BIOINFORMATICS

Vol. 18 Suppl. 1 2002
Pages S216–S224

Evaluating functional network inference using simulations of complex biological systems

V. Anne Smith[1], Erich D. Jarvis[1] and Alexander J. Hartemink[2]

[1]Department of Neurobiology, Duke University Medical Center, Box 3209, Durham, NC 27710, USA and [2]Department of Computer Science, Duke University, Box 90129, Durham, NC 27708, USA

Received on January 24, 2002; revised and accepted on April 1, 2002

ABSTRACT

Motivation: Although many network inference algorithms have been presented in the bioinformatics literature, no suitable approach has been formulated for evaluating their effectiveness at recovering models of complex biological systems from limited data. To overcome this limitation, we propose an approach to evaluate network inference algorithms according to their ability to recover a complex functional network from biologically reasonable simulated data.

Results: We designed a simulator to generate data representing a complex biological system at multiple levels of organization: behaviour, neural anatomy, brain electrophysiology, and gene expression of songbirds. About 90% of the simulated variables are unregulated by other variables in the system and are included simply as distracters. We sampled the simulated data at intervals as one would sample from a biological system in practice, and then used the sampled data to evaluate the effectiveness of an algorithm we developed for functional network inference. We found that our algorithm is highly effective at recovering the functional network structure of the simulated system—including the irrelevance of unregulated variables—from sampled data alone. To assess the reproducibility of these results, we tested our inference algorithm on 50 separately simulated sets of data and it consistently recovered almost perfectly the complex functional network structure underlying the simulated data. To our knowledge, this is the first approach for evaluating the effectiveness of functional network inference algorithms at recovering models from limited data. Our simulation approach also enables researchers *a priori* to design experiments and data-collection protocols that are amenable to functional network inference.

Availability: Source code and simulated data are available upon request.

Contact: amink@cs.duke.edu; asmith@neuro.duke.edu; jarvis@neuro.duke.edu

Keywords: Bayesian network; network inference algorithm; model induction; model inference; automatic discovery; simulation; complex system; systems biology; evaluation framework; songbird; brain; gene expression; electrophysiology; molecular neurobiology; functional genomics.

INTRODUCTION

One major goal of functional genomics research is to take large sets of biological data, usually correlational, and elucidate functional interactions between elements in a causal pathway or network. Such efforts have led to the recent development and use of linear (D'haeseleer *et al.*, 1999), nonlinear (Weaver *et al.*, 1999), target-regulator pair (Arkin *et al.*, 1997), Boolean (Liang *et al.*, 1998; Akutsu *et al.*, 2000), and Bayesian (Friedman *et al.*, 2000; Hartemink *et al.*, 2001) network inference algorithms to predict biological pathways. Some of the predicted functional interactions are biologically reasonable since they can be corroborated by published findings in the literature; some appear to be unreasonable. However, the validity of the vast majority of predicted interactions is difficult to assess because the interactions have not been biologically tested. To test all of them experimentally would involve multiple gene knockout studies or other types of interventions. This could take decades, if not several lifetimes, to accomplish for even a single complex network, as there are currently no high-throughput experimental resources for such an undertaking. Most daunting are networks that predict hundreds to thousands of biological interactions.

In an attempt to circumvent this problem, we developed a new approach. We begin by creating a computer simulation of a biological system in which we make and know all the rules. We then run the simulation and sample data from it, as one would sample data from a real biological system. Finally, we present the sampled data to algorithms that purport to discover underlying regulatory network structure and evaluate their ability to recover the original simulated network. In this manner, we can test an algorithm's accuracy, modify it when necessary,

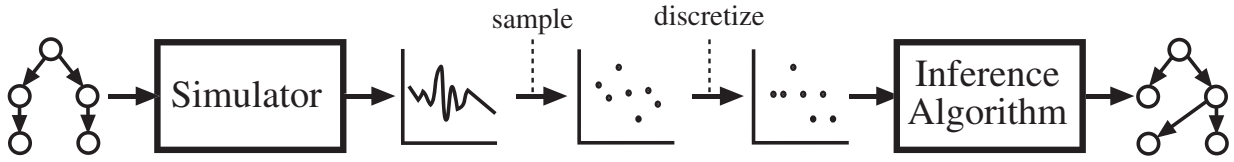

Fig. 1. Using simulation to evaluate functional network inference. Moving from left to right: We design an underlying structure for the system that is simulated by BRAINSIM to produce continuously changing data. We sample from the simulated data as one would in practice, discretize the sampled data, and then apply our network inference algorithm, NETWORKINFERENCE, to recover network structure. Comparing the recovered structure with that provided to the simulator enables us to evaluate both our network inference algorithm and our method of sampling data from the simulated system.

and help inform the design of real biological experiments for use with the algorithm (Figure 1). Although such an approach is not a complete substitute for intervention experiments, it has the potential to save years of time for functional genomics research, in part by helping to design experiments and data-collection methodologies to decipher the underlying mechanisms of biological systems more accurately. This study presents the details of this approach as developed for an integrative biological project (Jarvis *et al.*, 2002).

Our simulator, BRAINSIM, is modelled after the vocal communication system of the songbird brain. Songbirds are used because they have the rare trait of vocal learning, the substrate for human language. This trait has been most studied in songbirds, and thus they represent one of the best animals for modelling complex brain and behaviour functions. In addition, the songbird vocal learning system is built upon the basic features of sensory processing (hearing), motor processing (vocalizing), and sensorimotor integration of the two. It has been studied at the molecular, neural anatomical, electrophysiological, and behavioural levels. Thus, discoveries with the songbird vocal system and our simulation are expected to be applicable to a wide variety of systems.

Our functional network inference algorithm, called NETWORKINFERENCE, uses Bayesian networks to model complex biological systems like those governing vocal learning in the songbird brain. The advantage of Bayesian networks is that they capture conditional dependence and independence between variables, and are capable of incorporating information from multiple levels of analysis. They also can model stochastic processes, which are known to play a role in gene expression (McAdams and Arkin, 1997). An additional feature of our NETWORKINFERENCE algorithm is that it can exploit time series data to discover dynamic networks that capture cyclic phenomena occurring in complex biological systems, like feedback.

APPROACH

BRAINSIM design

BRAINSIM is written in C++ and models an animal's externally observable behaviour, electrophysiological activity in distinct regions of the brain, and gene expression in each of these regions of the brain (Figure 2). In the simulation we use in this paper, a *bird* exhibits a *behaviour*, modelled to have two possible states, 0 or 1, which are arbitrary for the purposes of the simulation, but which we took to correspond to silence and singing. Each bird also has five *regions* that make up its brain. Each region has values for electrophysiological *activity* and the expression levels of 100 *genes*. Electrophysiological activity values range from 0–400 Hz, the range found when measuring multi-unit action potentials with extracellular electrodes (Hessler and Doupe, 1999). Absolute levels of gene expression are modelled with values in the range 0–50 (arbitrary units).

In four of the five brain regions, hereafter called regulated regions, activity was correlated with behaviour. For two of the four regions, activity was low (0–100 Hz) when behaviour was 0 and high (300–400 Hz) when behaviour was 1. For the other two regions, this relationship was reversed. In the fifth region, activity was not correlated with behaviour. This represents the possibility that behaviour may correspond with increased activity in some regions of the brain, suppressed activity in others, and be unrelated to activity in still others (Figure 2, middle). The regulatory network for each region involved only 10 of the 100 genes. Two genes were directly dependent on activity in the region, while the other eight formed a regulatory network downstream of these two (Figure 2, bottom). We included 90 extra unregulated genes as distracters to represent the highly likely possibility that only a small subset of the measured genes will be involved in the network of interest.

Starting values for all variables were initialized in BRAINSIM. Behaviour began in state 0. Activity in regulated regions began at a random low or high value (ranges as above) to correspond with behaviour 0, and

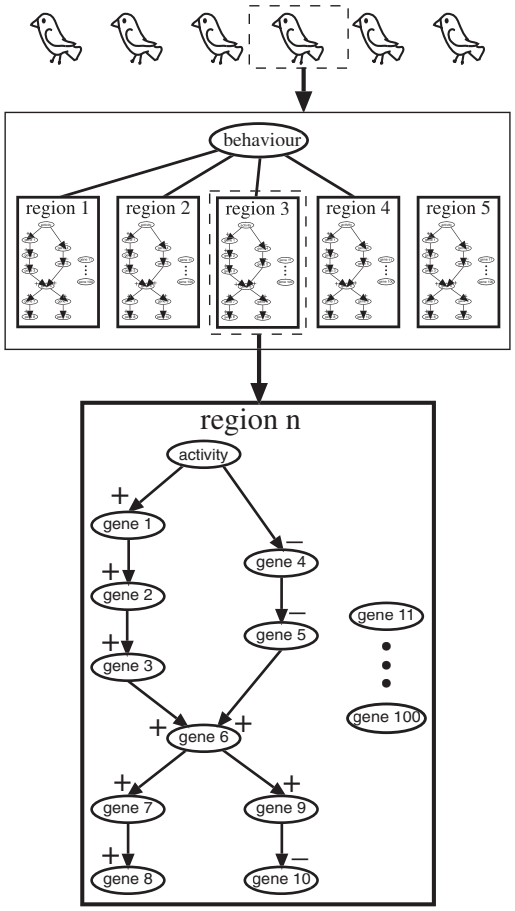

Fig. 2. Overview of BRAINSIM. A single data set produced by BRAINSIM consists of six birds (top), each of which consists of a behaviour and five brain regions (middle). Behaviour corresponds with activity in regions 1 through 4, but not 5. All five brain regions have the same regulatory network (bottom). Arrows from one node to another in the regulatory network represent an influence of the value of the upstream variable at one time step on the value of the downstream variable at the next time step. A plus (+) next to the arrowhead indicates a positive influence; a minus (−), a negative influence.

activity in the unregulated region began at a random value between 0 and 400. The initial level of gene expression for a gene (across all regions) was called its 'target' value, intended to correspond to its constitutive expression level. This value was selected as a random value in the range 0–10 for upregulated genes, 40–50 for downregulated genes, and 0–50 for the 90 unregulated genes. The target values for the 100 genes were saved in a text file to be read for each run of BRAINSIM. Thus, the constitutive expression level of each gene was the same for every bird across all data sets throughout this study.

BRAINSIM then generated values for behaviour, activity, and gene expression at all subsequent time steps, modelled to be approximately 1 minute apart. Values were calculated based on their previous values and on any regulatory influences. Because in a biological system changes in activity and behaviour occur on the order of milliseconds with respect to one another while regulatory influences on gene expression occur on the order of minutes, behaviour and activity were modelled to change simultaneously (within a single time step). For a specified number of time steps, behaviour remained at the same value, and activity in the regulated regions was chosen to be a new random high or low value each time step to correspond with behaviour. When behaviour switched between 0 and 1, activity in regulated regions switched to a random value from the opposite range. Activity in the unregulated region was equal to its value in the previous time step plus or minus a random amount (from 0–100 Hz); this range was chosen empirically to match experimentally observed patterns of activity (Hessler and Doupe, 1999, and our own lab).

Three influences summed to provide values for gene expression from one time step to the next. First, all genes had a returning function, where they added or subtracted a constant amount (chosen to be 4) to move closer to their original target levels. This represents degradation of mRNA in the cell or return to constitutive transcription after suppression. Second, the ten regulated genes adjusted their values based on the levels of their regulators in the previous time step. Upregulated genes added a proportion (0.2) of their regulator's expression level to their own, and downregulated genes subtracted a proportion (0.2). Thus, the higher the expression level of the regulator, the larger its influence. Because activity had a larger range than that of the genes, it was multiplied by the appropriate scaling factor (1/8) before computing its influence on the two genes that were directly dependent on it in the region. The 90 unregulated genes added or subtracted a random amount (from 0–5) to simulate regulation by other unmeasured processes. Third, a random amount (from 0–6) was added to or subtracted from each gene to simulate stochasticity in gene expression. Finally, computed expression levels were truncated to be in the range 0–50. This represents the inability to have negative levels of gene expression, and the likelihood of a maximum transcription rate for each gene. As with the dynamics of activity, the magnitudes of these gene expression influences were chosen empirically to provide biologically reasonable patterns of expression.

NETWORKINFERENCE design

The NETWORKINFERENCE algorithm takes a collection of observed data as input and then searches for Bayesian networks that are good at explaining the observed data without unnecessary complexity, returning the best Bayesian network, or networks, that it encounters during

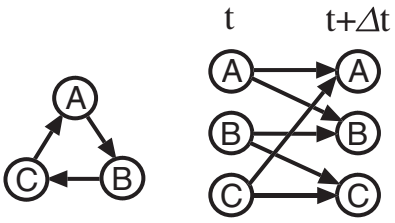

Fig. 3. Representing a regulatory network using a dynamic Bayesian network. On the left is a cyclic regulatory network with three elements. On the right is this same regulatory network but depicted as a dynamic Bayesian network. All connections from the regulatory network are shown from the controlling element in time t to the controlled element in time $t + \Delta t$.

its search. A Bayesian network is a model that captures probabilistic relationships among variables in the form of a graph; nodes in the graph correspond to observed variables and directed edges (links) in the graph mean that the child node is conditionally dependent on the parent node. Unlike a number of other modelling frameworks, Bayesian networks permit stochastic, combinatorial, and nonlinear relationships among variables, enabling them to represent, at the level of conditional dependence, many phenomena that we observe in complex biological systems. In addition, the probabilistic nature of Bayesian networks enables them to handle noisy data gracefully. Although static Bayesian networks are limited to acyclic directed graphs, dynamic Bayesian networks can be used to model cyclic behaviours that emerge temporally, like feedback (Figure 3). We designed NETWORKINFERENCE in a manner that is capable of searching for dynamic Bayesian networks from temporal data.

The scoring metric employed by NETWORKINFERENCE is the widely used Bayesian scoring metric (BSM). Under the BSM, the score of a Bayesian network is defined as the log probability of the network given the observed data. Any principled scoring metric must balance a network's ability to explain observed data with its ability to do so simply: scoring metrics with a penalty for unnecessary complexity are able to guard against the over-fitting of network models to observed data. The BSM includes an inherent penalty for unnecessary complexity and thus higher scores are given to networks with a better ability to explain observed data simply; the score of one network is higher than the score of another if and only if it is better at explaining the observed data simply. Furthermore, the difference between the scores for any two networks leads to a direct statistical significance measure for determining how strongly one should be preferred over the other. In the presence of complete data, the BSM for a discrete Bayesian network can be computed exactly in closed form (Cooper and Herskovits, 1992). Bayesian network methods have previously been applied in the context of genetic regulatory network modelling (Hartemink *et al.*, 2001; Friedman *et al.*, 2000), but, to our knowledge, this is the first application to modelling complex systems at multiple biological levels of organization beyond genetic regulation.

Identifying the highest-scoring network for an arbitrary set of observed data has been shown to be NP-complete under the BSM (Chickering, 1996); as a result, an exhaustive search would be futile and we must instead consider heuristic search strategies. In previous work (Hartemink, 2001; Hartemink *et al.*, 2002), we compared the performance of a number of heuristic search strategies and observed that simulated annealing (Kirkpatrick *et al.*, 1983) consistently found higher scoring networks than other methods. Consequently, our implementation of NETWORKINFERENCE, written in C, searches for high-scoring networks in the space of Bayesian networks by simulated annealing (with extensions for reannealing to avoid becoming trapped in local maxima). Our implementation also allows the user to modify the prior over network structures by specifying sets of links that are required to be present or required to be absent. For example, this mechanism allows NETWORKINFERENCE to search for dynamic Bayesian networks by disallowing links that flow backwards in time.

RESULTS

Data generation, sampling

We used BRAINSIM to generate data for six birds for 200 time steps each, switching between the two behaviour states every 50 time steps. We considered this a single data set. Analysis of BRAINSIM output revealed that activity in regulated regions remained high or low, but varied considerably within those subranges, while activity in the unregulated region wandered randomly throughout the whole range, sometimes making large jumps. Regulatory influences were propagated through all ten regulated genes in the network while the remaining 90 genes moved randomly. Noticeable time lags were observed in response to regulation, and the speed of response to increase of a regulator was not necessarily the same as the response to decrease of a regulator. Panel (a) of Figure 4 shows a graphical depiction of a small, representative sample of such BRAINSIM output. It can be seen that increased activity (top trace) is followed by upregulation of gene 1 (second trace from top) with a slight time lag. When activity drops, gene 1 then returns to hover near its target level, also with a slight time lag. Gene 10 (third trace from top) is at the terminus of the regulatory network and is downregulated by its regulator (whose trace is not shown). Its downregulation occurs considerably later than

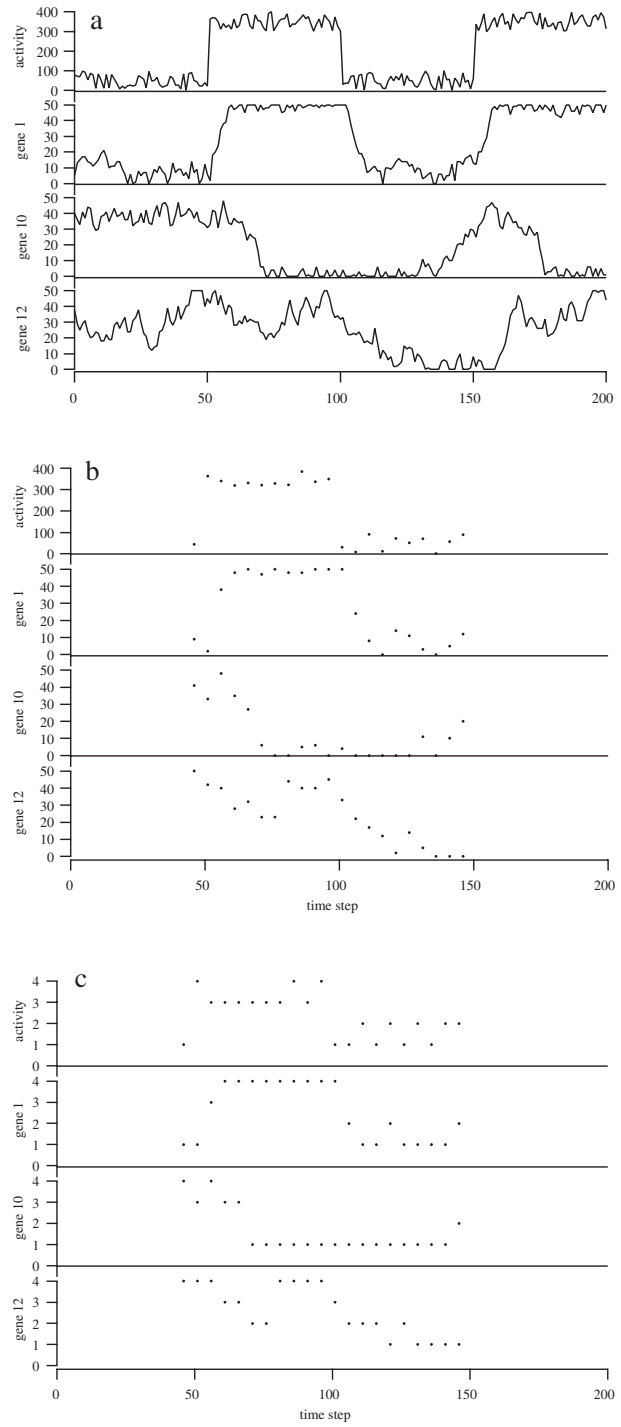

Fig. 4. BRAINSIM output, sampling, and discretization. (**a**) Values for activity and the expression of genes 1, 10, and 12 for one run of BRAINSIM in one region. The region displayed in the Figure is positively correlated with behaviour (behaviour 0 translates to low activity and behaviour 1 to high activity; behaviour graph not shown). (**b**) Data sampled every 5 time steps, beginning at time 45, from the simulation for the values depicted in panel (a). (**c**) Quartile discretized values for the sampled data in panel (b).

the response of gene 1 to activity, showing how regulatory effects are propagated through the network over time. The relaxation of gene 10's downregulation does not even occur until just before behaviour is ready to switch back to state 1. Gene 12 (bottom trace) is one of the 90 unregulated genes in the region and can be seen to wander randomly in a wide range but without respect to activity or the other genes shown. These traces (and others not shown) suggest that BRAINSIM generates biologically reasonable data as output.

To sample data from this system as one would sample data in an actual experiment, we started sampling just before the bird began to sing (just before behaviour switched to 1) and continued every 5 time steps while the bird sang, as the bird stopped singing (as behaviour switched to 0), and while the bird was silent. In all, 21 samples were taken from the 200 time point series, spanning one full cycle of singing and silence (panel (b) of Figure 4). Thus, while BRAINSIM generated data for each time step, we sampled the data at intervals, getting only periodic slices of the system, resulting in considerable loss of information. As is the case in a typical animal experiment, although changes in behaviour, electrophysiological activity, and gene expression occur continuously, a researcher usually is able to sample data only at occasional points and must draw conclusions from these limited data.

We chose to sample every 5 time steps based upon the intuition that the time interval between samples should match the intervals at which the phenomena of interest unfold. For example, approximately 5 minutes are required for mRNA transcripts to be exported from the nucleus, be translated into proteins, and return to the nucleus to have a regulatory effect on the transcription of other genes. Thus, time series of mRNA expression levels are taken at intervals of at least 5 minutes. Similarly, in our simulation, the effect of a regulator on its target was 0.2 in each time step, corresponding to a full effect over the course of 5 time steps.

Data discretization

When using discrete Bayesian networks, the number of possible values for each variable should be as small as reasonably possible for computational reasons. This requires that we discretize the data sampled from BRAINSIM. Discretization can result in information loss if important variation is lost; however, it also has the benefit of making the data more robust by smoothing out uninformative random noise. We chose to use four values for discretization to balance among these pressures. Behaviour already had only two states and was thus left unchanged. The different activity and gene expression values were discretized to four levels using quartile discretization: the lowest 25% of

values were labelled state 1, the next highest 25% were labelled state 2, the next 25% were labelled state 3, and the highest 25% were labelled state 4. The final discretization is shown for representative variables in panel (c) of Figure 4. This sampled and discretized data, several times removed from the output generated by BRAINSIM, are provided as input to NETWORKINFERENCE.

Functional network recovery

To recover the underlying structure of the system, we applied NETWORKINFERENCE in two stages. First, we determined the regulatory network of the genes and activity in all five regions. Second, we examined all five regions in relation to behaviour of the bird, using activity as the best summarizing value for each region. We then constructed the whole system by merging these two structures together at the activity nodes for each region. Our reason for breaking the problem into stages is as follows. Because gene regulation occurs relatively slowly, on the order of our sampling interval, we used NETWORKINFERENCE to recover a dynamic Bayesian network in the first stage, expecting expression levels at one time point to have an influence on levels at the next sampled time point. In contrast, changes in behaviour and activity occur simultaneously, relative to our sampling interval, so we would not expect behaviour at one sampled time point to be directly related to activity at the next sampled time point. Thus, in the second stage, we used NETWORKINFERENCE to recover a static Bayesian model representing dependence between variables at an instant in time.

To test the robustness of NETWORKINFERENCE in both stages, we applied it to 50 separate data sets generated from the same structure, one at a time. By examining the reproducibility of the results for independent inference tasks on the same underlying structure, we can provide approximate measures of confidence for results obtained when examining a single data set collected from an actual biological experiment.

First stage: brain region regulatory network recovery For the first stage, we considered each brain region to be a sample of the regulatory network under different conditions. To recover a dynamic Bayesian network, we created nodes for activity and the expression of all 100 genes in both time t and time $t + \Delta t$. Our 21 samples of data were therefore converted to 20 paired samples of data (for example, activity at time t paired with activity at time $t + \Delta t$). Since we simulated six birds with five brain regions each, this resulted in a data matrix of 600 observations for 202 variables. We mandated links from a variable at time t to itself at time $t + \Delta t$. To enforce recovery of a valid dynamic Bayesian network, we disallowed links within the same time period and links from a node in time $t + \Delta t$ backwards to a node in time t.

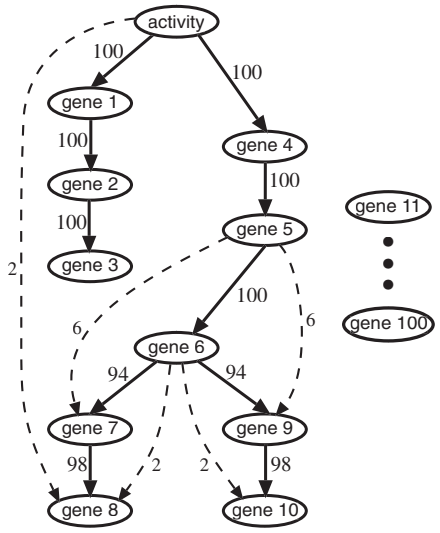

Fig. 5. Brain region regulatory network recovery. Links from the recovered structures of all 50 runs of NETWORKINFERENCE. The percentage of structures that had a particular link is noted next to that link. Links in 94% or more of the recovered structures are solid; links in 6% or fewer are dashed; no links appeared with an intermediate frequency. Note that this figure does not represent a Bayesian network. Rather, it is a projection of the dynamic Bayesian network structures produced by NETWORKINFERENCE onto a graphical representation of BRAINSIM's regulatory network (such a projection is depicted for a simple cycle in Figure 3.

The first stage was run for each data set for 1 hour on a 1 GHz Pentium III CPU running Linux. NETWORKINFERENCE searched an average of 8.5 million structures (range 8.1–8.7 million) and found its best scoring structure within the first 1.7 million structures on average (range 0.5–8.4 million). All 50 runs of NETWORKINFERENCE on separate data sets were highly effective at recovering the brain region regulatory network structure of the simulated system (Figure 5). They all correctly identified activity and the 10 regulated genes as being involved in a network, and correctly identified the remaining 90 genes as being unrelated to anything else in the data set. The ability to classify variables as being unrelated derives from the inherent penalty for complexity in the Bayesian scoring metric. Of the 50 recovered structures, 43 were identical, recovering 10 of the 11 links in the regulatory network; the remaining 7 structures had one or two incorrect links, but always between elements correctly belonging to the regulatory network. This resulted in a mean($\pm$SE) recovery of $89 \pm 0.1\%$ for the 11 correct links, with $98 \pm 0.1\%$ of the links in the recovered structures corresponding to a link in the original regulatory network.

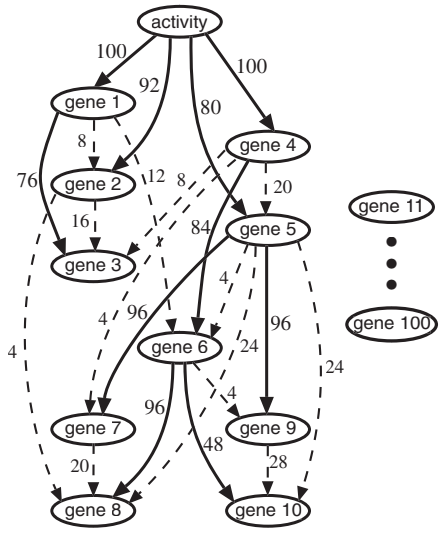

Fig. 6. Brain region regulatory network recovery with larger sampling interval. Links from the recovered structures from NETWORKINFERENCE on 25 separate data sets sampled every 10 time steps are shown. The percentage of structures that had a particular link is noted next to that link. Links in 48–100% of the recovered structures are solid; links in 4–28% of the recovered structures are dashed. An explanation for the link-jumping is presented in the text.

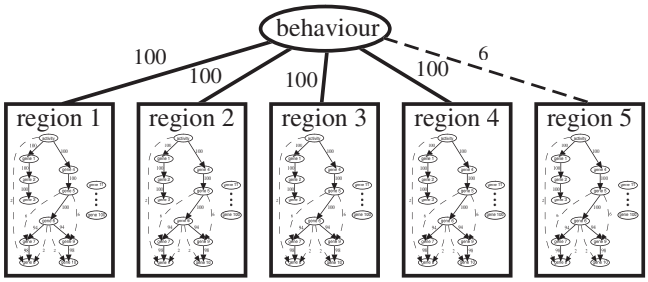

Fig. 7. Brain-behaviour network recovery. Links from proposed solutions of all 50 runs of NETWORKINFERENCE. Percentage of solutions with each link is noted next to that link. A full picture of the system can be obtained by merging the results of the two recovery stages together at the activity nodes for each region. Thus, the regulatory network, as recovered in Figure 5, is represented in each region here.

One link was consistently not recovered: from gene 3 to gene 6. In the simulation, gene 3 and gene 5 controlled gene 6 in a coordinated fashion, with the lower expression level of the pair serving as the limiting factor in the regulation of gene 5. In examining the simulation output across the data sampling period, we found that gene 5 had a lower expression level than gene 3 in $89 \pm 0.2\%$ of the cases, while gene 3 was lower than gene 5 in only $9 \pm 0.2\%$ of the cases. Thus gene 5 nearly always served as the effective regulator. Consequently, NETWORKINFERENCE identified a dependence between gene 5 and gene 6, but not between gene 3 and gene 6, because such a link did not explain a sufficient quantity of the data to overcome the BSM's inherent penalty for complexity. We verified that this was the case by calculating that the score for the 'correct' BRAINSIM network was lower (worse) than the best scoring structure found by NETWORKINFERENCE for all 50 data sets. Thus, the functional network inference algorithm was not flawed but rather, the data did not fully 'exercise' the links in the underlying structure. This reveals that when one variable is frequently the limiting regulator, other 'unexercised' regulators may not be identified.

To assess our intuitive choice of matching the sampling interval to the dynamics of the system, we next took samples every 10 time steps, a time interval twice as long as the one we believed matched the system, and compared NETWORKINFERENCE's ability to recover the functional network. To keep the number of data points consistent, we took the same number of data points across the same transitions; this required the simulator to run twice as long. We repeated the inference task for 25 different data sets, and although the correct nodes were again completely recovered in each case, we found only a mean recovery of $27 \pm 0.3\%$ of the 11 correct links, with $30 \pm 0.4\%$ of links in the recovered structures corresponding to a correct link in the regulatory network (Figure 6). An additional $62 \pm 0.4\%$ of the links in the recovered structures were off by one node; that is, they connected two genes in the network that should have been separated by a single node (for example, connecting gene 6 to gene 8, when the correct network connects gene 6 to gene 7 and then to gene 8). Thus, the larger sampling interval often skipped over connections, confirming that the sampling interval can have a dramatic effect on the ability to recover a network. One must sample at intervals that are small enough to capture the dynamic phenomena of interest in the system.

Second stage: brain-behaviour network recovery For the second stage, we presented NETWORKINFERENCE with data representing bird behaviour and discretized activity in the five regions for each of the 21 sampled points. For the six birds, this produced a data matrix of 126 observations for 6 variables, a relatively trivial problem. However, when more than two behaviours are observed and when these behaviours may depend combinatorially on activity in different brain regions which themselves may interact with one another, this stage could become as difficult as the first one (if not more so). The second stage was run for each data set for about 10 minutes on a 1 GHz Pentium III CPU running Linux. NETWORKINFERENCE

searched an average of 8.0 million structures (range 4.1–8.9 million), finding its best scoring structure within the the first 3.3 million on average (range 50 000–8.7 million).

Of the 50 runs of NETWORKINFERENCE on different data sets, 47 correctly identified only the activities of the four regulated regions and behaviour as related; the other 3 added an extra link between behaviour and the activity of the unregulated region. All 50 had links between only activity in the regions and behaviour, and none among the five activities. Thus, there was 100% recovery of the 4 correct links (between behaviour and activity in the 4 regulated regions), and a mean of 99 $\pm$ 0.7% of links in the recovered structures corresponding to correct links (Figure 7). For the 3 runs whose solutions had links to the fifth unregulated area, the 'correct' structure scored lower (worse) than the best scoring structure found by NETWORKINFERENCE. Thus, these sampled data sets likely had spurious correlations of behaviour with the randomly wandering activity that the algorithm correctly identified as existing.

DISCUSSION

In this report, we show that it is possible to evaluate and thus develop functional inference network algorithms for elucidating complex biological systems in reasonable time by using realistically simulated data. This approach requires two main phases: development of a realistic simulation and development and testing of an inference algorithm. The fact that we simulated the underlying system, and not just the data points provided to the algorithm, meant that we could also examine various methods for sampling the system to provide input to the algorithm. Such a simulation and network inference testing approach differs from previous methods in testing functional network algorithm effectiveness for a number of reasons. First, we generate biologically reasonable data rather than data from a Bayesian network. This means there is a mismatch between the system generating the data and the framework used to model the system, which is far more realistic. Second, our data has multiple distracters and is forced to cope with significant information loss, as might be expected in an actual experiment. Third, the algorithm is capable of handling the temporal data produced by the simulator using dynamic Bayesian networks, enabling modelling of biological feedback loops that are difficult to represent with standard (static) Bayesian networks. A dynamic representation can also help reveal possible causal relationships. Fourth, we infer networks involving multiple levels of biological complexity and not just networks limited to gene regulatory pathways.

It was remarkable that NETWORKINFERENCE displayed both high recovery fidelity and high robustness on the simulated data, even in the presence of over 90 noisy distracter variables, like unrelated genes and activity. This has been a worrisome aspect of high-throughput bioinformatics research because gene expression arrays and electrophysiological experiments produce noisy data and often measure many variables that are unrelated to the biological systems of interest. It appears that inference algorithms may be able to sort out the noise and distracters from the specific regulatory network of interest. The inclusion of noisy unrelated genes is especially useful in testing these methods because when conducting our experiments we will not know *a priori* which are the genes whose levels are being regulated. Similarly, at the level of brain-region organization, one of the regions passed to the algorithm generated data for the same 100 genes as the other regions but its electrophysiological activity level was independent of the bird's behaviour. An additional interesting find was that although the simulator generated continuous data, when those data were sampled, discretized, and passed to the inference algorithm for learning, NETWORKINFERENCE was forced to cope with information loss. The inference algorithm did not have access to all the output of the simulation, but instead only occasional time points like those that can be reasonably gathered in experiments today. The data were then discretized, producing even greater loss of information. Yet, NETWORKINFERENCE provided high recovery success on these limited data, demonstrating that functional network inference algorithms hold promise for revealing mechanisms underlying complex biological systems.

A benefit of our simulation and recovery testing approach is that it helps us understand how we should collect data before performing a biological experiment. This is likely one of the most powerful features of our approach. For example, tests confirmed our intuition that we need to sample data on the same time scale as that on which functional responses occur. By examining the effect of sampling at non-uniform intervals and varying the number of subjects tested, we will be able to design biological experiments that produce the most pertinent data for our network inference method. We expect to collect gene expression and electrophysiological data from brain regions with known anatomical connectivity, from animals in different behavioural states, such as silence, hearing, singing alone, and singing to a companion (Jarvis *et al.*, 2002).

There are still a great number of limitations of our approach. First, BRAINSIM models the songbird brain at a very simple level. Future work will be necessary to develop the simulator such that it includes more complex gene regulatory networks, anatomical connectivity, refined electrophysiological activity, *etc.* Second, we tested only one type of inference algorithm here, and one that needs discretized data. Other algorithms may be more effective

when the system becomes more complex. Third, final verification of our approach will require at least a minimal set of biological intervention experiments. Nevertheless, our approach appears to hold promise for both validating and developing inference algorithms and for use and development of advances in experimental design.

ACKNOWLEDGEMENTS

This work is supported by NIH training grant T32-NS07370-08 to V.A.S. and Duke Provost Bioinformatic grant to E.D.J. A.J.H. thanks Tomi Silander for sharing the source code to his B-Course software.

REFERENCES

Akutsu,T., Miyano,S. and Kuhara,S. (2000) Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. In *RECOMB*, Vol. 4, ACM Press, pp. 8–14.

Arkin,A., Shen,P. and Ross,J. (1997) A test case of correlation metric construction of a reaction pathway from measurements. *Science*, **277**, 1275–1279.

Chickering,D.M. (1996) Learning Bayesian networks is NP-complete. *Learning from Data: AI and Statistics V*, Chapter 12, Fisher,D. and Lenz,H.-J. (eds), Springer, pp. 121–130.

Cooper,G.F. and Herskovits,E. (1992) A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, **9**, 309–347.

D'haeseleer,P., Wen,X., Fuhrman,S. and Somogyi,R. (1999) Linear modeling of mRNA expression levels during CNS development and injury. In *Pac. Symp. Biocomput.*, Vol. 4, World Scientific, pp. 41–52.

Friedman,N., Linial,M., Nachman,I. and Pe'er,D. (2000) Using Bayesian networks to analyze expression data. In *RECOMB*, Vol. 4, ACM Press, pp. 127–135.

Hartemink,A.J. (2001) *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*, PhD thesis, MIT.

Hartemink,A.J., Gifford,D.K., Jaakkola,T.S. and Young,R.A. (2001) Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Pac. Symp. Biocomput.*, Vol. 6, World Scientific, pp. 422–433.

Hartemink,A.J., Gifford,D.K., Jaakkola,T.S. and Young,R.A. (2002) Combining location and expression data for principled discovery of genetic regulatory network models. In *Pac. Symp. Biocomput.*, Vol. 7, World Scientific, pp. 437–449.

Hessler,N.A. and Doupe,A.J. (1999) Singing-related neural activity in a dorsal forebrain-basal ganglia circuit of adult zebra finches. *Neuroscience*, **19**, 10461–10481.

Jarvis,E.D., Smith,V.A., Rivas,M.V., Wada,K., McElroy,M., Smulders,T.V., Carninci,P., Hayashisaki,Y., Dietrich,F., Wu,X., McConnell,P., Hartemink,P., Wang,A. and Lin,S. (2002) Integrating the songbird brain. *J. Comp. Physiol.,* submitted.

Kirkpatrick,S., Gelatt,C.D. and Vecchi,M.P. (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.

Liang,S., Fuhrman,S. and Somogyi,R. (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Pac. Symp. Biocomput.*, Vol. 3, World Scientific, pp. 18–29.

McAdams,H.H. and Arkin,A. (1997) Stochastic mechanisms in gene expression. *Proc. Natl Acad. Sci. USA*, **94**, 814–819.

Weaver,D.C., Workman,C.T. and Stormo,G.D. (1999) Modeling regulatory networks with weight matrices. In *Pac. Symp. Biocomput.*, Vol. 4, World Scientific, pp. 112–123.