Journal of Bioinformatics and Computational Biology
Vol. 2, No. 4 (2004) 765–783

Imperial College Press
www.icpress.co.uk

IMPROVING GENE NETWORK INFERENCE BY COMPARING EXPRESSION TIME-SERIES ACROSS SPECIES, DEVELOPMENTAL STAGES OR TISSUES

GUILLAUME BOURQUE

Centre de Recherches Mathématiques, Université de Montréal
Montréal, Québec, H3C 3J7, Canada

Genome Institute of Singapore
Singapore 138672, Singapore
bourque@gis.a-star.edu.sg

DAVID SANKOFF

Department of Mathematics and Statistics, University of Ottawa
Ottawa, Ontario, K1N 6N5, Canada
sankoff@uottawa.ca

Received 20 January 2004
Revised 30 July 2004
Accepted 9 August 2004

We present a method for gene network inference and revision based on time-series data. Gene networks are modeled using linear differential equations and a generalized stepwise multiple linear regression procedure is used to recover the interaction coefficients. Our system is designed for the recovery of gene interactions concurrently in many gene regulatory networks related by a tree or a more general graph. We show how this comparative framework can facilitate the recovery of the networks and improve the quality of the solutions inferred.

Keywords: Gene network inference; expression time-series; differential equations; multiple regression.

1. Introduction

Microarrays with their massively parallel capabilities for measuring gene expression have become an attractive tool to reverse-engineer gene regulatory networks.[1,2] The task is challenging as the genes which are part of such networks typically have to be detected within the thousands of other genes found in the genomes. Experiments remain expensive and, with limited data, the problem of sorting through the combinatorial number of potential networks becomes difficult.

The initial work in this area focused on clustering techniques to group genes based on correlations of their expression patterns.[3–5] Aspects of the functioning

of unknown genes could then be predicted based on the function of known genes also found in the same cluster, the method of *guilt-by-association*. Unfortunately, the predictive power of these methods remains limited for gene regulatory network inference as causal relationships between genes are not identified.[6]

Recently, many modeling techniques have been used in an attempt to identify gene interactions and to reconstruct gene networks at a genome-wide scale. These methods include: Boolean models, Bayesian models and models using differential equations. In Boolean models,[7,8] genes are assumed to be ON or OFF and the state configurations are governed by a set of logical rules. Although, these models can be used for the study of global or limiting behaviors, some of their assumptions are very restrictive when used on real networks. Probabilistic Boolean networks[9] attempt to circumvent some of these limitations by expanding to a stochastic set of logical rules but the method still suffers from the drastic discretization of the gene trajectories and a susceptibility to noise. In Bayesian models,[10,11] gene networks are modeled using directed acyclic graphs and conditional distributions. Although such networks can model complex interactions, recovering their structure typically requires lots of data. The fact that cycles can not be incorporated into these models is also problematic as they are known to be an intrinsic part of many biological networks. Finally, although dynamic Bayesian models[12–15] allow for cycles and use of the temporal aspect of the data, they also require discretization and their applicability to the limited data generated by typical microarray studies remains questionable.

In contrast, modeling networks using differential equations[16–20] (see van Someren *et al.*[21] for a review), takes advantage of the continuous aspect of gene expression data. Models in this category allow for feedback loops and, under some simple assumptions, can be solved with relatively few data points. We adopt this approach in the research reported here.

When modeling a system consisting of n genes with differential equations, even under the simplest linear model, there are $n(n-1)$ directional effects, n self effects and n constant effects for a total of $n(n+1)$ unknown parameters. Assume the gene expression of these n genes is measured at T time points, then we have a system with $n(n+1)$ unknown parameters and nT equations. Because in gene expression experiments we typically have that $T \ll n$, the problem is under-determined and extra constraints must be incorporated into the model to unambiguously resolve it. These constraints can involve, for instance, the smoothness of the differential equations.[17] But, recently, a different approach has been to favor the simplicity of the overall solution by minimizing the number of non-zero coefficients.[22–25] Such an approach makes sense biologically since it can be expected that each gene interacts with a limited number of other genes. It will also reduce the complexity of the network and focus on the most important interactions. The fact that very few interactions are necessary to explain gene expression data has been demonstrated by Holter *et al.*[26,27] and Hörnquist *et al.*[28] Thus, a simple model, with as few parameters as possible, may be the most appropriate to use on limited data.

The method we propose follows this track as it models the network by a system of linear differential equations and it limits the number of non-zero coefficients, or regulators, for each gene. The idea is not to force a fixed number of interactions,[24,29] or set a restrictive upper bound,[15] but to favor solutions with few interactions.[22,25] We use a generalized stepwise multiple linear regression procedure to solve the system of equations. Our approach could also be used in conjunction with additional biological knowledge (e.g. a list of genes that potentially play a role in the network, or a list of plausible interactions). The framework that allows for the comparison of alternative solutions could be used to refine the prediction of interactions in networks that are already partially understood. It could identify the best new candidate regulator of a gene or the interaction that is the least corroborated by the data.

Another key aspect of the present study is to define and use a comparative setting for the network inference process. Recently, Stuart *et al.*[30] have demonstrated, in the context of coexpression, that looking at microarray data in different species can improve the prediction of true associations. Our system takes this idea one step further in that it is designed for the recovery of gene interactions concurrently in many gene regulatory networks related by a tree or a more general graph. This evolutionary perspective allows us, for instance, to study a certain regulatory network in different species of known phylogeny. The fixed phylogeny implies a set of relationships between the regulatory networks and we can use this information in the inference process. See Fig. 1 for an example. Alternatively, we might be interested in the stages of development of this network or we could be studying the same system but in different tissues related to each other in various ways. The idea is that, given gene expression data for each species, or each stage of development, or each tissue, we seek to recover each individual network while minimizing a cost based on the differences along the edges of the graph or the tree.

The ultimate test of an inference algorithm is to compare its results to known regulatory networks from the biological literature. It is also important, however, as a preliminary step, to validate the algorithm by evaluating its sensitivity and specificity.[15] This is particularly true in studying the potential impact of using a comparative approach. For these reasons, we start by testing our approach on "realistic" simulations and we show how the comparative framework allows for new insights and facilitates the gene network inference process. We also test our method on a real data set obtained by Wen *et al.*[31] and consisting of measurements of gene expression levels during the development of the central nervous system (CNS) of rats in two different type of tissues (hippocampus and spinal cord).

In Sec. 2 we present the model for single and multiple network recovery. In Sec. 3 we describe the construction of the simulations and show the results of applying the method to simulated and real data. Finally, in Sec. 4 we discuss the strengths and weaknesses of the approach and suggest some future work.

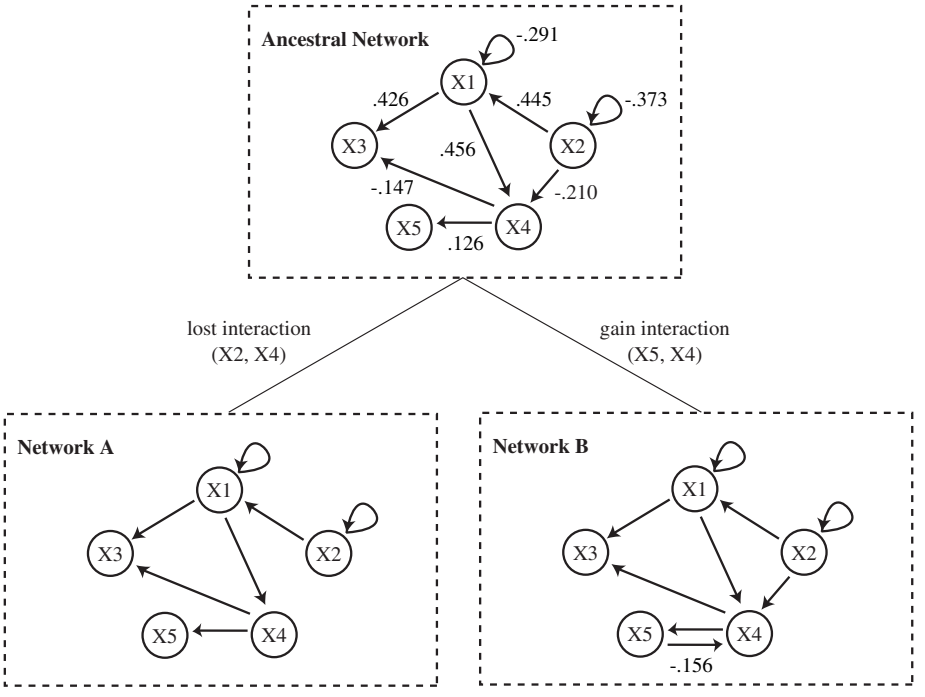


Fig. 1. Two gene networks (A and B) for which expression data are available, and an unobservable ancestral network, which are related by a tree. The networks describe the interactions between fives genes (X1-X5). The interaction coefficients are only displayed in the ancestral network but they are preserved in network A and B except for the interaction (X2, X4) which is lost in network A and the interaction (X5, X4) which is added in network B. The gene expression data is collected for both observable networks and we seek to recover all the interaction coefficients.

2. Method

First, we describe the method for the recovery of gene interactions in a single network using a stepwise multiple regression procedure. Next, we extend our approach to concurrently recover gene interactions in many networks related by a graph or a tree.

2.1. *Single network*

Suppose we have the expression level of n genes at various time-points: $x_1(t), x_2(t), \ldots, x_n(t)$ where $t = 0, 1, \ldots, T$. We call these the *gene trajectories*. When Δt is small enough, we have that the differential equations can be approximated by the difference equations and the linear model which we use to represent the gene expression levels is the following

$$\frac{dx_i(t)}{dt} \approx \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} = y_i(t) = \sum_{j=0,\ldots,n} a_{i,j} x_j(t), \tag{1}$$

where $x_0(t) = 1\ \forall t$. Each $a_{i,j}$ coefficient (when $j \neq 0$) corresponds to the regulatory impact of gene j on gene i and each $a_{i,0}$ coefficient correspond to a constant factor influencing the trajectory x_i. The problem of finding the gene interactions of this network corresponds to finding the matrix of interaction coefficients $A = (a_{i,j})$. As a way of limiting the number of non-zero coefficients, we ask that $n_i < T$ for $i = 1, 2, \ldots, n$ where n_i stands for the number of non-zero coefficients in row i of A. In other words, we ask that each gene be regulated by less than T genes. In this context, we will use stepwise multiple linear regression to solve Eq. (1). There are two stages to the method, first, for a given set of predicted regulators we must estimate the coefficients and evaluate the fit. Second, we must find the best set of predicted regulators through a combinatorial search over all the possible edges of the network.

For each gene i, assume we are given a set of predicted regulators R_i (i.e. $a_{i,j} \neq 0 \Leftrightarrow j \in R_i$) such that $|R_i| = n_i < T$. Then, we have a set of equations

$$y_i(t, R_i) = \sum_{j \in R_i} a_{i,j} x_j(t).$$

We find estimates of the coefficients, $\hat{a}_{i,j}$, by minimizing the total square error

$$SSE(R_i) = \sum_t (y_i(t) - \hat{y}_i(t, R_i))^2 \quad \text{where } \hat{y}_i(t, R_i) = \sum_{j \in R_i} \hat{a}_{i,j} x_j(t).$$

The smaller $SSE(R_i)$, the better the regulators in R_i are at explaining the data observed. In order to choose the best set R_i, or the best model, simply minimizing $SSE(R_i)$ is pointless since adding a new regulator to R_i always lowers $SSE(R_i)$. We need a measure which also takes into account the complexity of the augmented model, such as the number of non-zero coefficients $|R_i|$. For this purpose, we use a partial F test to compare two nested models, R_i and S_i such that $R_i \subset S_i$. The null hypothesis is that R_i, the smaller model, is the better model. We compute

$$F(R_i, S_i) = \frac{SSE(R_i) - SSE(S_i)}{(df(R_i) - df(S_i))(SSE(S_i)/df(S_i))},$$

where

$$df(R_i) = \text{degrees of freedom } (R_i) = T - 1 - |R_i|.$$

Assuming the independence of the observations and the normality of the errors, $F(R_i, S_i)$ should have an F distribution with $(df(R_i) - df(S_i))$ and $df(S_i)$ degrees of freedom. Based on the p-value obtained from this test, we can reject or not reject the null hypothesis.

To find the best set of regulators, the stepwise multiple regression algorithm looks for the best subset of new regulators to be added to the current set of regulators. If the p-value associated with this new subset is below a certain threshold (α), the subset is added. Otherwise, it identifies the best subset of regulators to be removed from the current set of regulators. If the p-value is above a certain threshold (β), the subset of regulators is removed. The procedure iterates until no

Single Network Inference Algorithm$(x_1, \ldots, x_n, \alpha, \beta, depth)$
1. $R_i = \emptyset$
2. $found\_something = true$
3. **while** $(found\_something)$
4. $found\_something = false$
5. $X = \text{find\_best\_subset\_add}(x_1, \ldots, x_n, depth, R_i)$
6. **if** $(p\text{-value}(F(R_i, R_i \cup X)) < \alpha)$ **then**
7. $R_i = R_i \cup X$
8. $found\_something = true$
9. $Y = \text{find\_best\_subset\_remove}(x_1, \ldots, x_n, depth, R_i)$
10. **if** $(p\text{-value}(F(R_i \setminus Y, R_i)) > \beta)$ **then**
11. $R_i = R_i \setminus Y$
12. $found\_something = true$
13. **return** R_i

Fig. 2. In the single network inference algorithm, the *depth* indicates the maximum number of variables we are allowed to add/remove in one step. The functions "find_best_subset_add" and "find_best_subset_remove" identify the subset to add/remove with the smallest/biggest p-value by performing an exhaustive search. α and β are the thresholds for adding and for removing respectively.

new subset of regulators is either added or removed. In the procedure, an extra parameter (*depth*) indicates the maximum size of subsets to be added or removed at any given step. See Fig. 2 for the pseudocode of the algorithm.

2.2. *Graph of networks*

Instead of having a single network to evaluate, we wish to study many networks which are related to each other by a graph $G = (V, E)$. The vertices V can be partitioned into two groups: the *observable networks* (V^0), the networks for which gene expression data has been collected, and the *unobservable networks* ($V \setminus V^0$). The edges E describe the relationships between the networks under study. For an example with two observable networks, one unobservable network and two edges, see Fig. 1. In that example, the graph G actually corresponds to a rooted tree.

With multiple networks, the goal is now not only to minimize the total square error and the complexity of the model for each observable network, but also to minimize the total *evolutionary cost* over the edges of G. Assume that for each gene i and for each vertex v, i.e. each network, we have a set of predicted regulators R_i^v as in the previous section. Assume further that $\mathcal{R}_i = \left(R_i^1, R_i^2, \ldots, R_i^{|V|}\right)$, then the evolutionary cost is defined as:

$$COST(\mathcal{R}_i) = \sum_{(u,v)\in E} |R_i^u \ominus R_i^v|$$

where $\ominus$ is the symmetric difference between two sets. In the current study, the types of events that we consider on the edges of the graph are only insertion and

deletion of predicted regulators. But, the cost function could also be modified to be use in conjunction with more complex evolutionary models.

As in Sec. 2.1, the method we propose recovers the interaction coefficients in two stages. First, for a given vector of sets of predicted regulators we estimate the coefficients, the fit and the cost function. Next, through a combinatorial search, we identify the best vector of sets of predicted regulators. The generalized stepwise multiple linear regression procedure works as follows. Assume there are $N = |V^0|$ observable networks and that each of these network is associated with a set of time-series gene expression data. Then, for a given gene i, a given observable vertex v and a given set of predicted regulators R_i^v, we find estimates of the interaction coefficients, $\hat{a}_{i,j}^v$, by minimizing the total square error

$$SSE(R_i^v) = \sum_t (y_i^v(t) - \hat{y}_i^v(t, R_i^v))^2,$$

where

$$y_i^v(t) = \sum_{j=0,\ldots,n} a_{i,j}^v x_j^v(t) \quad \text{and} \quad \hat{y}_i^v(t, R_i^v) = \sum_{j \in R_i^v} \hat{a}_{i,j}^v x_j^v(t).$$

As in the previous section, we seek a function that will allow the comparison of two nested models. Assume $\mathcal{R}_i$ and $\mathcal{S}_i$ are vectors of sets of predicted regulators, we say that the two models are nested, $\mathcal{R}_i \subset \mathcal{S}_i$, when $R_i^v \subset S_i^v \ \forall k$. If $\mathcal{R}_i$ and $\mathcal{S}_i$ are nested, we define, for all observable v such that $R_i^v \neq S_i^v$, a modified p-value which also takes into account the evolutionary cost:

$$p\text{-value}^*(R_i^v, S_i^v) = p\text{-value}(F(R_i^v, S_i^v)) + \epsilon |COST(\mathcal{R}_i) - COST(\mathcal{S}_i)|,$$

where the impact of the evolutionary cost is controlled by the parameter ϵ. To combine these various modified p-values into a score function allowing the comparison of the two nested models, we make the assumption that they follow a uniform distribution and we use a one-sided Kolmogorov–Smirnov Test. Specifically, we will use the score function

$$G(\mathcal{R}_i, \mathcal{S}_i) = p\text{-value}(KSTest\{p\text{-value}^*(R_i^v, S_i^v) \colon R_i^v \neq S_i^v\}). \tag{2}$$

The comparative inference algorithm for multiple networks works very similarly to the single network inference algorithm described in Sec. 2.1. The differences are that the subsets of regulators are added and removed on edges of the graph and that the Eq. (2) is used to evaluate them. See Fig. 3 for the pseudocode of the algorithm.

For large values of ϵ, the algorithm will converge to a unique gene network which attempts to minimizes the total square error over the gene expression data of all N networks. In contrast, for small values of ϵ, the evolutionary cost will become negligible and the comparative algorithm will work similarly to the single network algorithm except that all N problems will be solved independently, though concurrently.

Comparative Inference Algorithm($\mathcal{T}$, all x_i^v, α, β, *depth*)
1. $\mathcal{R}_i = (\emptyset, \emptyset, \ldots, \emptyset)$
2. $found\_something = true$
3. **while** ($found\_something$)
4. $found\_something = false$
5. **for** each edge e of $\mathcal{T}$
6. $\mathcal{X}$ = find_best_subset_add_edge($\mathcal{T}$, all x_i^v, *depth*, e, $\mathcal{R}_i$)
7. keep best $\mathcal{X}$
8. **if** ($\mathcal{X} \neq (\emptyset, \emptyset, \ldots, \emptyset)$) **then**
9. $\mathcal{R}_i = \mathcal{R}_i \cup \mathcal{X}$
10. $found\_something = true$
11. **for** each edge e of $\mathcal{T}$
12. $\mathcal{Y}$ = find_best_subset_remove_edge($\mathcal{T}$, all x_i^v, *depth*, e, $\mathcal{R}_i$)
13. keep best $\mathcal{Y}$
14. **if** ($\mathcal{Y} \neq (\emptyset, \emptyset, \ldots, \emptyset)$) **then**
15. $\mathcal{R}_i = \mathcal{R}_i \setminus \mathcal{Y}$
16. $found\_something = true$
17. **return** $\mathcal{R}_i$

Fig. 3. In the comparative inference algorithm, the *depth* indicates the maximum number of variables we are allowed to add/remove in one step. The functions "find_best_subset_add_edge" and "find_best_subset_remove_edge" identify the subset to add/remove on a given edge with the smallest/biggest value for the function (2) by performing an exhaustive search. In "find_best_subset_add_edge", we only consider subsets $\mathcal{X}$ for which p-value$^*(R_i^v, R_i^v \cup X^v) < \alpha\ \forall v$ such that $X^v \neq \emptyset$. Similarly, we only consider removing subsets $\mathcal{Y}$ for which p-value$^*(R_i^v \setminus Y^v, R_i^v) > \beta\ \forall v$ such that $Y^v \neq \emptyset$. This is to avoid backtracking in the algorithm.

3. Results

First, we describe how single networks are simulated. Next, we show results of using both the single network algorithm and the comparative algorithm on two distinct types scenarios with multiple networks. Finally, we describe the results of using the comparative inference algorithm on the CNS data set obtained by Wen *et al.*[31]

3.1. *Simulation of networks*

Simulating realistic networks of genes is not just a question of generating random regulator coefficients; this just tends to produce fragmented networks each component of which is unstable. Actual networks of genes have taken eons to evolve into the complex and stable systems that they are. To benchmark our algorithms and to simulate gene expression data sets, we have used three guidelines: (1) gene trajectories must remain bounded, (2) the correlation between any two gene trajectories must be limited, and (3) gene trajectories cannot be overly stable.

The first guideline makes reference to the fact that gene expression levels can neither become negative nor too large. The last two guidelines are directly related

to the assumptions inherent to the method. Because we are using multiple linear regression to recover the interaction coefficients, pairs of gene trajectories which are too highly correlated ($|\rho| > \rho_{\max}$) will typically be indistinguishable at the level of our analysis. Genes with such highly similar profiles would require other experiments, such as perturbation analysis, before their distinctive roles can be determined. Finally, because we seek to model the response of the genes to their regulators, trajectories with extremely low levels of response will be most difficult to model as their signal will be lost in noise. It is not even clear whether such genes are true members of the network. For that reason, a gene trajectory x_i will be ruled to be too stable and rejected if $\sum_{t=1,\ldots,T}(x_i(t) - x_i(t-1))^2 < s_{\min}\ T$.

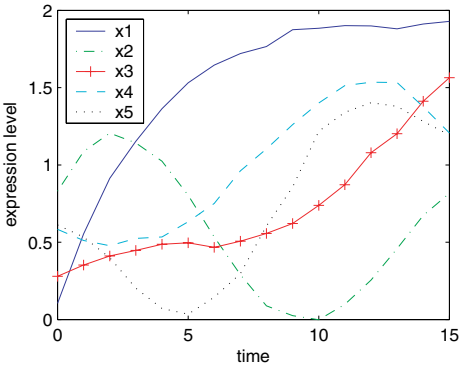
Assume we want to simulate a network with n genes such that each gene has at most c regulators. The goal is to find a matrix of interaction coefficients and a vector of initial gene expression levels such that the gene trajectories implied by these values satisfy the three conditions mentioned above. For that purpose, for each gene i (or each row of the interaction coefficients matrix), we randomly select n_i coefficients (or regulators) and assign them random values uniformly in the interval $[-b, +b] \setminus [-\delta, \delta]$. The choice of this interval is somewhat arbitrary but was selected to help the trajectories satisfy the three conditions. The actual number of regulators, n_i, is itself randomly selected uniformly between 1 and c. Finally, we also select a random starting point uniformly in the interval [0,1]. All the gene trajectories which do not satisfy the three guidelines stated above are removed and replaced by new gene trajectories randomly selected similarly. The procedure iterates until all gene trajectories satisfy the three conditions. Finally, gaussian noise is added to the gene expression data with mean $\mu = 0$ and various levels of standard deviation σ.

3.2. *Simulated networks: Scenario 1*

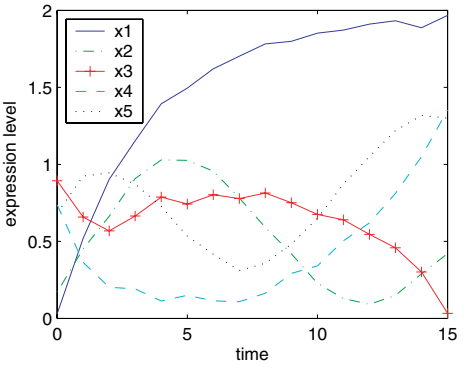
We start with simulations involving two networks. The first network, network A, is simulated as in Sec. 3.1. The second network, network B, is generated from network A by adding a new interaction coefficient to its coefficients matrix. The procedure used is similar to the one of the previous section. A null coefficient in the coefficients matrix of network A is randomly selected and randomly assigned to a value in the interval $[-b, +b] \setminus [-\delta, \delta]$. If one of the gene trajectories no longer satisfies one of the three conditions stated above, different starting points and different values for the new coefficient are tested. If no value succeeds at generating an acceptable set of gene trajectories, a different null coefficient in the interaction coefficient matrix is tested. If no new interaction coefficient is found in this process, network A is replaced by a new network and we repeat the procedure.

An example with two networks, 5 genes and 15 time-points is shown in Fig. 4. In that example, the maximum number of regulators per gene, c, was set to 3 and the noise level, σ, was set to 0.025. Network A has 10 non-null interaction coefficients while network B has one more with 11. For a wide range of the regression

(a) Gene trajectories for network A



(b) Gene trajectories for network B



(c) Matrix of interaction coefficients

x_1	0.491	−0.248	0	0	0	0
x_2	0	−0.190	0	0	0	0.399
x_3	0	0	0	0	**−0.390**	0.125
x_4	0	0.233	0	−0.453	0	0
x_5	0.484	0	−0.473	0	0	−0.256

Fig. 4. Simulation example from Scenario 1 with two networks, 5 genes, 15 time points and noise level $\sigma = 0.025$. (a) The gene trajectories for network A. (b) The gene trajectories for network B. (c) The matrix of interaction coefficients for both networks where the only difference is the value in bold (−0.390) which is present in network B and absent in network A. As a result, both the x_3 trajectory and the x_4 trajectory are significantly different.

parameters (α, β) and of the evolutionary cost (ϵ), the comparative inference algorithm successfully recovers these two networks. For this example, the single network algorithm, applied to both networks independently, also successfully recovers the networks.

More generally, we tested the method on instances of the problem with two networks, 5 genes, and different numbers of time points $(T = 5, 10, 15, 20, 25, 30)$, of maximum number of regulators $(c = 2, 3)$ and of noise levels $(\sigma = 0.01, 0.025)$. To measure the quality of the solutions recovered by the algorithm, we looked at the *False Positive* (*FP*) *rate* and the *False Negative* (*FN*) *rate* where

$$\text{False Positive rate} = \frac{\text{nb false positive interactions}}{\text{nb actual interactions}} * 100$$

and the FN rate is defined similarly. Note that the FP rate could theoretically be over 100. For each triplet (T, c, σ), we generated 20 pairs of networks and we computed the average FP rate and the average FN rate in the solutions recovered. We used both the single network inference algorithm, applied to each network independently, and the comparative inference algorithm. The results are shown in Figs. 5a–d.

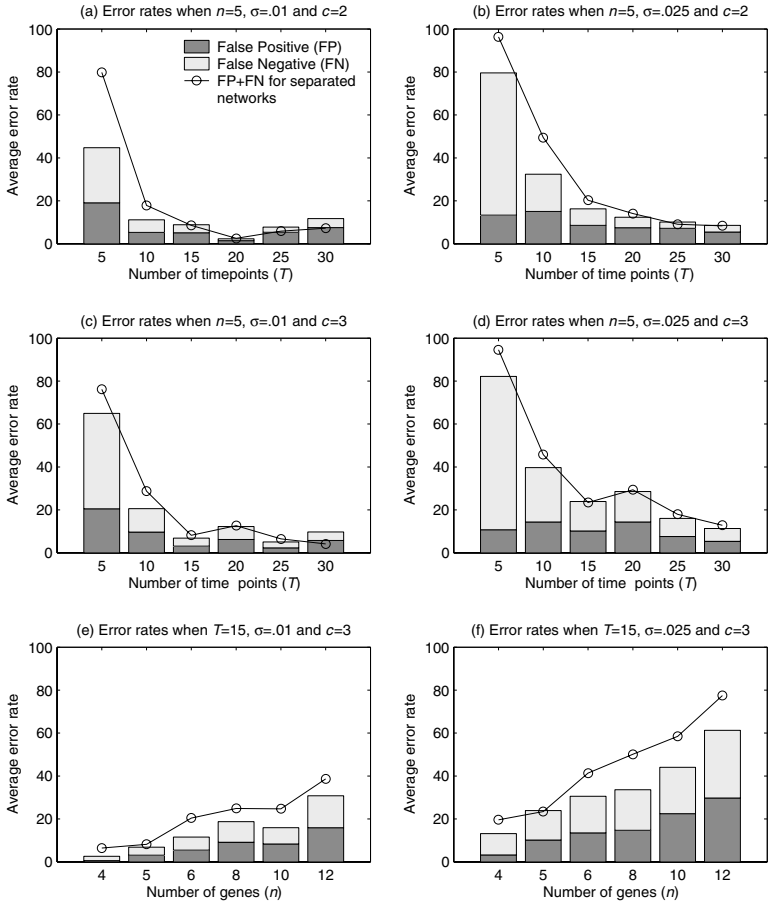


Fig. 5. Simulation results for Scenario 1 in which network B has one more interaction than network A. (a) The two networks have 5 genes and various numbers of time-points ($T = 5, 10, 15, 20, 25, 30$). The noise level, σ, is set to 0.01 and the maximum number of regulators per gene, c is set to 2. (b) Same except $\sigma = 0.025$ and $c = 2$. (c) Same except $\sigma = 0.01$ and $c = 3$. (d) Same except $\sigma = 0.025$ and $c = 3$. (e) The number of time-points, T, is set to 15, σ is set to 0.01, c is set to 3 and the results for various numbers of genes ($n = 4, 5, 6, 8, 10, 12$) are displayed. (f) Same except $\sigma = 0.025$. For each set of parameters, 20 pairs of networks were simulated and the average FP rate and FN rate were computed after running the comparative algorithm. Also shown, is the average cumulative rate (FP rate + FN rate) induced by running the algorithm on each network independently.

We experimented with many combinations of parameters for the stepwise procedure (α, β) and for the evolutionary cost (ϵ) (data not shown). Obviously, the choice of these parameters depends greatly on the data sets on which we want to apply the algorithm. Assume we want to minimize the false negative rate, this would suggests higher values for both α and β. In contrast, very accurate data would suggest smaller threshold values. For the purpose of this work, and because of its overall

good performance, we selected a specific triplet ($\alpha = 0.01$, $\beta = 0.03$, $\epsilon = 0.005$) for all runs of the comparative version of the algorithm. The single network version of the algorithm performed better with slightly smaller parameter values and so, after conducting more tests, we selected ($\alpha = 0.001$ and $\beta = 0.01$) for all the results described in this report. The parameter *depth*, the maximum size of subsets to be added or removed, was set to 3 throughout this work.

The results in Figs. 5a–d show that the comparative algorithm performs better than the regular single network algorithm especially for difficult problems (e.g. problems with very few time-points). To test this idea further, we constructed new simulations involving more genes in each network. For these simulations, there are 15 time-points, the maximum number of regulators per gene is set to 3, the number of genes varies $n = 4, 5, 6, 8, 10, 12$ and so does the noise level $\sigma = 0.01, 0.025$. The results are shown in Figs. 5e–f. We see once again that for difficult instances of the problem, simulations with more genes per network hence more potential regulators, the comparative algorithm outperforms the single network algorithm.

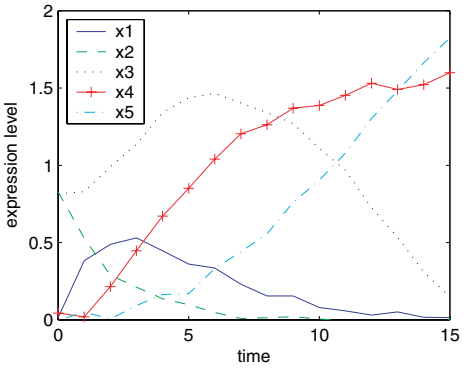
3.3. *Simulated networks: Scenario 2*

The simulations also involve two networks except that network A and network B are now generated from an ancestral network which we will consider has being unobservable. See Fig. 1 for an example. The ancestral network is simulated as in Sec. 3.1. Network A is generated from the ancestral network by removing one of its interactions. The procedure is the following: a non-null coefficient of the ancestral network is selected at random and is set to 0. The trajectories of the new network are tested as in Sec. 3.1 and, if one them fails to satisfy the conditions, a different non-null coefficient is set to 0. If no coefficient can be remove in that way, the ancestral network is replaced and the process restarts. Once a coefficient is found and network A is generated, the second network, network B, is generated by adding a new interaction coefficient to the ancestral network. The procedure used is similar to the one described in Sec. 3.2. If no coefficient can be added, the ancestral network is replaced, and we start looking for network A again.

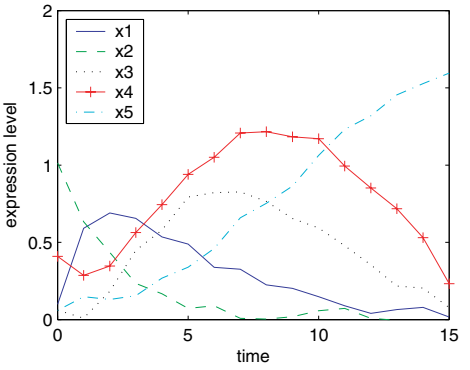
The trajectories and the coefficient matrix for the example with two networks, 5 genes and 15 time-points that was shown in Fig. 1 are shown in Fig. 6. In that example, the maximum number of regulators per gene, c, was set to 3 and the noise level, σ, was set to 0.025. Network A has 8 non-null interaction coefficients while network B has two more with 10. For a wide range of the regression parameters (α, β) and of the evolutionary cost (ϵ), the comparative inference algorithm successfully recovers these two networks. In contrast, the single network algorithm, applied to both networks separately, predicted the wrong set of regulators for x_1 in network A.

We tested the method on instances of the problem with 5 genes, and different numbers of time points ($T = 5, 10, 15, 20, 25, 30$) and of noise levels ($\sigma = 0.01, 0.025$). The maximum number of regulators c was set to 3. We used both

(a) Gene trajectories for network A



(b) Gene trajectories for network B



(c) Matrix of interaction coefficients

x_1	0	−0.291	0.445	0	0	0
x_2	0	0	−0.373	0	0	0
x_3	0	0.426	0	0	−0.147	0
x_4	0	0.456	*−0.210*	0	0	**−0.156**
x_5	0	0	0	0	0.126	0

Fig. 6. Simulation example from Scenario 2 with two networks, 5 genes, 15 time points and noise level $\sigma = 0.025$ continued from Fig. 1. (a) The gene trajectories for network A. (b) The gene trajectories for network B. (c) The interaction coefficients where the value in bold (−0.156) was added in network B and the value in italic (−0.210) was removed from network A. As a result, the x_4 trajectory is significantly different.

the single network inference algorithm, applied to each network separately, and the comparative inference algorithm. The results are shown in Figs. 7a–b. We also constructed simulations involving more genes in each network. For these simulations, $T = 15$, $c = 3$, the number of genes varies $n = 4, 5, 6, 8, 10, 12$ and so does the noise level $\sigma = 0.01, 0.025$. The results are shown in Figs. 7c–d. The results confirm that the comparative algorithm performs better than the single network algorithm especially with few time-points or many genes (potential regulators).

3.4. *Real networks: central nervous system of rats*

Even with the growing availability of new gene expression data sets, finding data sets for two or more related systems remains difficult. To validate our approach on real data, we use the data obtained by Wen *et al.*,[31] which consists of measurement taken during the development of the central nervous system of rats. The particularity of this data set is that the expression levels are measured in two different developing tissues: hippocampus and spinal cord. This allows us to test our method and look for similarity and differences between the two networks. The original data consists

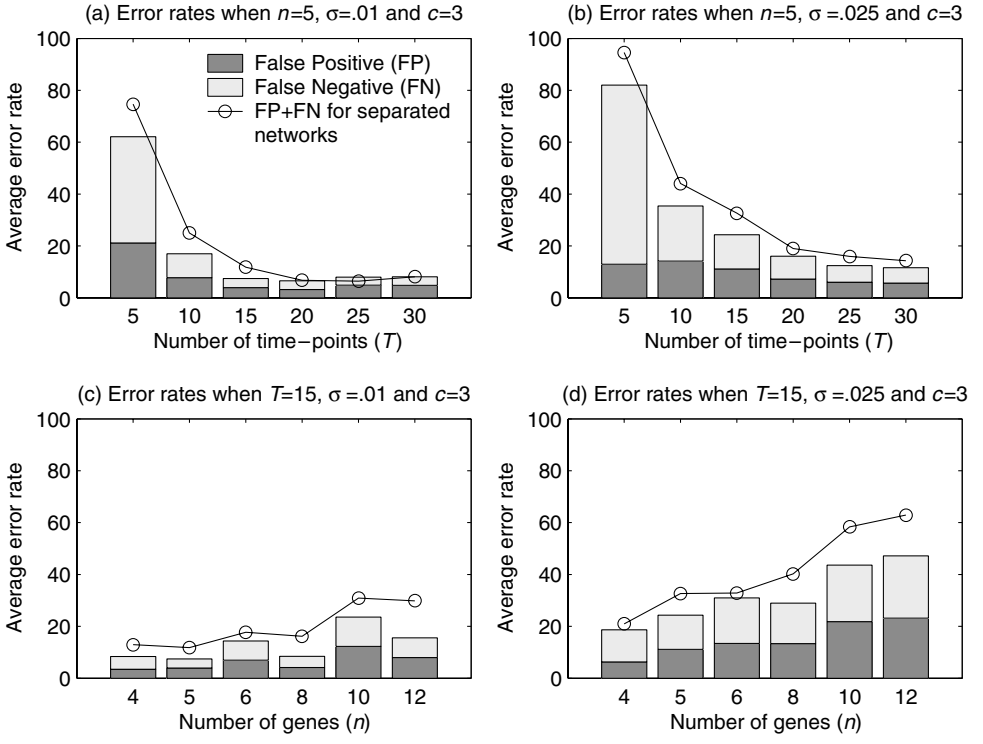


Fig. 7. Simulation results for Scenario 2 in which an ancestral network is simulated and one interaction is removed/added to the network to generate network A and B respectively. (a) The networks have 5 genes and various numbers of time-points ($T = 5, 10, 15, 20, 25, 30$). The noise level, σ, is set to 0.01 and the maximum number of regulators per gene, c is set to 3. (b) Same except $\sigma = 0.025$ and $c = 3$. (c) The number of time-points, T, is set to 15, σ is set to 0.01, c is set to 3 and the results for various numbers of genes ($n = 4, 5, 6, 8, 10, 12$) are displayed. (d) Same except $\sigma = 0.025$.

of 112 genes measured at 9 different points in time although not all genes are measured in both tissues.

This data set was also analyzed by Wahde and Hertz[19] in an attempt to reverse engineer the underlying gene network. In order to obtain comparable results, we follow the same guidelines in preprocessing the data. First, we only keep the 66 genes for which the expression levels are measured in both tissues. Next, we only use the first 8 data points taken in the interval starting 10 days before birth and ending 7 days after birth, ignoring the last data point which was measured in the adult animal. Finally, following the procedure used in Wen *et al.*,[31] we cluster these genes into five "waves," each having a distinct temporal expression profile. As in Wahde and Hertz,[19] we ignore the fifth wave corresponding to genes having an essentially constant expression level since we do not expect them to play a dynamical role that we can model. We show in Figs. 8a–b the trajectories of the four waves for the hippocampal and spinal cord tissues.

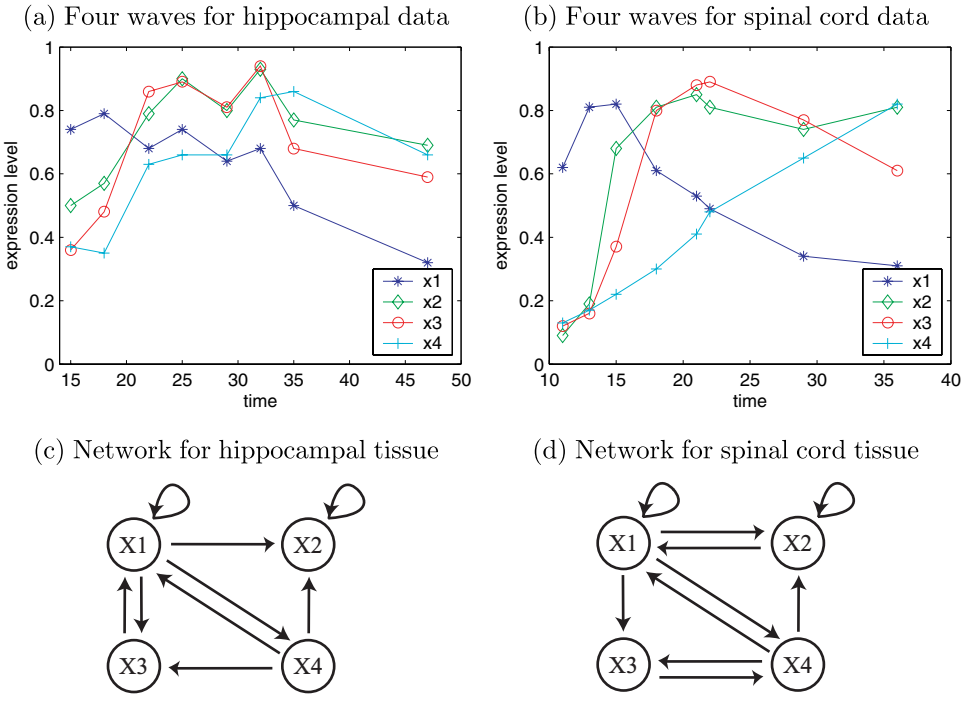


Fig. 8. Waves of expression levels and recovered interaction networks for CNS (Central Nervous System) data set. (a) Four waves of expression levels for hippocampal data. (b) Four waves of expression levels for spinal cord data. (c) Recovered interaction network for hippocampal tissue using the comparative inference algorithm with ($\alpha = 0.2$, $\beta = 0.3$, $\epsilon = 0.1$). (d) Same but for spinal cord tissue.

We used our comparative inference algorithm with relatively high thresholds for the stepwise regression procedure because of the limited amount of data (only 8 time points) and the high noise level. More specifically, we selected ($\alpha = 0.2$, $\beta = 0.3$); lower thresholds left some of the waves with absolutely no regulators. Next, we made use of our evolutionary cost parameter ϵ to test various hypothesis. In the first experiment, we set $\epsilon = \alpha/2$ to allows some flexibility/differences between the networks of two tissues. The networks we recovered are shown in Figs. 8c–d. We observed two new interactions ((X2, X1) and (X3, X4)) and the loss of one interaction (X3, X1) in the spinal cord network vis-a-vis the hippocampal network. In contrast, setting the evolutionary cost parameter at a preemptively high level (e.g. $\epsilon = \alpha$) forces the inference of two networks with no topological differences, identical to the hippocampal network found for $\epsilon = \alpha/2$ and shown in Fig. 8c. Testing intermediate values of ϵ (i.e. $\epsilon \in [\alpha/2, \alpha]$) allows us to determine that the difference between these two networks that is the most corroborated by the data is the new interaction (X3, X4) in the spinal cord network. Indeed, we can observe that the fourth wave is perhaps the wave which is the most different in the two networks (see Figs. 8a–b).

Because independent biological characterization of the interactions between these waves of genes is not yet available, there are no external criteria for evaluating the quality of the solutions. Instead, Wahde and Hertz[19] used an internal, *ad hoc*, method to identify iteratively which of the interaction coefficients they recovered were significant. They call these *significant connections*. We can observe that most of these interactions were also recovered by our method: (X1, X2), (X1, X3), (X1, X4), (X3, X4), (X4, X1) and (X4, X3).

4. Discussion

The comparative approach to network inference involves studying them in several species, developmental stages or tissues concurrently. The method we propose combines gene expression data on a number of related networks in order to improve the inference of gene interactions. The advantage lies in using the similarity among the networks to reduce uncertainty and noise. The method could be applied, for instance, to detect small differences in a given network in a set of closely related species. In contrast to some previous studies,[15,24,29] the number of regulators for each gene is selected dynamically and conditioned to be low but with no arbitrary upper limit. This flexibility is probably more reasonable biologically. Our simulations in Sec. 3 illustrate how the comparative framework helps recover the gene interactions in two networks that have evolved with a small change in the number of gene interactions. The improvement over independent analyses of each individual network becomes more significant as the problem becomes more difficult (i.e. fewer time-points, more genes or more noise).

Our method trades off the fit within the networks, i.e. between the data and each inferred network, against the evolutionary cost over a graph representing the relationships among the networks. The evolutionary cost function here is simply a count of the number of insertions and deletions of non-zero interaction coefficients over the edges of the graph, but other functions could also be developed to take into account more complex evolutionary processes. For example, we could extend our qualitative cost function, based only on the presence or absence of interactions, to a quantitative measure based on changes in the coefficients associated with the predicted sets of regulators. Similarly, we could extend the evolutionary cost function so that it weights various kinds of network alteration differently.

An advantage of our method is that it can conveniently incorporate biological knowledge at every stage of the decision process. It would be possible to pre-classify genes and interaction coefficients into categories ranging from "should be in final network" to "should not be in final network". The procedure that decides which edge should be added or removed from the network could then be adapted to take into account the pre-determined categories. This could enable the user to force, suggest, or disallow genes and interactions into the final network. The method could also be used to improve an existing model of a gene network by identifying the interaction coefficient least corroborated by the observed data or finding the

new coefficient most justified by the observed data. In still another application, if a network has been well characterized in one species, it could be used as an initial guess in inferring the analogous network in a related species by looking for changes in the original network. A benefit of using an iterative algorithm that compares alternative solutions is that a user can monitor the output from the algorithm and potentially overrule difficult decisions. If two potential interactions score almost as well at some point, a user could take advantage of prior knowledge to influence the decision.

There are also some limitations to the model we propose. For instance, our analysis of expression levels assumes the effects of the stoichiometric and kinetic parameters of the biochemical reactions are all rolled into the linear coefficients of our model, and that any nonlinear regulatory behavior will not obscure our results. Given the nature and the quality of the data available from a typical microarray time-series experiment at present, we cannot hope to estimate additional parameters to account for such effects separately. Even if we were to constrain the majority of these effects to be zero, with very few time-points, a more general model would tend to over-fit the data and completely miss the true interactions of the network. Using a simpler model allows us to detect the true interactions in those regulatory networks that can reasonably be modeled by a system of linear differential equations.

The most important point of the present study is to demonstrate how looking at the problem of network inference from an evolutionary perspective can help improve the quality of the solutions recovered. The framework itself is not restricted to multiple linear regression and could be applied to other methods (e.g. robust regression, nonlinear regression or Bayesian networks). Similar ideas of comparative analysis could also be explored for other types of microarray experiments (e.g. perturbation analysis).

Acknowledgements

GB holds a post-doctoral fellowship from the Quebec Minister of Science and Technology. DS holds the Canada Research Chair in Mathematical Genomics and is a Fellow in the Evolutionary Biology Program of the Canadian Institute for Advanced Research. Research supported by grants from the Natural Sciences and Engineering Research Council (NSERC).

References

1. Lockhart DJ, Winzeler EA, Genomics, gene expression and DNA arrays, *Nature* **405**(6788):827–836, 2000.
2. Schulze A, Downward J, Navigating gene expression using microarrays — a technology review, *Nat Cell Biol* **3**(8):190–195, 2001.
3. DeRisi JL, Iyer VR, Brown PO, Exploring the metabolic and genetic control of gene expression on a genomic scale, *Science* **278**(5338):680–686, 1997.

4. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B, Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization, *Mol Biol Cell* **9**(12):3273–3297, 1998.
5. Roberts CJ, Nelson B, Marton MJ, Stoughton R, Meyer MR, Bennett HA, He YD, Dai H, Walker WL, Hughes TR, Tyers M, Boo C, Friend SH, Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles, *Science* **287**(5454):873–880, 2000.
6. Quackenbush J, Genomics. microarrays — guilt by association, *Science* **302**(5643): 240–241, 2003.
7. Akutsu T, Miyano S, Kuhara S, Identification of genetic networks from a small number of gene expression patterns under the Boolean network model, in *Pac Symp Biocomput* pp. 17–28, 1999.
8. Liang S, Fuhrman S, Somogyi R, Reveal, a general reverse engineering algorithm for inference of genetic network architectures, in *Pac Symp Biocomput* pp. 18–29, 1998.
9. Shmulevich I, Dougherty ER, Kim S, Zhang W, Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks, *Bioinformatics* **18**(2): 261–274, 2002.
10. Friedman N, Linial M, Nachman I, Pe'er D, Using Bayesian networks to analyze expression data, *J Comput Biol* **7**(3–4):601–620, 2000.
11. Imoto S, Goto T, Miyano S, Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression, in *Pac Symp Biocomput* pp. 175–186, 2002.
12. Smith VA, Jarvis ED, Hartemink AJ, Evaluating functional network inference using simulations of complex biological systems, *Bioinformatics* **18**(Suppl 1):216–224, 2002.
13. Ong IM, Glasner JD, Page D, Modelling regulatory pathways in E. coli from time series expression profiles, *Bioinformatics* **18**(Suppl 1):241–248, 2002.
14. Perrin BE, Ralaivola L, Mazurie A, Bottani S, Mallet J, D'Alche-Buc F, Gene networks inference using dynamic Bayesian networks, *Bioinformatics* **19**(Suppl 2): II138–II148, 2003.
15. Husmeier D, Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks, *Bioinformatics* **19**(17):2271–2282, 2003.
16. Chen T, He HL, Church GM, Modeling gene expression with differential equations, *Pac Symp Biocomput* pp. 29–40, 1999.
17. D'Haeseleer P, Wen X, Fuhrman S, Somogyi R, Linear modeling of mRNA expression levels during CNS development and injury, in *Pac Symp Biocomput* pp. 41–52, 1999.
18. Wahde M, Hertz J, Coarse-grained reverse engineering of genetic regulatory networks, *Biosystems* **55**(1–3):129–136, 2000.
19. Wahde M, Hertz J, Modeling genetic regulatory dynamics in neural development, *J Comput Biol* **8**(4):429–442, 2001.
20. Weaver DC, Workman CT, Stormo GD, Modeling regulatory networks with weight matrices, in *Pac Symp Biocomput* pp. 112–123, 1999.
21. van Someren E, Wessels L, Reinders M, Genetic network models: a comparative study, in *Proc. of SPIE* 2001.
22. De Hoon M, Imoto S, Miyano S, Inferring gene regulatory networks from time-ordered gene expression data using differential equations, in Steffen Lange, Ken Satoh, and Carl H. Smith, (eds.), *Fifth International Conference on Discovery Science*, Lecture Notes in Computer Science 2534, pp. 267–274, Lbeck, Germany, Springer-Verlag, 2002.

23. van Someren EP, Wessels LFA, Reinders MJT, Information extraction for modeling gene expressions, in Biemond J, (ed.), *21st Symp. on Information Theory in the Benelux*, pp. 215–222, Wassenaar (NL), 2000. Werkgemeenschap Informatieen Communicatietheorie, Enschede (NL).
24. van Someren EP, Wessels LFA, Reinders MJT, Backer E, Searching for limited connectivity in genetic network, in *Proc. of ICSB* 2001.
25. Yeung MKS, Tegner J, Collins JJ, Reverse engineering gene networks using singular value decomposition and robust regression, *Proc Natl Acad Sci USA* **99**(9):6163–6168, 2002.
26. Holter NS, Maritan A, Cieplak M, Fedoroff NV, Banavar JR, Dynamic modeling of gene expression data, *Proc Natl Acad Sci USA* **98**(4):1693–1698, 2001.
27. Holter NS, Mitra M, Maritan A, Cieplak M, Banavar JR, Fedoroff NV, Fundamental patterns underlying gene expression profiles: simplicity from complexity, *Proc Natl Acad Sci USA* **97**(15):8409–8414, 2000.
28. Hornquist M, Hertz J, Wahde M, Effective dimensionality of large-scale expression data using principal component analysis, *Biosystems* **65**(2–3):147–156, 2002.
29. Gardner TS, di Bernardo D, Lorenz D, Collins JJ, Inferring genetic networks and identifying compound mode of action via expression profiling, *Science* **301**(5629): 102–105, 2003.
30. Stuart JM, Segal E, Koller D, Kim SK, A gene-coexpression network for global discovery of conserved genetic modules, *Science* **302**(5643):249–255, 2003.
31. Wen X, Fuhrman S, Michaels GS, Carr DB, Smith S, Barker JL, Somogyi R, Large-scale temporal gene expression mapping of central nervous system development, *Proc Natl Acad Sci USA* **95**(1):334–339, 1998.

Guillaume Bourque is a Group Leader at the Genome Institute of Singapore in the department of Information & Mathematical Sciences. He received his Ph.D. degree in Applied Mathematics at the University of Southern California in 2002. His main research interest is the analysis of genome rearrangements in multiple species.

David Sankoff holds the Canada Research Chair in Mathematical Genomics at the University of Ottawa. His current research focuses on the evolution of genomes as the result of chromosomal rearrangement processes. He is a Fellow of the Royal Society of Canada and of the Canadian Institute for Advanced Research, and was the recipient of the first Senior Scientist Accomplishment Award of the International Society for Computational Biology.