Figure 11.37: Comparison of the true and the random models in seeking regulatory modules. Note that the random model's maximal module size is 3, in contrast to module sizes in the true model that goes beyond 30.

have limitations: They are sensitive to the size and complexity of the model (e.g., number of variables, interactions, and feedback loops), the certainty in the logics, and the amount of data available. The robustness of the described methods to these parameters still needs further exploration. Strong positive indication for the robustness of the prediction process and logical refinement procedure on small networks were achieved ([4]). The robustness of the expansion procedure is yet needed to be systematically explored, although the biological validations in this study are highly promising. In the future, hopefully this study will lead to creation of more sophisticated mathematical models and robust improvement algorithms for the analysis of genome-wide datasets. A key advantage of the module identification approach overviewed here is that a discriminative scoring scheme is used which specifically identifies modules along with their model regulators. Consequently, modules on a finer level than was previously possible can be detected (for example, novel HOG pathway-dependent repressed modules). This method outperforms extant methods mainly because it exploits prior knowledge on the signaling pathways and on the experimental procedure. This prior knowledge helps to detect minute expression differences that are the result of distinct regulatory mechanisms, and thus the method can discard better differences that are due to noise. The main limitation in this module identification approach is that it requires high quality of prior knowledge on the signaling pathways, whereas many biological systems are only partially known. To overcome this obstacle, the model should be corrected by applying a refinement procedure before elucidating the modules. In the current study, refinement steps that cause global effects, such as novel feedbacks or disconnected networks were not allowed. Hopefully, within the formalism of the presented model, it will be possible to develop techniques to handle those cases as well. Although there is much to be developed both in the modeling and the algorithmic parts, by extending the concepts derived here, it is clear that simultaneous analysis of qualitative knowledge with high throughput data is a

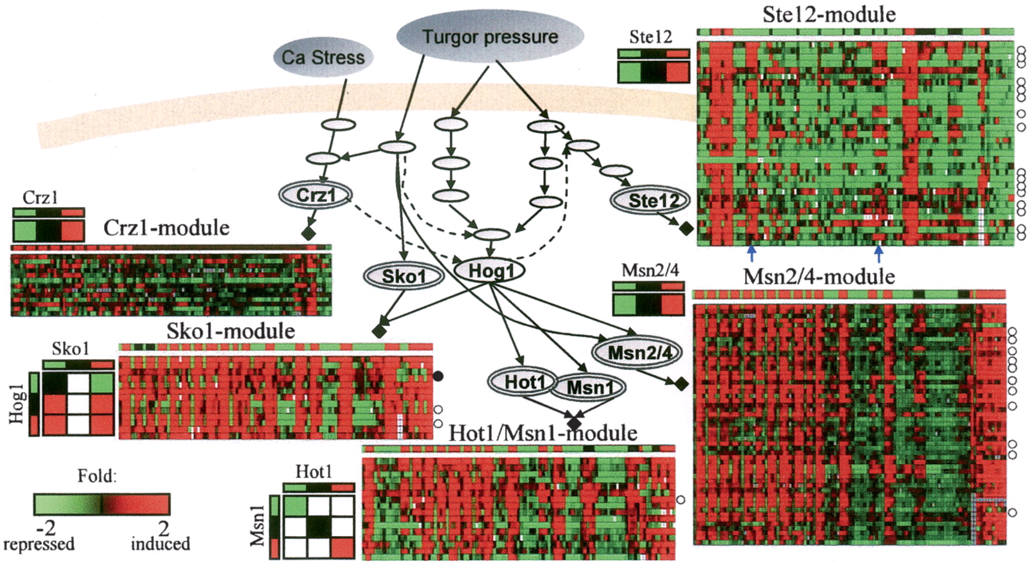


Figure 11.38: Expansion of the osmotic network model [2]. The expansion algorithm assigns known and novel target genes to known modules (black diamonds). Each module is represented by a matrix showing the expression of its target genes (rows) across the 106 conditions (columns). Known target genes that were assigned to their module correctly/incorrectly are marked with white/black dots to the *right* of the corresponding row (known targets were excluded from the model before expansion, to allow validation and to avoid circularity). The predicted expression levels in each condition appear as a separate row *above* the matrix. The logic of each module, obtained by the refinement procedure, appears *near* the matrix. Only logic entries with significant improvements in score are colored. In general, there is high agreement between model predictions and observed levels. The few cases of disagreement (e.g., columns marked by blue arrows in the Ste12 module) highlight our incomplete understanding (and hence modeling) of the biological system.

useful approach. The approach is applicable to other types of perturbations, such as siRNA, to other environmental conditions, such as pharmaceutical agents, and to other molecular data, such as protein activity levels measured by microarrays. High throughput phosphorylation measurements might allow an automated construction of kinase signaling modules using known signaling pathways. As new databases of curated knowledge on signaling pathway are developed (such as BioModels [8], Reactome [6], and SPIKE [9]), it will be easier to obtain the prior information on many biological systems and apply the methodology to them.

Bibliography

[1] Metabolic network modelling [url].

[2] Irit Gat-Viks and Ron Shamir. Refinement and expansion of signaling pathways: The osmotic response network in yeast. *Genome Research*, 17(3):358–367, 2007.

[3] J. Hertz. Statistical issues in the reverse engineering of genetic networks. In *Pacific Symposium on Biocomputing*, 1998.

[4] Gat-Viks I., Tanay A., Raijman D., and Shamir R. A probabilistic methodology for integrating knowledge and experiments on biological networks. *Journal of Computational Biology*, 13(2):165–181, 2006.

[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] G. Joshi-Tope, Marc Gillespie, Imre Vastrik, Peter D'Eustachio, Esther Schmidt, Bernard de Bono, Bijay Jassal, G. R. Gopinath, G. R. Wu, Lisa Matthews, Suzanna Lewis, Ewan Birney, and Lincoln Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33:428–432, 2005.

[7] Per Kraulis. Lecture notes: Metabolic networks [url].

[8] Le Novère N., Bornstein B., Broicher A., Courtot M., Donizelli M., Dharuri H., Li L., Sauro H., Schilstra M., Shapiro B., Snoep J.L., and Hucka M. BioModels Database: A free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids research*, 34:D689–D691, 2006.

[9] Vesterman R., Amit N., Weisz M., Assa J., Ulitsky I., Elkon R., Shamir R., and Shiloh Y. SPIKE software [url].