BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 7 2007, pages 866–874
doi:10.1093/bioinformatics/btm021

Systems biology

Boolean dynamics of genetic regulatory networks inferred from microarray time series data

Shawn Martin[1], Zhaoduo Zhang[2], Anthony Martino[3] and Jean-Loup Faulon[4,*]

[1]Sandia National Laboratories, Computational Biology Department PO Box 5800, Albuquerque, NM, 87185-1316, USA, [2]Sandia National Laboratories, Biosystems Research, PO Box 969, Livermore, CA 94551-9291, USA [3]Sandia National Laboratories, Biomolecular Analysis and Imaging, PO Box 5800, Albuquerque, NM 87185-0895, USA and [4]Sandia National Laboratories, Computational Biosciences Department PO Box 5800, Albuquerque, NM 87185-1413, USA

Received on September 14, 2006; revised on December 29, 2006; accepted on January 19, 2007

Advance Access publication January 31, 2007

Associate Editor: Golan Yona

ABSTRACT

Motivation: Methods available for the inference of genetic regulatory networks strive to produce a single network, usually by optimizing some quantity to fit the experimental observations. In this article we investigate the possibility that multiple networks can be inferred, all resulting in similar dynamics. This idea is motivated by theoretical work which suggests that biological networks are robust and adaptable to change, and that the overall behavior of a genetic regulatory network might be captured in terms of dynamical basins of attraction.

Results: We have developed and implemented a method for inferring genetic regulatory networks for time series microarray data. Our method first clusters and discretizes the gene expression data using *k*-means and support vector regression. We then enumerate Boolean activation–inhibition networks to match the discretized data. Finally, the dynamics of the Boolean networks are examined. We have tested our method on two immunology microarray datasets: an IL-2-stimulated T cell response dataset and a LPS-stimulated macrophage response dataset. In both cases, we discovered that many networks matched the data, and that most of these networks had similar dynamics.

Contact: jfaulon@sandia.gov

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Modeling and inferring genetic regulatory networks are important problems in systems biology. Accordingly, there are numerous computational approaches to these problems, including partial differential equations, ordinary differential equations, Bayesian networks and Boolean networks (de Jong, 2002; Smolen *et al*., 2000). Less common approaches include Petri nets (Goss and Peccoud, 1998) and matrix decomposition methods (Alter and Golub, 2005; Liao *et al*., 2003).

Of these different approaches, one of the simplest methods is the use of Boolean networks to infer genetic regulatory systems from time series gene expression data. Modeling genetic regulatory networks was first proposed in Kauffman (1969, 1993). Later Boolean networks were proposed to infer genetic regulatory systems from time series gene expression data (Akutsu *et al*., 2000; Liang *et al*., 1998). More recently, probabilistic Boolean networks have been proposed to infer genetic networks (Shmulevich *et al*., 2002a,b). Also related to probabilistic Boolean networks are dynamic Bayesian networks (Friedman *et al*., 2000; Lahdesmaki *et al*., 2006; Murphy and Mian, 1999).

Once a network has been inferred, the next step is to consider its dynamical properties. Huang (1999) suggests that Boolean network dynamics can be used to understand cellular states such as proliferation, differentiation and apoptosis. By analyzing the attractor basins of a Boolean network, it may be possible to determine its cellular functions under different initial conditions. In this article, we propose a method for the investigation of these claims using time series gene expression data.

Although the probabilistic Boolean and dynamic Bayesian methods are both well founded theoretically, they are also computationally complex (Ching *et al*., 2005; Zou and Conzen, 2005). Furthermore, both methods produce a single most probable network. This network is in reality one of many possible representations. This fact makes the interpretation of the network dynamics difficult. In the best case, Monte Carlo approaches can be used with probabilistic Boolean networks to approximate dynamics (Shmulevich *et al*., 2003). There are also theoretical results (Brun *et al*., 2005).

In our investigation of network dynamics inferred from time series gene expression data, we do not employ either probabilistic Boolean networks or dynamic Bayesian networks. Instead, we consider the use of dynamics in combination with the simpler qualitative Boolean methods suggested in Akutsu *et al*. (2000). Unlike Akutsu *et al*. (2000), we do not attempt to obtain only a single network to match the data. We consider all possible networks that match the data and group them by attractor basin. We find that while a great many networks can

*To whom correspondence should be addressed.

match a given dataset, a very large percentage of these networks fall into the same attractor basins, suggesting that many of the networks are equally valid. Further, we find that many of these networks have locally consistent substructures. These results agree with the general intuition that biological systems are modular, robust and can often function adequately despite extreme change (Wagner, 2005).

In terms of Bayesian and dynamic Bayesian networks, our approach is most similar to the work in Friedman and Koller (2003) and Segal *et al*. (2005). In Segal *et al*. (2005), data points are grouped as nodes in a Bayesian network in order to obtain networks of modules and in Friedman and Koller (2003) a Bayesian approach is used to discover multiple Bayesian networks matching a given dataset. The work in Segal *et al*. (2005) is also applied to microarray data. These methods differ from our approach in that they do not take into account dynamics and attractors, and that they model networks as directed acyclic graphs, thus prohibiting feedback loops. Nevertheless, if the two methods were combined with dynamic Bayesian networks (Murphy and Mian, 1999; Zou and Conzen, 2005), the result would be very similar to the approach we have taken.

In the following sections, we describe our method and the results of the method applied to two time series gene expression datasets. In Section 2, we describe the datasets and normalization procedures, our method for clustering and discretizing the datasets and our inference algorithms. In Section 3, we apply our method to an interleukin (IL-2)-stimulated T cell immune response dataset and a LPS-stimulated macrophage response dataset. In the Section 4, we summarize the advantages and disadvantages of using dynamics to select Boolean models inferred from time series gene expression data.

2 METHODS

2.1 Datasets and normalization

We used two gene expression time series datasets to benchmark our approach. The first dataset is a time series gene expression dataset taken from an IL-2-stimulated immune response experiment performed at Sandia National Laboratories using arrays hybridized by the Stanford PAN Biotechnology Facility. The experiment was performed using a murine T cell line called CTLL-2. Mouse CTLL-2 cells were cultured without IL-2 (IL-2 starvation). Cells were then collected (0 h, no IL-2 stimulation) immediately before IL-2 was added (IL-2 stimulation). Cells were further collected at 11 time points: 15, 30 mins, 1, 2, 4, 6, 8, 10, 12, 16 and 24 h, all after IL-2 stimulation. Three replicates were done for each time point.

Affymetrix GeneChip Mouse Genome 430 2.0 Arrays were used for the gene expression experiments. This array provides complete coverage of the transcribed mouse genome, with 45 000 probe sets to analyze the expression level of over 39 000 transcripts from over 34 000 well-characterized mouse genes. Target hybridization was processed following the manufacturer's recommendation using the instrument operated by Affymetrix GeneChip Operating Software (GCOS) version 1.3 and Microarray Suite version 5.1 (MAS 5.1). The fluorescent intensity of each probe was quantified using MAS 5.1 and GCOS 1.3. This software makes a detection call (present [P], marginal [M] or absent [A]) for each gene or probe set. This call is based on the consistency of the performance of the individual probe pairs, the hybridization above background and the signal-to-noise ratio. Two-way comparisons of the microarray data were also performed using GCOS 1.3. Specifically, changes in gene expression between the control cells (time point 0 h, no IL-2 stimulation) and IL-2 stimulated cells were evaluated at each time point. These comparisons provided data including the signal log ratio (fold change presented in logarithmic form) and the 'change call' (increased [I], decreased [D], marginally increased [MI], marginally decreased [MD] or no change [NC]) for each gene being interrogated.

To identify genes that exhibited differences in expression between the control cells and IL-2-stimulated cells, the datasets were trimmed using the following inclusion criteria. For a probe set to be included in this trimmed dataset, it had to display in all the three replicates: (1) a change call other than no change (NC), (2) the same trend of change call (I, increase, D, decrease), (3) a present call (P) and/or signal intensity $\geq$100 and (4) at least a 1.5-fold difference in expression between the two compared conditions. The trimmed dataset was log normalized and mean subtracted.

The second dataset was a LPS-stimulated macrophage response dataset downloaded from the cell signaling gateway at http://www.signaling-gateway.org. In this dataset, RAW264.7 murine macrophage cell lines were stimulated by LPS. The response was measured using microarrays. The measurements were made at six time points (1, 2, 4, 8, 16, and 32 hs) with six replicates from each time point. Normalization consisted of first removing all genes without names as well as fiducials. Replicates were treated as new experiments, resulting in additional virtual genes. We removed virtual genes with missing data and screened out genes with expression $<$0.05 SD over time. After clustering the resulting dataset (as described in the following section), the virtual genes were mapped back to the original genes by a voting method. In this method, each replicate 'votes' for the cluster to which it belongs. The original gene then belongs to the cluster with the most votes. Ties are broken by weighting votes according to the distance of the replicates to their cluster centers.

2.2 Clustering and discretization

After normalization, we clustered the time series microarray data. We clustered the data because many co-regulated genes are indistinguishable when they are discretized. Clustering the microarray data also simplified the task of network inference by reducing the problem size. Finally, the clustering resulted in networks of gene groups, instead of actual genes. The gene groups, which we call meta-genes, made the biological analysis and interpretation of the inferred networks more tractable.

There are a variety of algorithms available for clustering microarray data, including k-means, hierarchical clustering (Eisen *et al*., 1998), self-organizing maps (SOMs) (Tamayo *et al*., 1999), and biclustering (Getz *et al*., 2000). We chose k-means because it was the simplest method that provided a partition of our data into groups. We also considered the use of hierarchical clustering and SOMs (as discussed further in the online supplement), but avoided them because they are both heavily oriented towards discovering relations between clusters for visualization, unnecessary for our purposes. We did not use biclustering because it partitions both rows and columns of a matrix and is therefore inappropriate (without modification) for time series data (Zhang *et al*., 2005). A comparison of k-means with hierarchical clustering and SOMs using the IL-2 dataset can be found in the online supplement.

To decide how many clusters should be produced (the value of k), we developed a measure of internal consistency. Our measure is defined for a given partition of the dataset using the singular value decomposition (SVD) (Trefethen and Bau, 1997). To define internal consistency, suppose we are given k and we compute a k-means partition of our $m \times n$ dataset X, where m is the number of time points and n is the number of genes. For the jth cluster ($j = 1, \ldots, k$) we have a matrix X_j of microarray measurements, where the rows are time points and the columns are genes, so that X_j is a $m \times g_j$ matrix, where g_j is the number

of genes in the jth cluster. Using the SVD, we decompose $X_j = U_j S_j V_j^T$, where U_j and V_j are orthogonal matrices and S_j is a diagonal matrix whose entries describe the importance of the columns of U_j and V_j. The matrix $S_j V_j^T = U_j^T X_j$ contains the projections of the columns (time courses) of X_j onto the basis U_j. The entries of S_j (singular values) give the relative importance of the columns of U_j. If the first entry of S_j is much larger than the second entry then we know that most of the information in the columns of X_j is captured by a single dimension. We thus define the internal consistency of the jth cluster to be the ratio of the first and second singular values in S_j. This is a measure of the correlation between all of the time courses in the jth cluster. The internal consistency also provides a measure of how well a single dimension can describe all the time courses. For the problem of network inference, we want each of our clusters to have a high internal consistency. We can decide how many clusters we should use by comparing the average internal consistency of different partitions of the dataset (different values of k) and choosing the clustering with the best average internal consistency.

Given an appropriate set of meta-genes, we next discretized the meta-gene expression levels. Such a discretization is necessary because we use a Boolean network inference algorithm. Our discretization is accomplished in two steps. First, support vector regression (SVR) (Smola and Scholkpof, 1998) is used to provide a continuous, smooth representation of the 10 genes closest to the cluster center in a given group. This type of regression is performed by solving a quadratic programming problem and has two parameters: an ε width and a kernel function. The ε width is used to encapsulate the curves in a given meta-gene group within an ε-tube, and the kernel function is used to fit different types of curves (e.g. linear or non-linear). The end result of SVR encapsulates the time courses in a meta-gene group within an ε-tube centered around a smooth curve, where the curve is a linear combination of kernel functions. For this work, we used Gaussian kernels with a width $\sigma = 1$, and we chose ε to be one and a half times the average standard deviation of the values at each time point.

The second step in our discretization consists of thresholding the curve obtained by the SVR. Assuming that the meta-gene group is well represented by the SVR curve, we can produce a discrete version of the meta-gene by thresholding the curve against its average value: a higher than average meta-gene expression is given a value of 1 (up-regulated), while a lower than average meta-gene expression value is given a value of 0 (down-regulated).

2.3 Boolean inference and network dynamics

After the data has been clustered and discretized, we then infer and analyze the regulatory network. Our algorithms for network inference are similar to the algorithms presented in Akutsu *et al.* (2000). We incorporate potential errors (mismatches), we limit the number of possible inputs to a Boolean function and we restrict our output to activation or inhibition Boolean functions. Our algorithms are different from the algorithms in Akutsu *et al.* (2000) in that we do not place any restrictions on the amount of data necessary to perform an inference. Instead of requiring enough data to infer a unique network, we consider all possible networks matching the data. Pseudo-code for our algorithms (denoted using `ALL_CAPS` in this section), as well as additional explanation, can be found in the online supplement.

Our algorithms count, sample, enumerate, identify attractors and simulate the dynamics of all the possible networks matching a given set of discretized expression profiles. Networks are counted, sampled and enumerated at the node (meta-gene) level. For every node v, we determine all the possible sets of nodes that might control the expression profile of v. Expression profiles are given over time and the algorithms can accept several time courses corresponding to different initial conditions. These initial conditions can be different stimuli, or various knockout experiments.

The algorithms take as input a set of n nodes $V = \{v_1, v_2, \ldots, v_n\}$ corresponding to the meta-genes and a set of discretized expression profiles. For any given profile in the set, the expression of every node is specified over the time course, although different profiles are not required to have the same time course length. To avoid constructing redundant networks, we require that different nodes should have different profiles, but this requirement is not necessary to run the algorithms.

The basic step used by the algorithms is `INFER_FUNCTION`, a routine that determines if a set of nodes $v_1, v_2, \ldots, v_q$ with $q \leq n$ can explain the expression profile of a given node v_i. `INFER_FUNCTION` returns the activation–inhibition Boolean function by which $v_1, v_2, \ldots, v_q$ control the expression of v_i. An activation–inhibition function is a Boolean function of the form $v(t) = (v_1(t)$ OR $v_2(t)$ OR $\ldots)$ `AND NOT` $(v_j(t)$ OR $v_{j+1}(t)$ OR $\ldots)$, where $v_1(t)$, $v_2(t), \ldots$ are activators and $v_j(t)$, $v_{j+1}(t), \ldots$ are inhibitors. As an example, suppose we are given a time series with six time points for three genes v_1, v_2 and v_3, as shown in Table 1, where we write 1 when the gene is up-regulated and 0 when down-regulated. The Boolean function for v_3 in terms of v_1 and v_2 as returned by `INFER_FUNCTION` is given by $v_3(t+1) = v_1(t)$ AND NOT $v_2(t)$. In other words, `INFER_FUNCTION` finds that v_3 is activated by v_1 and inhibited by v_2.

Using `INFER_FUNCTION`, we infer Boolean networks by processing each node in sequence. This is done using the `INFER_NETWORKS` routine, which returns possible connections within a network and the associated Boolean functions for each node v_i. To count the number of possible networks matching a given set of expression profiles we use `COUNT_NETWORKS`. `COUNT_ NETWORKS` runs `INFER_NETWORKS` and computes the product of the number of possible inputs for each node. We have also coded `SAMPLE_NETWORKS`, which first runs `INFER_NETWORKS` and then for each node selects at random one of its possible inputs. Finally, to enumerate all networks, we use `ENUMERATE_NETWORKS`. `ENUMERATE_NETWORKS` first runs `INFER_NETWORKS` and then lists and prints all possible inputs for each node.

Using `INFER_NETWORKS`, `SAMPLE_NETWORKS` and `ENUMERATE_NETWORKS`, we can infer networks using sampling and enumeration. In addition, we can explore the dynamics of the inferred networks. The dynamics of these networks are used to (1) verify that the expression profiles given as input can indeed be reproduced by these inferred networks, (2) explore the dynamics beyond the times series that were provided as input and (3) predict expression profiles under different initial conditions. Of particular interest is computing the steady state or equilibrium dynamics of the networks (Huang, 1999). These steady states are called attractors (Kauffman, 1969). An attractor is a cyclic pattern of expression that all networks will eventually exhibit (due to the finite nature of Boolean networks).

We use two additional routines to locate attractors. First we use `RUN_NETWORK`, which takes as input an inferred network along with

Table 1. An example of discrete time courses with three genes and six time points. Zero (light grey) denotes down-regulation and 1 (dark grey) denotes up-regulation. In this example, we see that $v_3(t+1) = v_1(t)$ AND NOT $v_2(t)$. We say that v_1 activates v_3 and v_2 inhibits v_3

		Time					
		1	2	3	4	5	6
Gene	v_1	0	0	1	1	0	0
	v_2	1	0	0	1	1	0
	v_3	1	0	0	1	0	0

initial conditions and returns the resulting expression of the genes up to some time T. Then we use ATTRACTOR to find attractors. ATTRACTOR takes as input expression profiles given up to time T, and identifies the time step t_1 that an attractor is found.

To finish this section, we briefly remark on the computational complexity of our inference algorithms. Recall that n is the number of nodes in the networks and q is the maximum number of inputs to a node. While q is in theory unbounded and can be equal to n, we restrict q to be no greater than 5. As justification for this choice, we note that gene regulatory networks follow a power law with exponent greater than 2 (Basso *et al.*, 2005), so that q should not be greater than 5 for <100 nodes. In addition, our experiences inferring parsimonious networks (minimum number of edges) indicates that q never exceeds 3.

Let P be the number of expression profiles and T be the number of time points. We assume that P and T are constant independent of n. INFER_FUNCTION runs in $O(PT)$ time steps. INFER_NETWORKS runs in $O(nPTn+nPTn^2+nPTn^3+\cdot+nPTn^q)=O(n^{q+1})$ steps. COUNT_NETWORKS and SAMPLE_NETWORKS both run in $O(n)$ steps, while ENUMERATE_NETWORKS runs in $O(nI)$ where I is the number of possible input vectors (i.e. the number of solutions). Note that this number can be exponential in n, and thus ENUMERATE_NETWORKS can run for an exponential time and can output an exponential number of solutions. Both RUN_NETWORK and ATTRACTOR run in $O(nT)$ steps. Although some of these routines are computationally complex (most notably INFER_ NETWORKS), the run time of the algorithms can be controlled by using a smaller number k of meta-genes; using a smaller number q of inputs for the Boolean functions; using SAMPLE_NETWORKS instead of ENUMERATE_NETWORKS; and/or using a shorter maximum time T when simulating the networks.

3 RESULTS

3.1 IL-2-stimulated T cell immune response

The T cell immune response dataset consisted of mouse microarrays with 45 119 probes per array taken at 12 time points. Normalization reduced the datasets to 5085 probes. After normalization, we performed clustering and discretization. We first computed the appropriate value of k to use in k-means. In Figure 1, we show the value of the mean internal consistency versus k for $k=2,\ldots,40$. The internal consistency increases until k is ~25 then flattens out. We chose a small local peak at $k=23$. Using k-means with $k=23$, we obtained the partition of the full dataset shown in Figure 2.

Using the 23 clusters obtained by k-means, we computed the SVR representations of the meta-genes. The SVR representations were computed using the 10 time courses closest to the

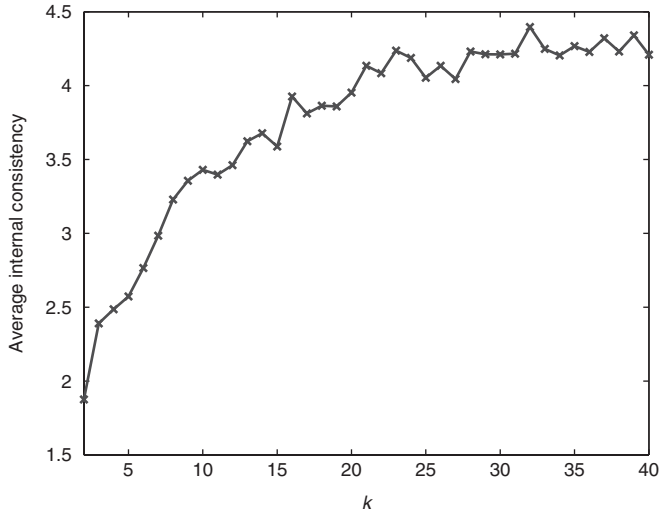

Fig. 1. Average internal consistency as a function for k for the IL-2 immune response dataset. Based on this curve we selected $k=23$.

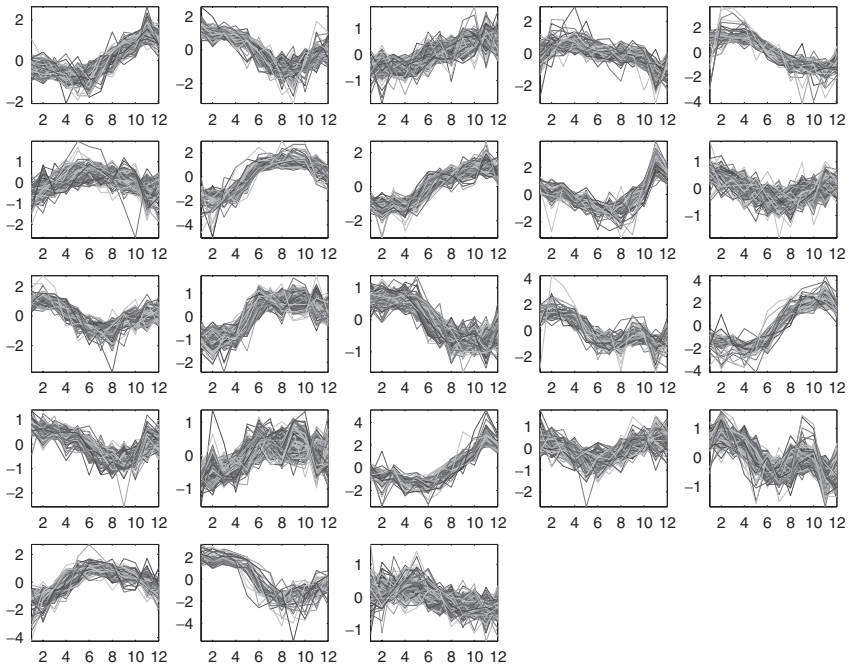

Fig. 2. Profiles of the 23 meta-genes used in our analysis.

cluster centers and were discretized by comparison with the average expression value of the representations. An example of the SVR representation for cluster 6 (Bcl3) is shown in Figure 3.

The discretization resulted in 23 discrete profiles, which were further reduced to 12 unique profiles. These 12 profiles are shown in Table 2. We also performed a comparison of our final discretization with discretizations that we would have obtained using alternate clustering algorithms (hierarchical clustering and SOMs). This comparison, which can be found in the online supplement, revealed that for $k = 23$ there was >80% agreement between discretization regardless of clustering algorithm.

We used the discrete profiles shown in Table 2 as input to the network inference algorithms. These profiles grouped naturally into two distinct sets. The first set consisted of measurements made prior to 1 h and represented the IL-2 starvation state. The second set consisted of measurements made after 1 h and represented the IL-2-stimulated state. These two groups (IL-2 starved and IL-2 stimulated) were separated and used as input (simultaneously) to `INFER_NETWORKS` and `ENUMERATE_NETWORKS`. The inference algorithms discovered a total of 161 558 networks.

The dynamics of the 161 558 networks were analyzed using the `RUN_NETWORK` and `ATTRACTOR` routines. Attractors were determined for the two initial conditions corresponding to IL-2 starved and IL-2 stimulated. It was found that 160 657 (99.4%) of these networks had a single fixed point steady-state dynamic for the IL-2-stimulated initial conditions. In the case of IL-2 stimulation gene expression should fluctuate as IL-2 stimulated T cells proliferate (Nelson and Willerford, 1998). We therefore discarded these 160 657 networks, leaving 901 (0.6%) networks to be interpreted. These 901 networks had the same steady-state dynamic, that dynamic consisting of three time points, shown in Table 3.

An example illustrating our model correctly describing the steady-state fluctuation of genes due to IL-2 stimulation is given by the cyclin-dependent kinase inhibitor p27 (AFFX ID 1419497_at). This inhibitor has been shown experimentally to fluctuate during proliferation of endothelial cells (Huang and Ingber, 2000). Consistent with these results, we found p27 to fluctuate at steady state with cluster E-Stat5b (Table 3).

The 901 networks were analyzed for similarities, yielding the consensus network shown in Figure 4. The viable networks

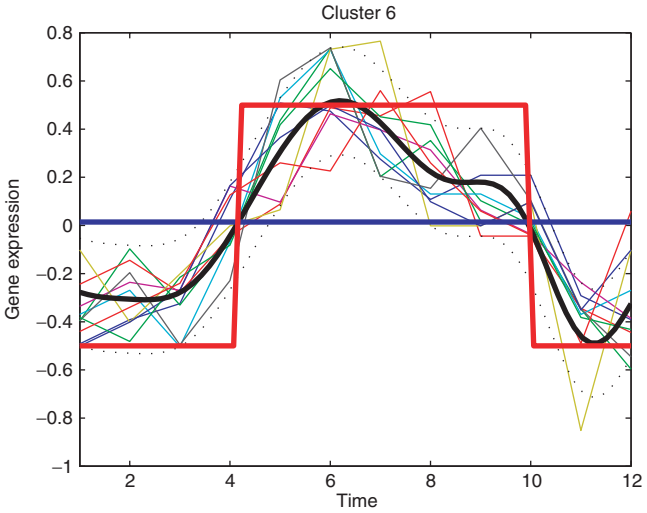

Fig. 3. Smoothing and discretization of expression profiles in cluster 6 (Bcl3) using SVR. The solid black curve was obtained using SVR, the dotted curves show the ε-tube, the blue line gives the mean value of the SVR and the red curve gives the final discretization.

Table 2. Twelve unique discretized profiles from the 23 meta-genes in Figure 2 (dark grey is up-regulated and light grey is down-regulated). Each profile is labeled with a representative gene in the cluster preceeded by E, I or L. E stands for early genes up-regulated after 1 h, I stands for intermediate genes up-regulated after 2 hs and L stands for late genes up-regulated after 8 hs. Further, the time series is divided into two groups: IL-2 starved (before 1 h) and IL-2 stimulated (after 1 h)

	IL-2 starved			IL-2 stimulated							
	Time (h)			Time (h)							
Name	0	0.25	0.5	1	2	4	6	8	10	12	16	24
L-Mybl2												
L-Mcmd												
I-Rpo1-Hnr												
I-Bcl3												
I-Myc												
L-Foxm1												
L-Nsbp1												
E-Cdkn2c												
E-Stat5b												
E-Stat1-6												
E-Stat5a												
E-Jun-Fos												

Table 3. Steady-state dynamics for 0.6% of the inferred networks (dark grey up-regulated, light grey down-regulated). For the IL-2-stimulated condition, the steady-state dynamic follows a three-step cycle (t1, t2, t3)

	IL-2 stimulated									Attractor			
	Time (h)									Time			
Name	1	2	4	6	8	10	12	16	24	t1	t2	t3	t1
L-Mybl2													
L-Mcmd													
I-Rpo1-Hnr													
I-Bcl3													
I-Myc													
L-Foxm1													
L-Nsbp1													
E-Cdkn2c													
E-Stat5b													
E-Stat1-6													
E-Stat5a													
E-Jun-Fos													

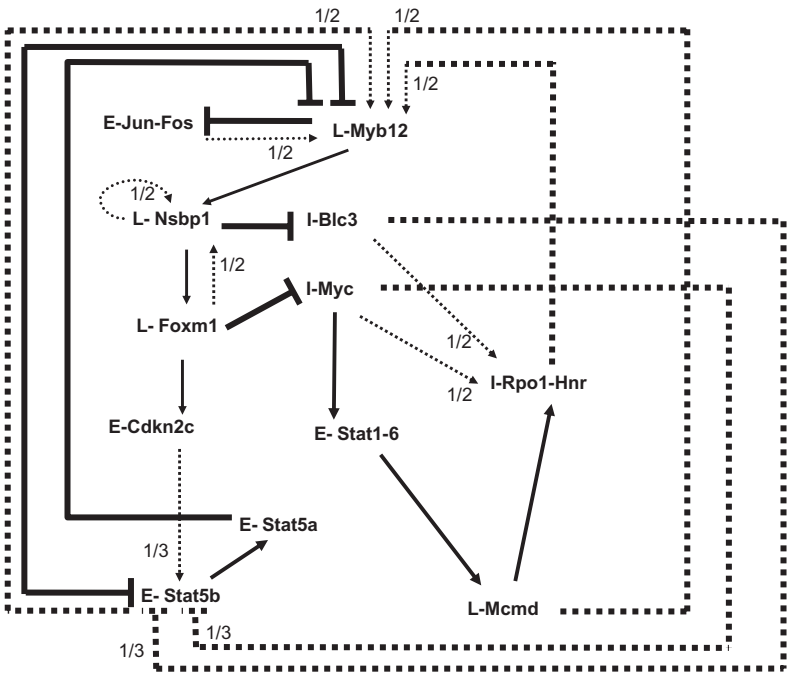

Fig. 4. Activation and inhibition relationships between the 12 meta-genes in Table 2. Solid arrows indicate relationships occurring in all of the 901 networks, while the numbers associated with the dashed arrows indicate the fraction of networks having that relationship.

inferred from the IL-2 time series data and depicted in Figure 4 reveal that (in general) early genes (E) activate other early genes and late genes (L); intermediate genes (I) activate late genes and inhibit early genes; early genes inhibited by intermediate genes can be up-regulated when the intermediate genes are down-regulated; late genes activate other late genes and inhibit early and intermediate genes; and early and intermediate genes inhibited by late genes can be up-regulated when late genes are down-regulated.

3.2 LPS-stimulated macrophage response

The LPS-stimulated macrophage dataset consisted of 15 142 genes measured over six time steps. Using replicates we obtained 90 852 virtual genes. Removing virtual genes with missing values and <0.05 SD in expression resulted in 60 831 virtual genes, corresponding to 14 779 actual genes, or 93.3% of the original 15 142 genes. Using the 60 831 virtual genes, we obtained 23 meta-genes (again by the internal consistency measure). These 23 meta-genes were smoothed and discretized to obtain 15 unique discrete expression profiles given in Table 4. The virtual genes were mapped back to the actual genes before they were assigned to the 15 profiles as described in Section 2.1.

We verified the contents of the discretized clusters against current knowledge of toll-like receptor signaling networks (see for example, the toll-like receptor Kegg maps at http://www.genome.jp). In the RANTES cluster, for example, we found cytokines such as I-TAC, MIPS-1β, IP-10 and IL-1B. All of these are induced by NF-κB and should therefore be co-expressed. Another example is the TNFα cluster, including genes such as A20 and IKBα. These genes are transcribed by NF-κB, but (unlike the previous cytokines) are negative regulators that shutdown NF-κB activity. As a final example, we found that genes such as P38, JNK, IKKε and NIK cluster together. These kinases regulate phosphorylation and ultimately activity in the transcription factors NF-κB and AP-1.

Using `COUNT_NETWORK` we identified a total of 311 039 826 possible networks matching the 15 expression profiles. From these possible networks, we sampled 100 000 networks and computed their steady-state dynamics using the `RUN_NETWORK` and `ATTRACTOR` routines. These 100 000 networks produced only 16 steady-state dynamics, all of which were fixed points. These attractors are shown in Table 4.

In addition to the fact that there were only 16 attractors from 100 000 networks, it is also interesting to note that most of these attractors were very similar. In particular, the 16 attractors discovered were all fixed points, meaning that the genes in the sampled networks did not fluctuate at steady state. This may indicate cell death, which would be consistent with previous knowledge that LPS triggers an innate immunity response through TLR4 (Beutler, 2004) eventually leading to apoptosis in macrophage cells (Xaus *et al.*, 2000).

To corroborate these findings, we searched the microarray dataset for probes corresponding to activators of apoptosis (GO ID = 43 065) and probes corresponding to inhibitors of apoptosis (GO ID = 43 066). Out of 4188 annotated probes, we found 25 positive regulators and 14 negative regulators. For the 15 discrete expression profiles, the average number of up-regulated genes was 48 with 5% SD. This average percentage can be contrasted with the average percentages for the 16 fixed-point attractors. For the attractors, an average

Table 4. The original times series (left) and 16 attractors (right) of the 15 discrete expression profiles (labeled with representative genes) for 100 000 networks inferred from the LPS dataset (dark gret—up-regulated, light grey—down-regulated)

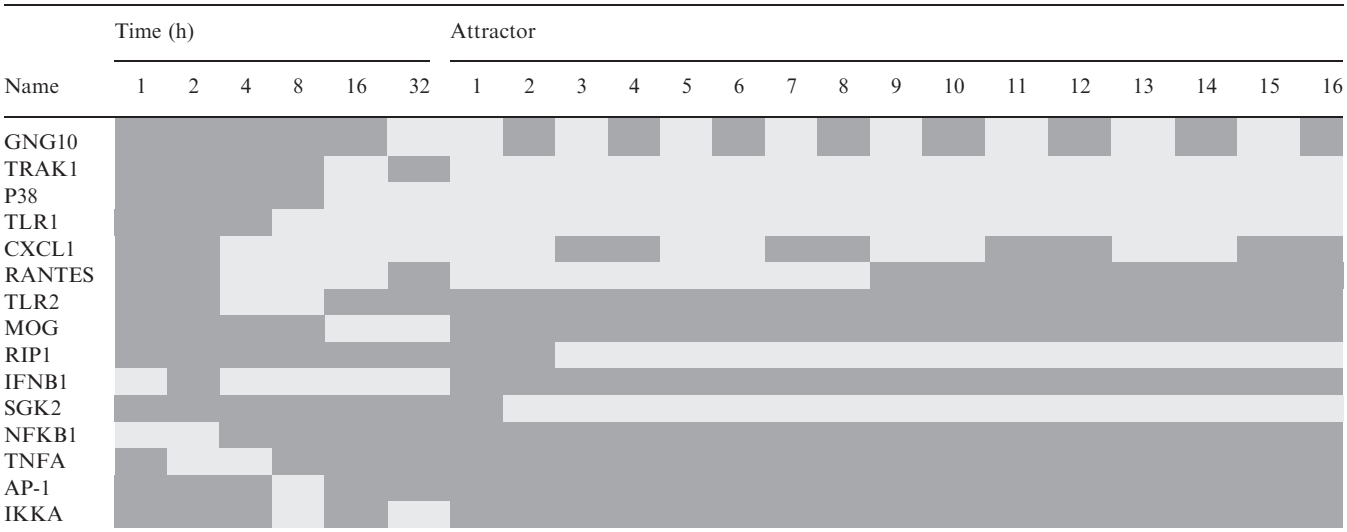

Table 5. Attractor distribution among 100 000 sampled networks

Attractor	# Networks	% Activators	% Inhibitors	Steady state	Fixed point
1	15 000	48	43	0.230	0.030
2	5000	52	50	0.266	0.035
3	3750	64	43	0.042	0.006
4	1250	68	50	0.029	0.004
5	15 000	48	36	0.123	0.016
6	5000	52	43	0.175	0.023
7	3750	64	36	0.023	0.003
8	1250	68	43	0.020	0.003
9	15 000	48	43	0.230	0.030
10	5000	52	50	0.266	0.035
11	3750	64	43	0.042	0.006
12	1250	68	50	0.029	0.004
13	15 000	48	36	0.123	0.016
14	5000	52	43	0.175	0.023
15	3750	64	36	0.023	0.003
16	1250	68	43	0.020	0.003

For each attractor, the percentage of up-regulated apoptosis activators and inhibitors is shown. Columns 5 and 6 give the probability that an attractor would occur in a set of randomly generated networks.

of 58% apoptosis activators was up-regulated, while only 42% of apoptosis inhibitors were up-regulated. In certain cases, these contrasts were even more pronounced for particular steady states (64 and 36% for attractors 7 and 15). The full set of percentages can be found in Table 5.

To further quantify these results, we inferred a set of 100 000 networks using random expression profiles for the 15 meta-genes in Table 4. These 100 000 networks produced 77 583 steady-state dynamics, 10 114 of which were fixed points. Using the random networks, we computed for each of our 16 microarray attractors the fraction of random networks (with steady-state dynamics) having more or less than the percentage of apoptosis activators or inhibitors found using the microarray data. These values are recorded in the fifth column of Table 5 and are empirical probabilities that a given pair of percentages could be obtained at random. For instance, the probability that attractor 16 would have >68% apoptosis activators and <43% apoptosis inhibitors is 0.02. If we further restrict ourselves to fixed-point steady states, we obtain the fractions recorded in column 6 of Table 5. Thus the probability that attractor 16 would occur by random chance with a fixed-point steady state having >68% apoptosis activators and <43% apoptosis inhibitors is 0.003. In general, the numbers in column 6 of Table 5 indicate that our results are significant.

4 DISCUSSION

There are typically many genetic regulatory networks that will match a given time series dataset. Despite this fact, current algorithms available for the inference of regulatory networks produce only a single network. Depending on the method this network might be chosen to be the most probable (Bayesian) or have the lowest dimensional hidden representation (matrix decomposition). In this article we consider all possible networks matching a given time series dataset and group the networks according to dynamics. This approach is appealing due to the fact that we are considering networks which may be missed using other approaches and that biological systems are thought to be robust to variation (Wagner, 2005) so that different networks with similar dynamics may very well be biologically equivalent.

We first considered an IL-2-stimulated immune response dataset. Using this dataset we discovered that

dynamics could be used to eliminate 99.4% of 161 558 possible matching networks. We then produced a composite network using the remaining 0.6% of possible networks. This network confirmed known biological results, namely the identification of early-, intermediate- and late-responding genes to IL-2 stimulation.

Next, in the case of a LPS-stimulated macrophage response dataset we reduced 100 000 sampled networks to 16 fixed point dynamics. These dynamics identified up- and down-regulated apoptosis activators and inhibitors which again agreed with known results for the TLR4 apoptosis pathway.

However, our direct use of dynamics raises additional questions that have not been considered in previous algorithms. These questions have been investigated abstractly in recent work, and should be taken into account in a practical setting such as ours. First, there is the issue of attractor scaling with network size. It was originally thought (Kauffman, 1993) that the number of attractors scaled with the square root of the number of nodes in a network. Recent studies (Bilke and Sjunnesson, 2001) and theoretical work (Samuelsson and Troein, 2003) have shown this to be untrue. In fact, the number of attractors scales superpolynomially with network size. Second, there is the issue of computational artifact. In particular, Boolean networks are typically modeled by simultaneous (synchronous) update of all nodes at each time step. Such networks reduce both theoretical and practical complications. However, it has been discovered that many attractors in synchronous Boolean networks disappear when using asynchronous updates (Bagley and Glass, 1996). To further complicate these issues, stable attractors may be immune to both of these problems (Klemm and Bornholdt, 2005).

Our approach deals with the first issue (attractor scaling) by limiting the number of nodes by using clusters and limiting the network type by using activiation–inhibition functions only. We also limit the number of attractors due to fixed initial conditions. The second issue (attractor artifact) is much more difficult to accommodate, since it implies that attractors found by our method may be artifacts of computation with no biological relevance. To address this issue, we compared the results of our method with experimentally confirmed and/or suspected behavior. A possible computational solution for future study would be the use of some asynchronous update and/or stability criterion, as suggested by the work in Klemm and Bornholdt (2005).

We have shown that the use of dynamics can be an interesting approach for analyzing different networks matching expression profiles for a time series gene expression dataset. Dynamics can be useful when trying to understand the overall behavior of a system and the consequences of this behavior on possible pathways. Dynamics can be particularly useful for isolating networks of interest that relate to a particular behavior under investigation.

ACKNOWLEDGEMENTS

This work was funded by Sandia Laboratory Directed Research and Development. Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Conflict of Interest: none declared.

REFERENCES

Akutsu,T. *et al.* (2000) Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, **16**, 727–734.

Alter,O. and Golub,G.H. (2005) Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations. *Proc. Natl. Acad. Sci. USA*, **102**, 17559–17564.

Bagley,R.J. and Glass,L. (1996) Counting and classifying attractors in high dimensional dynamical systems. *J. Theor. Biol.*, **183**, 269–284.

Basso,K. *et al.* (2005) Reverse engineering of regulatory networks in human B cells. *Nat. Genet.*, **37**, 382–390.

Beutler,B. (2004) Inferences, questions and possibilities in Toll-like receptor signalling. *Nature*, **430**, 257–263.

Bilke,S. and Sjunnesson,F. (2001) Stability of the Kauffman model. *Phys. Rev. E.*, **65**, 016129-1–016129-5.

Brun,M. *et al.* (2005) Steady-state probabilities for attractors in probabilistic Boolean networks. *Signal Processing*, **85**, 1993–2013.

Ching,W.-K. *et al.* (2005) On construction of stochastic genetic networks based on gene expression sequences. *Inter. J. Neur. Sys.*, **15**, 297–310.

de Jong,H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, **9**, 67–103.

Eisen,M.B. *et al.* (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, **95**, 14863–14868.

Friedman,N. and Koller,D. (2003) Being Bayesian about network structure: a Bayesian approach to structure discovery in Bayesian networks. *Mach. Learn.*, **50**, 95–126.

Friedman,N. *et al.* (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Getz,G. *et al.* (2000) Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, **97**, 12079–12084.

Goss,P.J. and Peccoud,J. (1998) Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proc. Natl. Acad. Sci. USA*, **95**, 6750–6755.

Huang,S. (1999) Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *J. Mol. Med.*, **77**, 469–480.

Huang,S. and Ingber,D.E. (2000) Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks. *Exp. Cell. Res.*, **261**, 91–103.

Kauffman,S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.

Kauffman,S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York.

Klemm,K. and Bornholdt,S. (2005) Stable and unstable attractors in Boolean networks. *Phys. Rev. E*, **72**, 055101.

Lahdesmaki,H. *et al.* (2006) Relationships between probabalistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Processing*, **86**, 814–834.

Liang,S. *et al.* (1998) REVEAL: a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific Symposium on Biocomputing (PSB'98)*. World Scientific Publishing, Singapore.

Liao,J.C. *et al.* (2003) Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Natl. Acad. Sci. USA*, **100**, 15522–15527.

Murphy,K. and Mian,S. (1999) *Modelling Gene Expression Data Using Dynamic Bayesian Networks*. University of California, Berkeley, CA.

Nelson,B.H. and Willerford,D.M. (1998) Biology of the interleukin-2 receptor. *Adv. Immunol.*, **70**, 1–81.

Samuelsson,B. and Troein,C. (2003) Superpolynomial growth in the number of attractors in Kauffman networks. *Phys. Rev. Lett.*, **90**, 098701-1–098701-4.

Segal,E. *et al.* (2005) Learning module networks. *J. Mach. Learn. Res.*, **6**, 557–588.

Shmulevich,I. *et al.* (2002a) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18**, 261–274.

Shmulevich,I. *et al*. (2002b) From Boolean to probabalistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE.*, **90**, 1778–1792.

Shmulevich,I. *et al*. (2003) Steady-state analysis of genetic regulatory networks modeled by probabilistic Boolean networks. *Comp. Funct. Genomics*, **4**, 601–608.

Smola,A.J. and Scholkpof (1998) *A Tutorial on Support Vector Regression*. Holloway College, University of London, London, UK.

Smolen,P. *et al*. (2000) Mathematical modeling of gene networks. *Neuron*, **26**, 567–580.

Tamayo,P. *et al*. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, **96**, 2907–2912.

Trefethen,L. and Bau,D. (1997) *Numerical Linear Algebra*. SIAM, Philadelpha, PA.

Wagner,A. (2005) *Robustness and Evolvability in Living Systems*. Princeton University Press, Princeton, NJ.

Xaus,J. *et al*. (2000) LPS induces apoptosis in macrophages mostly through the autocrine production of TNF-alpha. *Blood*, **95**, 3823–3831.

Zhang,Y. *et al*. (2005) A time-series biclustering algorithm for revealing co-regulated genes. In *Proc. Int. Conf. Inf. Tech. Coding and Comp. (ITCC). IEEE Computer Society*.

Zou,M. and Conzen,S.D. (2005) A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, **21**, 71–79.