

Chapter 11

INFERENCE OF GENETIC REGULATORY NETWORKS VIA BEST-FIT EXTENSIONS

Ilya Shmulevich¹, Antti Saarinen², Olli Yli-Harja², and Jaakko Astola²

¹*Cancer Genomics Laboratory, Department of Pathology*

The University of Texas M. D. Anderson Cancer Center, Houston, Texas, USA

²*Institute of Signal Processing, Tampere University of Technology, Tampere, Finland*

1. Introduction

One of the most important breakthroughs in recent years in molecular biology is microarray technology, which allows monitoring of gene expression at the transcript level for thousands of genes in parallel (Schena *et al.*, 1995; Celis *et al.*, 2000). Even though mRNA is not the final product of a gene, armed with the knowledge of gene transcript levels in various cell types, under different developmental stages (Wen *et al.*, 1998), and under a variety of conditions, such as in response to specific stimuli (Iyer *et al.*, 1999; DeRisi *et al.*, 1997), scientists can gain a deeper understanding of the functional roles of genes, of the cellular processes in which they participate, and of their regulatory interactions. Thus, gene expression data for many cell types and organisms at multiple time points and experimental conditions are rapidly becoming available (Brazma and Vilo, 2000). In fact, the amounts of data typically gathered in experiments call for computational methods and formal modeling in order to make meaningful interpretations (Huang, 1999). The emerging view is that as biology becomes a more quantitative science, modeling approaches will become more and more usual (Brazma and Vilo, 2000).

One popular computational approach to gene expression analysis is to compare gene expression *profiles*, that is, the dynamic behavior of genes over time points or cell types, and to apply clustering (Eisen *et al.*, 1998; Ben-Dor and Yakhini, 1999) and data reduction and visualization techniques such as the self-organizing map (Tamayo *et al.*, 1999; Kaski

et al., 2001) (also see Chapter 5) or principle components analysis (Alter *et al.*, 2000; Holter *et al.*, 2000). An inherent assumption in many such approaches is that if two gene profiles are similar, the respective genes are co-regulated and possibly functionally related (Brazma and Vilo, 2000). Although this assumption does not always hold, such methods can nevertheless be useful in uncovering important underlying mechanisms in gene regulation. Another difficulty with these approaches is that currently, there is no theory on how to choose the best distance or similarity measure (Brazma and Vilo, 2000; Shmulevich and Zhang, 2002a) (e.g. correlation coefficient, rank/ordinal correspondence measures, various norms), and each one may lead to possibly very different results. But perhaps a more fundamental criticism that such approaches have received is that they are essentially “genocentric” to use a term of Huang (1999), in that they focus on functions of individual genes.

In order to understand the nature of cellular function, it is necessary to study the behavior of genes in a holistic rather than in an individual manner. A significant role is played by the development and analysis of mathematical and computational methods in order to construct formal models of genetic interactions. This research direction provides insight and a conceptual framework for an integrative view of genetic function and regulation and paves the way toward understanding the complex relationship between the genome and the cell. Moreover, this direction has provided impetus for experimental work directed toward verification of these models.

There have been a number of attempts to model gene regulatory networks, including linear models (van Someren *et al.*, 2000; D’Haeseleer *et al.*, 1999), Bayesian networks (Murphy and Mian, 1999; Friedman *et al.*, 2000), and neural networks (Weaver *et al.*, 1999). The model system that has received, perhaps, the most attention is the so-called *Random Boolean Network* model originally introduced by Kauffman (Kauffman, 1993), approximately thirty years ago. In this model, gene expression is quantized to only two levels: ON and OFF. The expression level (state) of each gene is functionally related to the expression states of some other genes. These connections are represented by the network ‘wiring’.

Recent research seems to indicate that many realistic biological questions may be answered within the seemingly simplistic Boolean formalism, which in essence emphasizes fundamental, generic principles rather than quantitative biochemical details (Huang, 1999; Shmulevich and Zhang, 2002a). Moreover, this is the only model system that has yielded insights into the overall behavior of large genetic networks (Szallasi and Liang, 1998; Wuensche, 1998). For example, the dynamic behavior of such networks corresponds to and can be used to model many biologically

meaningful phenomena, such as, for example cellular state dynamics, possessing switch-like behavior, stability, and hysteresis (Huang, 1999).

Besides the conceptual framework afforded by such models, a number of practical uses may be reaped by inferring the structure of the genetic models from experimental data, that is, from gene expression profiles. One such use is the identification of suitable drug targets in cancer therapy. To that end, much recent work has gone into identifying the structure of gene regulatory networks from expression data (Liang *et al.*, 1998; Akutsu *et al.*, 1999; Akutsu *et al.*, 1998; Akutsu *et al.*, 2000; Shmulevich *et al.*, 2002b).

Most of the work, however, has focused on the so-called Consistency Problem, namely, the problem of determining whether there exists a network that is consistent with the examples. While this problem is important in computational learning theory, as it can be used to prove the hardness of learning for various function classes, it may not be applicable in a realistic situation in which noisy observations or errors are contained, as is the case with microarrays. Measurement errors can arise in the data acquisition process or may be due to unknown latent factors. A learning paradigm that can incorporate such inconsistencies is called the *Best-Fit Extension Problem*. Essentially, the goal of this problem is to establish a rule or in our case, network, that would make as few misclassifications as possible.

In order for an inferential algorithm to be useful, it must be computationally tractable. In this chapter, we consider the computational complexity of the Best-Fit Extension Problem for the Random Boolean Network model. We show that for many classes of Boolean functions, the problem is polynomial-time solvable, implying its practical applicability to real data analysis. We first review the necessary background information on Random Boolean Networks and then discuss the Best-Fit Extension Problem for Boolean functions and its complexity for Boolean networks.

2. Boolean Networks

For consistency of notation with other related work, we will be using the same notation as in (Akutsu *et al.*, 1999). A Boolean network $G(V, F)$ is defined by a set of nodes $V = \{v_1, \dots, v_n\}$ and a list of Boolean functions $F = (f_1, \dots, f_n)$. A Boolean function $f_i(v_{i_1}, \dots, v_{i_k})$ with k specified input nodes is assigned to node v_i . In general, k could be varying as a function of i , but we may define it to be a constant without loss of generality as $k = \max_i k(i)$ and allowing the unnecessary variables (nodes) in each function to be *fictitious*. For a function f , the

variable x_i is fictitious if

$$\begin{aligned} f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) = \\ f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n), \end{aligned}$$

for all $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$. A variable that is not fictitious is called *essential*. We shall also refer to k as the *indegree* of the network. Each node v_i represents the state (expression) of gene i , where $v_i = 1$ means that gene i is expressed and $v_i = 0$ means it is not expressed. The list of Boolean functions F represents how genes regulate each other. That is, any given gene transforms its inputs (regulatory factors that bind to it) into an output, which is the state or expression of the gene itself. All genes (nodes) are updated synchronously in accordance with the functions assigned to them and this process is then repeated. The artificial synchrony simplifies computation while preserving the qualitative, generic properties of global network dynamics (Huang, 1999; Kauffman, 1993; Wuensche, 1998).

To capture the dynamic nature of these networks, it is useful to consider a ‘wiring diagram’ $G'(V', F')$ (Akutsu *et al.*, 1999). Let $k(i)$ be the number of essential variables of function f_i in F . We then construct n additional nodes v'_1, \dots, v'_n and for each $i = 1, \dots, n$, we draw an edge from v_{i_j} to v'_i , for each $1 \leq j \leq k(i)$. Then, $V' = \{v_1, \dots, v_n, v'_1, \dots, v'_n\}$ and the list F' is actually the same as F , but with the functions being assigned to nodes v'_1, \dots, v'_n (with inputs from V) while the functions assigned to v_1, \dots, v_n are just the trivial identity functions, e.g. $f(v_i) = v_i$. In other words, $v'_i = f_i(v_{i_1}, \dots, v_{i_{k(i)}})$ and thus, the expression pattern $\{v_1, \dots, v_n\}$ corresponds to the states of the genes at time t (INPUT) and the pattern $\{v'_1, \dots, v'_n\}$ corresponds to the states of the genes at time $t + 1$ (OUTPUT). Collectively, the states of individual genes in the genome form a *gene activity profile* (GAP) (Huang, 1999).

Consider the state space of a Boolean network with n genes. Then, the number of possible GAPs is equal to 2^n . For every GAP, there is another successor GAP into which the system transitions in accordance with its structural rules as defined by the Boolean functions. Thus, there is a directionality that is intrinsic to the dynamics of such systems. Consequently, the system ultimately transitions into so-called *attractor* states. The states of the system that flow into the same attractor state make up a *basin of attraction* of that attractor (Wuensche, 1998). Sometimes, the system periodically cycles between several *limit-cycle* attractors. It is interesting to note that such behavior even exists for some infinite networks (networks with an infinite number of nodes) (Moran, 1995), such as those in which every Boolean function is the majority function. Moreover, the convergence of a discrete dynamical system to attractors

should be well known to many researchers from the area of non-linear signal processing, where convergence to *root signals* has been studied for many classes of digital filters (Gabbouj *et al.*, 1992). Root signals are those signals that are invariant to further processing by the same filter. Some filters are known to reduce any signal to a root signal after a finite number of passes while others possess cyclic behavior.

Although the large number of possible GAPs would seem to preclude computer-based analysis, simulations show that for networks with low connectivity, only a small number of GAPs actually correspond to attractors (Kauffman, 1993). Since other GAPs are unstable, the system is normally not found in those states unless perturbed. In fact, real genetic regulatory networks, at least in bacteria, are known to have very low connectivity (2 or 3) (Thieffry *et al.*, 1998).

3. The Best-Fit Extension Problem

One of the central goals in the development of network models is the inference of their structure from experimental data. In the strictest sense, this task falls under the umbrella of computational learning theory (Kearns and Vazirani, 1994). Essentially, we are interested in establishing “rules” or, in our case, Boolean functions by observing binary INPUT/OUTPUT relationships. Thus, this task can also be viewed as a system identification problem. One approach is to study the so-called Consistency Problem, considered for Boolean networks in (Akutsu *et al.*, 1999).

The Consistency Problem is important in computational learning theory (Valiant, 1984) and can be thought of as a search of a rule from examples. That is, given some sets T and F of “true” and “false” vectors, respectively, we aim to discover a Boolean function f that takes on the value 1 for all vectors in T and the value 0 for all vectors in F . We may also assume that the target function f is chosen from some class of possible target functions. One important reason for studying the complexity of the consistency problem is its relation to the PAC approximate learning model of Valiant (Valiant, 1984). If the consistency problem for a given class is NP-hard, then this class is not PAC-learnable. Moreover, this would also imply that this class cannot be learned with equivalence queries (Angluin, 1987).

Unfortunately, in realistic situations, we usually encounter errors that may lead to inconsistent examples. This is no doubt the case for gene expression profiles as measured from microarrays, regardless of how the binarization is performed. In order to cope with such inconsistencies, we can relax our requirement and attempt to establish a rule that makes

the minimum number of misclassifications. This is called The Best-Fit Extension Problem and has been extensively studied in (Boros *et al.*, 1998) for many function classes.

We now briefly define the problem for Boolean functions. The generalization to Boolean networks is straightforward. A *partially defined Boolean function* pdBf is defined by a pair of sets (T, F) such that $T, F \subseteq \{0, 1\}^n$, where T is the set of true vectors and F is the set of false vectors. A function f is called an *extension* of $\text{pdBf}(T, F)$ if $T \subseteq T(f)$ and $F \subseteq F(f)$, where $T(f) = \{x \in \{0, 1\}^n : f(x) = 1\}$ and $F(f) = \{x \in \{0, 1\}^n : f(x) = 0\}$. Suppose that we are also given positive weights $w(x)$ for all vectors $x \in T \cup F$ and define $w(S) = \sum_{x \in S} w(x)$ for a subset $S \subseteq T \cup F$ (Boros *et al.*, 1998). Then, the *error size* of function f is defined as

$$\varepsilon(f) = w(T \cap F(f)) + w(F \cap T(f)). \quad (11.1)$$

If $w(x) = 1$ for all $x \in T \cup F$, then the error size is just the number of misclassifications. The goal is then to output subsets T^* and F^* such that $T^* \cap F^* = \emptyset$ and $T^* \cup F^* = T \cup F$ for which the $\text{pdBf}(T^*, F^*)$ has an extension in some class of functions \mathcal{C} (chosen a priori) and so that $w(T^* \cap F) + w(F^* \cap T)$ is minimum. Consequently, any extension $f \in \mathcal{C}$ of $\text{pdBf}(T^*, F^*)$ has minimum error size.

It is clear that the Best-Fit Extension Problem is computationally more difficult than the Consistency Problem, since the latter is a special case of the former, that is, when $\varepsilon(f) = 0$. The computational complexity of these problems has been studied for many function classes in (Boros *et al.*, 1998). For example, the Best-Fit Extension Problem was proved to be polynomially solvable for all transitive classes and some others, while for many classes including threshold, Horn, Unate, positive self-dual, it was shown to be NP-hard.

It is important to note here that if the class \mathcal{C} of functions is not restricted (i.e. all Boolean functions), then an extension exists if and only if T and F are disjoint. This can be checked in $O(|T| \cdot |F| \cdot \text{poly}(n))$ time, where $\text{poly}(n)$ is the time needed to answer “is $x = y$?” for $x \in T$, $y \in F$. This is precisely why attention has been focused on various subclasses of Boolean functions.

For the case of Boolean networks, we are given n partially defined Boolean functions defined by sets $(T_1, F_1), \dots, (T_n, F_n)$. Since we are making “genome-wide” observations, it follows that $|T_1 \cup F_1| = \dots = |T_n \cup F_n| = m$. Given some class of functions \mathcal{C} , we say that the network $G(V, F)$ is consistent with the observations if f_i from F is an extension of $\text{pdBf}(T_i, F_i)$, for all i . In (Akutsu *et al.*, 1999) it was shown that when \mathcal{C} is the class of Boolean functions containing no more than k essential

variables (maximum indegree of the network), the Consistency Problem is polynomially solvable in n and m . In fact, it turns out that if we make no restriction whatsoever on the function class, the Consistency Problem for Boolean networks is still polynomial-time solvable, because for each node v_i , all we need to do is check whether or not $T_i \cap F_i = \emptyset$.

For a restricted class \mathcal{C} , we can say that if the Consistency Problem is polynomially solvable for one Boolean function (i.e. one node), then it is also polynomially solvable for the entire Boolean network, in terms of n and m . The reason is that the time required to construct an extension simply has to be multiplied by n - the number of nodes. For example, as shown in (Akutsu *et al.*, 1999), the time needed to construct one extension from the class of functions with k essential variables (k fixed), is $O(2^{2^k} \cdot n^k \cdot m)$ because there are a total of 2^{2^k} Boolean functions that must be checked for each of the $\binom{n}{k}$ possible combinations of variables and for m observations. Thus, the Consistency Problem for the entire network can be solved in $O(2^{2^k} \cdot n^k \cdot m \cdot n)$ time, for fixed k .

We now see that the same must hold true for the Best-Fit Extension Problem as well. Consider again the class of functions with k essential variables. Then, all we must do is calculate the error size $\varepsilon(f)$ for every Boolean function f , for each of the $\binom{n}{k}$ possible combinations of variables, over all m observations, and keep track of the minimum error size as well as the corresponding function and its variables. To generalize this for a Boolean network, we must simply repeat the process for every one of the n nodes, essentially multiplying the time needed for obtaining a best-fit extension by n . Consequently, the Best-Fit Extension Problem is polynomial-time solvable for Boolean networks, when all functions are assumed to have no more than k essential variables. Moreover, if \mathcal{C} is the class of all Boolean functions (i.e. no restrictions), then the Best-Fit Extension Problem for Boolean networks can also be solved in polynomial time by virtue of it being polynomially solvable for general Boolean functions (see (Boros *et al.*, 1998)). So, we can say the following:

Proposition 1 *If it is known that the Best-Fit Extension Problem is solvable in polynomial time in n and m for one Boolean function from class \mathcal{C} , then the Best-Fit Extension Problem has a polynomial time solution for a Boolean network in which all functions belong to class \mathcal{C} .*

For example, it is known that for the class of monotone (positive) Boolean functions, the Boolean function version of the Best-Fit Extension Problem is polynomially solvable (Boros *et al.*, 1998). Then, it immediately follows that the Boolean network version of the Best-Fit Extension Problem is also polynomial-time solvable.

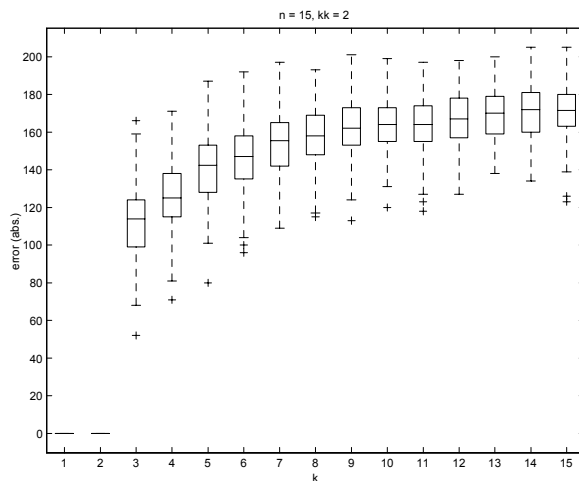


Figure 11.1. Results from the first simulation. The x-axis shows the indegree k of the network, which was used to obtain the original OUTPUT matrix. The y-axis shows the absolute error (i.e. the number of wrong OUTPUTs in the OUTPUT matrix of the network, which was inferred with the Best-Fit method. In this figure, medians, upper and lower quartiles, and extreme values are shown.

4. Simulation Analysis

The idea behind the following simulations was to computationally assess the behavior of the error attained by the Best-Fit method, when presented with given INPUT-OUTPUT pairs, as described in Section 3.

In the first simulation, networks with $n = 15$ (i.e. the number of genes) were used. First, a network with indegree k (in this simulation k varied from 1 to 15) was constructed. Then, 50 INPUTs were created randomly and the constructed network was used to produce 50 OUTPUTs, forming a so-called OUTPUT matrix. For example, for a network with 15 genes, the OUTPUT matrix is of size 15×50 . After this, the Best-Fit method was used to infer a network from these 50 INPUT-OUTPUT pairs. For the error in Eq. (11.1), we used $w(x) = 1$ for all vectors, thus simply counting the number of wrong OUTPUTs.

The inference was constrained so that the indegree of the inferred network was set to be $kk = 2$. Another OUTPUT matrix was produced from the inferred network with the same 50 INPUTs. After having these two OUTPUT matrices, the error was calculated by comparing the corresponding elements of these matrices. Every wrong OUTPUT in the matrix that was derived from the inferred network increased the

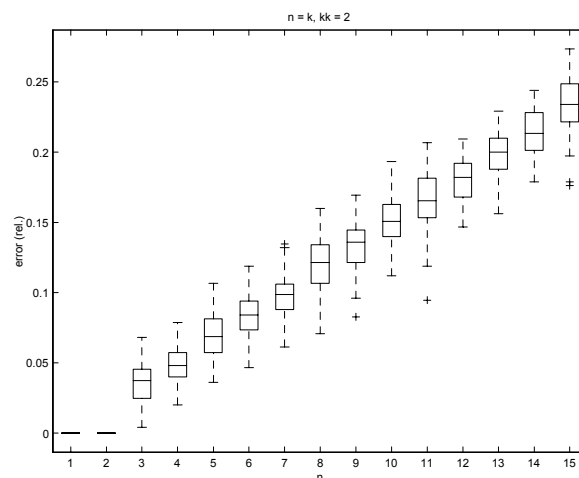


Figure 11.2. Results from the second simulation. The x-axis shows the number of genes in the network. In this simulation, this is also the indegree of that network (i.e. $n = k$). The y-axis shows the relative error; i.e. the number of wrong OUTPUTs was normalized by dividing it by the number of all OUTPUTs. Medians, upper and lower quartiles, and extreme values are presented.

error by one. Thus, because the OUTPUT matrices were of size 15×50 , the maximum error was 750.

To produce Figure 11.1, the procedure described above was repeated 250 times for each value of k . Thus, for each k we have 250 different errors that were produced. The x-axis shows the indegree k that was used to create the network that generated the original OUTPUT matrix. The y-axis contains the absolute error, that is, the number of wrong elements in the OUTPUT matrix generated by the inferred network. The figure shows the median of the errors as well as the upper and the lower quartiles. The outliers are also presented and are marked with a '+' symbol. The figure indicates that the behavior of the error appears to be logarithmic.

In the second simulation, we took a different approach. Rather than having a fixed network size n , we set $n = k$ for every indegree k . Then, methods similar to those in the first simulation were used. For every k , a network with $n = k$ was constructed and 50 INPUTs were randomly created. Then an OUTPUT matrix was produced from the constructed network. At this point, the OUTPUT matrix was of size $n \times 50$. As before, the Best-Fit method was used to infer a network from these INPUT-OUTPUT pairs, but under the constraint that the indegree was fixed to be $kk = 2$. The number of wrong OUTPUTs was calculated

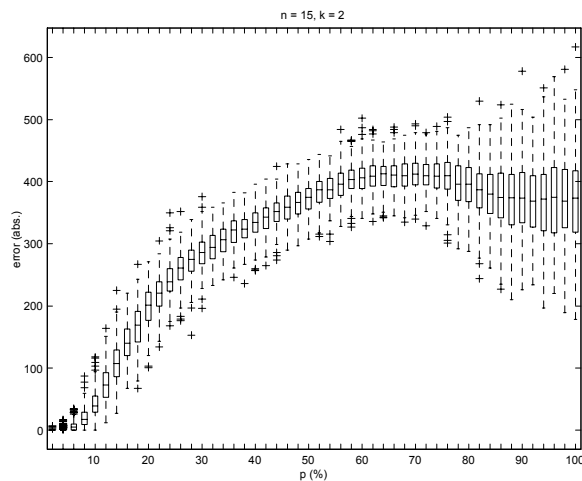


Figure 11.3. Results from the third simulation. The x-axis shows the probability p (in this simulation 2%, 4%, 6%, ..., 98%, 100%) and the y-axis is the absolute error (i.e. the number of wrong OUTPUTs in the OUTPUT-matrix). Medians, upper and lower quartiles, and extreme values are presented.

as before. However, because for each k the OUTPUT matrix was of a different size, the number of wrong OUTPUTs was normalized by dividing by the number of all OUTPUTs (i.e., the percentage of wrong OUTPUTs was calculated). This way the errors in each network could be compared.

To produce Figure 11.2, we repeated the procedure above 250 times. The x-axis shows the network size n and the y-axis shows the relative error. As in Figure 11.1, the medians, upper and lower quartiles, as well as the extreme values are presented. As opposed to the first figure, the behavior of the relative error seems to be linear rather than logarithmic.

Finally, we performed a third simulation designed to observe networks that are inferred from noisy data. Such a situation was considered by Akutsu *et al.* (2000), who proposed so-called noisy Boolean networks together with an identification algorithm, in order to deal with noise present in expression patterns. In that model, they relax the requirement of consistency intrinsically imposed by the Boolean functions. In our simulation, we first constructed a network with $n = 15$ and $k = 2$. Then 50 random INPUTs were created and the corresponding 50 OUTPUTs were derived from the constructed network. Then each bit (both in the INPUT- and OUTPUT-matrix) was flipped with probability p . After this the Best-Fit method was used to infer a network from these

matrices. Having done that, another OUTPUT-matrix was derived from the inferred network with the original INPUT-matrix (i.e., the matrix which has no flipped bits). Finally, the error was calculated by comparing the derived OUTPUT-matrix to the original OUTPUT-matrix. In this simulation $w(x) = 1$ for every vector x so the error is simply the number of wrong OUTPUTs in the OUTPUT-matrix. The results from this simulation are presented in Figure 11.3.

5. Conclusions

The ability to efficiently infer the structure of Boolean networks has immense potential for understanding the regulatory interactions in real genetic networks. We have considered a learning strategy that is well suited for situations in which inconsistencies in observations are likely to occur. This strategy produces a Boolean network that makes as few misclassifications as possible and is a generalization of the well-known Consistency Problem. We have focused on the computational complexity of this problem. It turns out that for many function classes, the Best-Fit Extension Problem for Boolean networks is polynomial-time solvable, including those networks having bounded indegree and those in which no assumptions whatsoever about the functions are made. This promising result provides motivation for developing efficient algorithms for inferring network structures from gene expression data.

References

- Akutsu, T., Kuhara, S., Maruyama, O. and Miyano, S. (1998) Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. *Proc. the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, 695-702.
- Akutsu, T., Miyano, S. and Kuhara, S. (1999) Identification of Genetic Networks from a Small Number of Gene Expression Patterns Under the Boolean Network Model. *Pacific Symposium on Biocomputing* 4, 17-28.
- Akutsu, T., Miyano, S., and Kuhara, S. (2000) Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics* 16, 727-734.
- Alter, O., Brown, P. O. and Botstein, (2000) D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA* 97:18, 10101-10106.
- Angluin, D. (1987) Learning regular sets from queries and counterexamples. *Information and Computation*, 75:2, 87-106.

- Ben-Dor, A. and Yakhini, Z. (1999) Clustering Gene Expression Patterns *Proc. of the 3rd International Conference on Computational Molecular Biology, 3342*. Lyon, France: ACM Press.
- Boros, E., Ibaraki, T., and Makino, K. (1998) Error-Free and Best-Fit Extensions of Partially Defined Boolean Functions. *Information and Computation*, 140, 254-283.
- Brazma, A. and Vilo, J. (2000) Gene expression data analysis. *FEBS Letters* 480, 17-24.
- Celis, J. E., Kruhøffer, M., Gromova, I., Frederiksen, C., Østergaard, M., Thykjaer, T., Gromov, P., Yu, J., Pálisdóttir, H., Magnusson, N., & Ørntoft, T. F. (2000) Gene expression profiling: monitoring transcription and translation products using DNA microarrays and proteomics. *FEBS Letters* 480, 2-16.
- D'Haeseleer, P., Wen, X., Fuhrman, S. and Somogyi, R. (1999) Linear modeling of mRNA expression levels during CNS development and injury *Pacific Symposium on Biocomputing*, 4, 41-52.
- DeRisi, J.L., Iyer, V.R., and Brown, P.O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278, 680-686.
- Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95, 14863-14868.
- Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000) Using Bayesian Network to Analyze Expression Data. *Journal of Computational Biology*, 7, 601-620.
- Gabbouj, M., Yu, P-T., and Coyle, E. J. (1992) Convergence behavior and root signal sets of stack filters. *Circuits Systems & Signal Processing*, 11:1, 171-193.
- Holter, N. S., Mitra, M., Maritan, A., Cieplak, M., Banavar, J. R., & Fedoroff, N. V. (2000) Fundamental patterns underlying gene expression profiles: Simplicity from complexity. *Proc. Natl. Acad. Sci. USA*, 97, 8409-8414.
- Huang, S. (1999) Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine* 77, 469-480.
- Iyer, V. R., Eisen, M. B., Ross, D. T., Schuler, G., Moore, T., Lee, J. C. F., Trent, J. M., Staudt, L. M., Hudson Jr., J., Boguski, M. S., Lashkari, D., Shalon, D., Botstein, D., and Brown, P. O. (1999) The transcriptional program in the response of human fibroblasts to serum. *Science*, 283, 83-87.
- Kaski, S., Nikkilä, J., Törrönen, P., Castrén, E., and Wong, G. (2001) Analysis and visualization of gene expression data using self-

- organizing maps. *IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP-01)*, Baltimore, Maryland, June 3-6.
- Kauffman, S. A. (1993) *The origins of order: Self-organization and selection in evolution*, Oxford University Press, New York.
- Kearns, M. J. and Vazirani, U. V. (1994) *An Introduction to Computational Learning Theory*, MIT Press.
- Liang, S., Fuhrman, S. and Somogyi, R. (1998) REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures. *Pacific Symposium on Biocomputing* 3, 18-29.
- Moran, G. (1995) On the period-two-property of the majority operator in infinite graphs. *Trans. Amer. Math. Soc.* 347, No. 5, 1649-1667.
- Murphy, K. and Mian, S. (1999) Modelling Gene Expression Data using Dynamic Bayesian Networks. *Technical Report, University of California, Berkeley*.
- Schena, M., Shalon, D., Davis, R. W., and Brown, P.O. (1995) Quantitative monitoring of gene expression pattern with a complementing DNA microarray. *Science*, 270, 467-470.
- Shmulevich, I. and Zhang, W. (in press) Binary Analysis and Optimization-Based Normalization of Gene Expression Data, *Bioinformatics*.
- Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (in press) Probabilistic Boolean Networks: A Rule-based Uncertainty Model for Gene Regulatory Networks. *Bioinformatics*.
- Szallasi, Z. and Liang, S. (1998) Modeling the Normal and Neoplastic Cell Cycle With Realistic Boolean Genetic Networks: Their Application for Understanding Carcinogenesis and Assessing Therapeutic Strategies. *Pacific Symposium on Biocomputing* 3, 66-76.
- Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., & Golub, T. R. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA* 96, 2907-2912.
- Thieffry, D., Huerta, A. M., Pérez-Rueda, E., and Collado-Vides, J. (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*. *BioEssays*, 20:5, 433-440.
- van Someren, E. P., Wessels, L.F.A., and Reinders, M.J.T. (2000) Linear modeling of genetic networks from experimental data. *Intelligent Systems for Molecular Biology (ISMB 2000)*, San Diego, August 19-23.
- Valiant, L. G. (1984) A theory of the learnable. *Comm. Assoc. Comput. Mach.*, 27, 1134-1142.

- Weaver, D.C., Workman, C.T. and Stormo, G.D. (1999) Modeling Regulatory Networks with Weight Matrices. *Pacific Symposium on Biocomputing*, 4, 112-123.
- Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, J. L., and Somogyi, R. (1998) Large-Scale Temporal Gene Expression Mapping of Central Nervous System Development. *Proc Natl Acad Sci USA*, 95, 334-339.
- Wuensche, A. (1998) Genomic Regulation Modeled as a Network with Basins of Attraction. *Pacific Symp. on Biocomp.* 3, 89-102.