Statistical Applications in Genetics and Molecular Biology

Volume 9, *Issue* 1 2010 *Article* 9

An Empirical Bayesian Method for Estimating Biological Networks from Temporal Microarray Data

Andrea Rau[*] Florence Jaffrézic[†]

Jean-Louis Foulley[‡] Rebecca W. Doerge[**]

[*] Purdue University, INRA AgroParisTech, arau@stat.purdue.edu

[†] INRA AgroParisTech, florence.jaffrezic@jouy.inra.fr

[‡] INRA AgroParisTech, jean-louis.foulley@jouy.inra.fr

[**] Purdue University, doerge@purdue.edu

An Empirical Bayesian Method for Estimating Biological Networks from Temporal Microarray Data[*]

Andrea Rau, Florence Jaffrézic, Jean-Louis Foulley, and Rebecca W. Doerge

Abstract

Gene regulatory networks refer to the interactions that occur among genes and other cellular products. The topology of these networks can be inferred from measurements of changes in gene expression over time. However, because the measurement device (i.e., microarrays) typically yields information on thousands of genes over few biological replicates, these systems are quite difficult to elucidate. An approach with proven effectiveness for inferring networks is the Dynamic Bayesian Network. We have developed an iterative empirical Bayesian procedure with a Kalman filter that estimates the posterior distributions of network parameters. We compare our method to similar existing methods on simulated data and real microarray time series data. We find that the proposed method performs comparably on both model-based and data-based simulations in considerably less computational time. The R and C code used to implement the proposed method are publicly available in the R package ebdbNet.

[*] We gratefully acknowledge helpful comments from Paul L. Auer, computing help from My Truong and Doug Crabill, useful discussion and Matlab code from David L. Wild, and R code for the VAR method from Korbinian Strimmer. We are grateful to Guy Demoment and Ali Mohammad-Djafari (SUPELEC, University Paris Sud) and Gilles Celeux (INRIA-Paris Sud) for their insightful comments on the inference procedures used for state space models in the early stages of this work. We also thank an anonymous reviewer for helpful comments and suggestions. This research is supported by a grant from the National Science Foundation Plant Genome (DBI 0733857) to RWD.

1 Introduction

A gene regulatory network is loosely defined as a set of genes that interact with one another through other genes, transcription factors, and protein products. Because these interactions contribute to the regulation of gene transcription and translation, the structure of gene regulatory networks plays an important role in cell behavior and structure. Structurally, these gene regulatory networks tend to have several properties in common. First, most genes are regulated just one step away from their regulator, and long regulatory cascades are rare (Alon, 2007). In addition, feedback loops, also known as self-regulating processes, are common motifs in the network structure (Brandman and Meyer, 2008). Finally, gene networks tend to be sparse; that is, genes are typically only influenced by a limited number of other genes (Leclerc, 2008).

As microarray technology has become increasingly accessible and affordable, the goal of inferring gene regulatory networks from temporal gene expression data has received growing interest in the systems biology community. Over the past ten years, a variety of techniques have been proposed to infer network structure from microarray data, including Boolean networks (D'haeseleer et al., 2000), mutual information networks (Basso et al., 2005), ordinary differential equations (Cao and Zhao, 2008), Graphical Gaussian Models (Schäfer and Strimmer, 2005), and auto-regressive models (Opgen-Rhein and Strimmer, 2007). In addition to these techniques, the seminal works of Murphy and Mian (1999) and Friedman (2000) have motivated a growing body of work dedicated to using Bayesian networks and dynamic Bayesian networks for biological network inference (see Husmeier, 2003; Perrin et al., 2003; Zou and Conzen, 2005).

Linear Gaussian state space models (SSM), a subclass of dynamic Bayesian networks, seem to be particularly well-suited for dealing with time-series gene expression data, as they incorporate a number of attractive features, including: a) the ability to handle continuous, noisy data, b) the ability to model the effect of hidden variables, such as proteins, transcription factors, or genes not included on a particular microarray, and c) the use of Markovian dynamics to describe the fluctuations in gene expression measurements and hidden variables over time. However, because the number of time points and biological replicates in microarray data are typically much smaller than the number of genes, the classic $n \ll p$ paradigm requires some care in the estimation of model parameters in the SSM. In addition, choosing the dimension of the state space remains a difficult statistical problem with ramifications to the applicability of state space models in gene regulatory networks.

Recently, several authors have examined the use of state space models for reverse-engineering gene regulatory networks. Perrin et al. (2003) applied a generalized Expectation-Maximization (EM) algorithm with a parsimony constraint on network connections to penalize the model likelihood, but limited the choice of the hidden state dimension to 0, 1, or 2. Wu et al. (2004) used a factor analysis and Bayesian Information Criterion (BIC) penalization for model selection. More recently, Bremer and Doerge (2009) used a SSM model with Kalman smoothing and maximum likelihood estimation techniques to identify regulated genes in time-course gene expression data. Beal et al. (2005) considered a SSM with feedback in a hierarchical Bayesian framework, using a Variational Bayes procedure to calculate a bound on the marginal likelihood in order to learn the network structure and the dimensionality of the hidden state. The hierarchical nature of this prior structure is particularly appealing, as its structure allows a shrinkage of the network parameters towards zero, corresponding to the biological assumption of network sparsity.

In this paper, we introduce an empirical Bayes estimation procedure for a feedback state space model in a hierarchical Bayesian framework that is complementary to the method developed by Beal et al. (2005). Both models rely on an approximation to a full hierarchical Bayesian analysis, base model inference on the posterior distributions of model parameters, and explicitly model the feedback of gene expression from one time to another. However, in this work we take advantage of the defined prior structure to implement a straightforward estimation of the hyperparameters using an EM-like algorithm, and the singular value decomposition (SVD) of a block-Hankel matrix to determine the dimension of the hidden state *a priori* (Bremer, 2006). This significantly reduces the computation time required for the algorithm to run, as it eliminates the need to run the algorithm over a wide range of values for the hidden state dimension (as is the case in the variational Bayes procedure).

2 System and Methods

2.1 Dynamic Bayesian Networks

Bayesian networks (BN) have become a popular tool used for the inference of gene regulatory networks, due in part to their flexibility and intuitive interpretation. BN are formally defined by a graphical structure $M = \{V, E\}$ made up of a set of vertices and edges, and a family of conditional probability distributions F parameterized by q (see Husmeier et al., 2005, for greater detail). Consequently, BN fall in the intersection of graph theory and probability theory, as they use

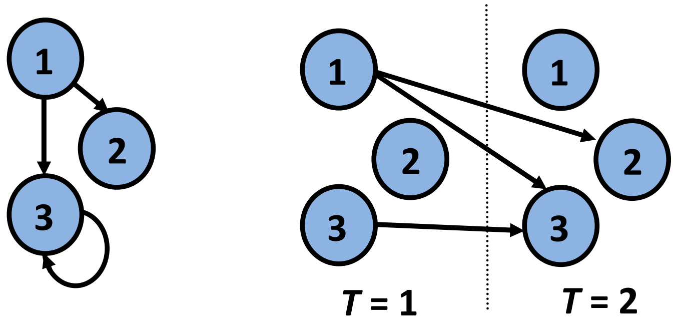

Figure 1: (Left) A network with three nodes and three edges, including one feedback loop for node 3. Due to the presence of this feedback loop, this network does not meet the acyclicity constraint of Bayesian networks. (Right) The same network, unrolled over time as a Dynamic Bayesian network. By directing arrows with respect to the flow of time, the network shown on the left can be fully represented without violating the acyclicity constraint, despite the presence of a feedback loop (similar image shown in Husmeier et al., 2005).

graphical models to represent the conditional probabilistic relationships among a set of random variables.

In the context of gene networks, however, three properties of BN have limited their applicability. First, data is typically discretized for BN analysis, which incurs a loss of information from continuous gene expression measurements. Second, BN must be Directed Acyclic Graphs (DAG), excluding the possibility of representing feedback loops in the graphical structure. Third, the existence of equivalence classes is possible, implying in some cases that changing the direction of an arrow in the graph will yield the same factorization over the joint probability.

To deal with these limitations, we use a Dynamic Bayesian Network (DBN), which essentially unfolds a Bayesian network over time. In a DBN, continuous observations may be used without the need for discretization. Furthermore, because edges are directed with respect to the flow of time, the acyclicity constraint can be met without eliminating feedback loops, and any ambiguity in the direction of the arrows can be resolved (see Figure 1).

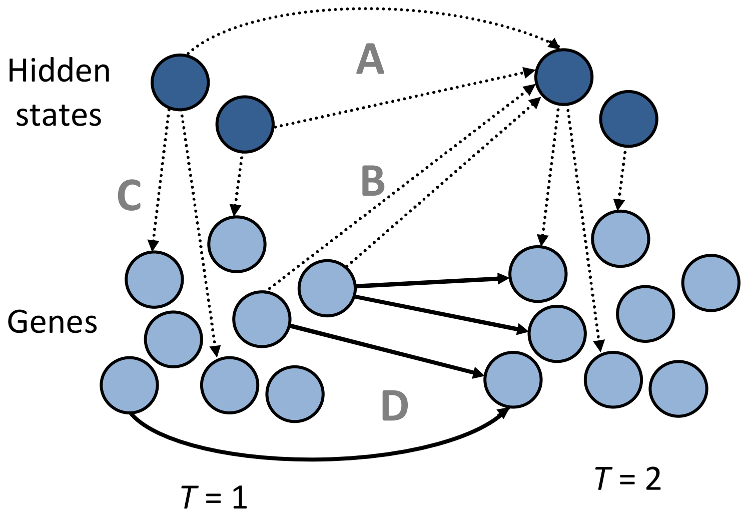

Figure 2: A visual representation of the linear feedback state space model, with the observed expression of a set of genes (light blue nodes) and the unobserved expression of a set of hidden states (dark blue nodes) at two time points, $T = 1$ and $T = 2$, where A, B, C, and D correspond to the matrices in Equation (1). The solid arrows, representing the nonzero elements of D, correspond to the direct gene-gene interactions that make up the gene regulatory network.

2.2 State Space Models

A special case of the DBN is the state space model (SSM), also referred to as a linear dynamical system (LDS). SSM make use of a set of continuous noisy measurements of observed variables and continuous unobserved hidden states. Under the SSM framework, a pair of linear equations, known as the state and dynamic equations, is used to relate the expression of genes and a set of hidden states from one time point to the next. In general, these equations can be time-variant or nonlinear, but in this work we restrict our attention to the linear, time-invariant model.

2.2.1 Application to Gene Regulatory Networks

Consider time-course gene expression data with P genes, K hidden states, T time points, and R biological replicates. Let $\mathbf{x_{tr}} = \{x_{tr1}, ..., x_{trK}\}$ and $\mathbf{y_{tr}} = \{y_{tr1}, ..., y_{trP}\}$ represent the expression of the sets of hidden states and genes, respectively, in replicate r at time t. The state and dynamic equations for the

DOI: 10.2202/1544-6115.1513

SSM are

$$\begin{aligned} \mathbf{x_{tr}} &= A\mathbf{x_{t-1,r}} + B\mathbf{y_{t-1,r}} + \mathbf{w_{tr}} \\ \mathbf{y_{tr}} &= C\mathbf{x_{t,r}} + D\mathbf{y_{t-1,r}} + \mathbf{z_{tr}} \end{aligned} \tag{1}$$

or in matrix notation,

$$\left[\begin{array}{c} \mathbf{x_{tr}} \\ \hdashline \mathbf{y_{tr}} \end{array}\right] = \left[\begin{array}{c:c:c} A & \mathbf{0} & B \\ \hdashline \mathbf{0} & C & D \end{array}\right] \left[\begin{array}{c} \mathbf{x_{t-1,r}} \\ \hdashline \mathbf{x_{tr}} \\ \hdashline \mathbf{y_{t-1,r}} \end{array}\right] + \left[\begin{array}{c} \mathbf{w_{tr}} \\ \hdashline \mathbf{z_{tr}} \end{array}\right] \tag{2}$$

where $\mathbf{w_{tr}} \sim MVN(\mathbf{0}, I_{K\times K})$, $\mathbf{z_{tr}} \sim MVN(\mathbf{0}, diag(\mathbf{v})^{-1})$, $\mathbf{v}$ is a P-dimensional vector of gene precisions, A is the $(K \times K)$ state dynamics matrix, B is the $(K \times P)$ observation-to-state matrix, C is the $(P \times K)$ observation matrix, D is the $(P \times P)$ observation-to-observation matrix, $t = 1, ..., T$, and $r = 1, ..., R$ (see Figure 2 for a visual representation of this model).

Note that in general, state space models are unidentifiable, as the hidden state can be re-scaled and the matrices of the state and dynamic equations adapted accordingly. This implies that two models can have equivalent gene-gene interactions, but different values for the hidden variables. However, since the matrices D and $CB + D$ are sub-identifiable (see Rangel et al., 2004), inference on the structure of the network based on these matrices is possible. In this work, we focus our attention on the D matrix, which corresponds to the structure of the direct gene-to-gene interaction matrix.

3 Algorithm

We describe in detail the Empirical Bayes Dynamic Bayesian Network (EBDBN) algorithm, which is composed of three principal parts: model selection (choice of K), estimation of hidden states, and calculation of posterior distributions. The method is implemented in an R and C script, which has been made available in the R package `ebdbNet` available on CRAN (R Development Core Team, 2009).

3.1 Model Selection

The choice of the optimal dimension K of the hidden states is a difficult problem, but crucial to the application of state space models to network structure recovery. Commonly used criteria for model selection include Akaike's Information Criterion (AIC) (Akaike, 1969) and the Bayesian Information Criterion (BIC) (Schwarz, 1978). Unfortunately, both of these criteria tend to perform poorly for microarray data due to the large number of observations and model parameters. Following Bremer (2006) and Aoki and Havenner (1991), we apply a time series method for model selection, based on the autocovariances between observations. This technique shortens computation time considerably, as the algorithm does not run over a wide range of values for K.

Specifically, we construct a block-Hankel matrix of autocovariances of the time series gene expression observations

$$H = \begin{pmatrix} \hat{\Gamma}_1 & \hat{\Gamma}_2 & \cdots & \hat{\Gamma}_m \\ \hat{\Gamma}_2 & \hat{\Gamma}_3 & \cdots & \hat{\Gamma}_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Gamma}_m & \hat{\Gamma}_{m+1} & \cdots & \hat{\Gamma}_{2m-1} \end{pmatrix} \tag{3}$$

where $\hat{\Gamma}_i = \frac{1}{T}\sum_{t=1}^{T-i} \mathbf{y_t}\mathbf{y_{t+i}}'$ is the autocovariance matrix of the observations $\mathbf{y}$ at time lag i, and m represents the maximum relevant biological time lag between a gene and its regulators. In other words, m is the number of forward time units that a gene is able to influence the expression of other genes. This value must be pre-specified depending on the data under consideration, but in microarray experiments this value is typically small ($m =$ 1, 2, or 3) and depends on the biological process being studied and the time lag between consecutive measurements.

In the absence of error, the rank of $\mathbf{H}$ equals the number of hidden states K needed to characterize the time series (Aoki and Havenner, 1991). However, microarray data contain both biological and technical errors, meaning the rank of $\mathbf{H}$ is not exactly equal to K. As such, after taking the singular value decomposition (SVD) of $\mathbf{H}$, there will be K singular values of "large" amplitude (as discussed below), provided the signal-to-noise ratio (SNR) is also large (SNR $\gg 1$).

The SVD for $\mathbf{H}$ is $\mathbf{H} = \mathbf{USV}'$, where $\mathbf{S}$ is a diagonal matrix with diagonal entries $\lambda_1, ..., \lambda_{mP}$ ordered by size, such that $\lambda_1 > ... > \lambda_{mP}$. We scale these singular values by the value of the largest singular value, such that $1 > \frac{\lambda_2}{\lambda_1} > ... > \frac{\lambda_{mP}}{\lambda_1}$. Note that if there are T time points in a particular microarray dataset, only the first $T - 1$ singular values will be non-zero.

When plotting the values of these scaled singular values, we typically note a rapidly decreasing value for the first singular values, followed by a more moderate decrease; intuitively, the SVD reduces H to a small set of singular values which still contain a large fraction of the original variability. Choosing the number of large singular values results in finding the point at which the inclusion of an additional singular value does not increase the amount of explained variation enough to justify its inclusion (this is similar to choosing the number of components in a Principal Components Analysis). This “optimal” value of K could be chosen by finding the “elbow” of a plot of the singular values (similar to a scree plot), by using the Eigenvalue-One criterion of Kaiser (1960), or by fixing a cutoff based on percent of total variance explained by the singular values. Here, we use the latter criterion based on the somewhat arbitrary cutoff of 90% of total variance explained, as this cutoff was found to accurately identify the correct hidden state dimension K in simulated data.

3.2 Estimation of Hidden States

When A, B, C, D, and $\mathbf{v}$ are known, the Kalman filter and smoother (Kalman, 1960) may be used to estimate the values of the hidden states of a given dimension K. The Kalman filter and smoother are essentially a set of recursive calculations, where the former consists of a prediction and update step, and the latter smooths the filtered estimations using the full dataset (see Bremer, 2006, for additional details about the use of the Kalman filter and smoother in the context of gene expression data). In the filter step,

$$\begin{aligned} \hat{\mathbf{x}}_\mathbf{t}^- &= A\hat{\mathbf{x}}_\mathbf{t-1} + B\mathbf{y}_\mathbf{t-1} \\ \hat{\mathbf{x}}_\mathbf{t} &= \hat{\mathbf{x}}_\mathbf{t}^- + K(\mathbf{y}_\mathbf{t} - C\hat{\mathbf{x}}_\mathbf{t}^- - D\mathbf{y}_\mathbf{t-1}) \end{aligned} \tag{4}$$

where $\mathbf{y}_\mathbf{t}$ and $\mathbf{x}_\mathbf{t}$ are the observations and hidden state values, respectively, at time t, $\hat{\mathbf{x}}_\mathbf{t}$ represents the filtered estimate of $\mathbf{x}_\mathbf{t}$ at time t, $\hat{\mathbf{x}}_\mathbf{t}^-$ represents the *a priori* estimate of $\mathbf{x}_\mathbf{t}$ based on the previous time step, A, B, C, and D are as in (1), and K is the Kalman gain matrix. Then, in the smoothing step,

$$\hat{\mathbf{x}}_\mathbf{t}^T = \hat{\mathbf{x}}_\mathbf{t} + J(\hat{\mathbf{x}}_\mathbf{t+1}^T - A\hat{\mathbf{x}}_\mathbf{t} - B\mathbf{y}_\mathbf{t}) \tag{5}$$

where $\hat{\mathbf{x}}_\mathbf{t}^T$ represents the smoothed estimate of $\mathbf{x}_\mathbf{t}$ at time t, J is the Kalman smoothing matrix, and $\hat{\mathbf{x}}_\mathbf{t}$, $\mathbf{y}_\mathbf{t}$, A, and B are as before. In its implementation in the EBDBN algorithm, the posterior means $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{D}$ are used in the Kalman filter and smoother equations. Both the Kalman gain matrix K and smoothing matrix J are calculated using the standard formulas (see Kalman, 1960, for more details).

3.3 Calculation of Posterior Distributions

For the estimation of the model parameters, we implement the same hierarchical Bayesian structure as Beal et al. (2005). Let $\mathbf{a}_{(j)}$, $\mathbf{b}_{(j)}$, $\mathbf{c}_{(j)}$, and $\mathbf{d}_{(j)}$ denote vectors made up of the jth rows of matrices A, B, C, and D, respectively. Then

$$\begin{aligned}\mathbf{a}_{(j)}|\alpha &\sim N(\mathbf{0}, diag(\alpha)^{-1}) \\ \mathbf{b}_{(j)}|\beta &\sim N(\mathbf{0}, diag(\beta)^{-1}) \\ \mathbf{c}_{(i)}|\gamma, v_i &\sim N(\mathbf{0}, v_i^{-1} diag(\gamma)^{-1}) \\ \mathbf{d}_{(i)}|\delta, v_i &\sim N(\mathbf{0}, v_i^{-1} diag(\delta)^{-1})\end{aligned} \tag{6}$$

where $\alpha = \{\alpha_1, ..., \alpha_K\}$, $\beta = \{\beta_1, ..., \beta_P\}$, $\gamma = \{\gamma_1, ..., \gamma_K\}$, $\delta = \{\delta_1, ..., \delta_P\}$, v_i is the ith component of vector $\mathbf{v}$, $j = 1, ..., K$ and $i = 1, ..., P$. Thus, we have a set of parameters $\theta = \{A, B, C, D, \mathbf{v}\}$ and a set of hyperparameters $\psi = \{\alpha, \beta, \gamma, \delta\}$ describing the a priori precisions of the parameter set.

Because the SSM is in the exponential family, the EM algorithm (see Dempster et al., 1977; Bilmes, 1997) can be used to find a point estimate $\hat{\psi}$ of the hyperparameters ψ, conditioned on the current estimates $\hat{\mathbf{x}}$ of the hidden states. The posterior means of A, B, C, and D, given the point estimates $\hat{\psi}$, can subsequently be calculated.

3.3.1 Implementation in EBDBN

In implementing the EBDBN algorithm to estimate ψ, we apply the EM algorithm to stabilize the hyperparameter estimates, followed by an estimation of the gene precisions $\mathbf{v}$ and a subsequent fine-tuning run of the EM algorithm (see Figure 3). Because of our unique use of the EM algorithm, we refer to this portion of the algorithm as an "EM-like algorithm." Let the current values of the hyperparameters, gene precisions, and hidden states be $\psi^{(i)}$, $\mathbf{v}^{(i)}$, and $\mathbf{x}^{(i)}$, respectively. We proceed as follows:

1. Run the EM algorithm with $\mathbf{x}^{(i)}$, holding $\mathbf{v}^{(i)}$ constant. When convergence is reached (using stopping criterion Δ_1), set the hyperparameter estimate at $\tilde{\psi}^{(i+1)}$.

2. Calculate $\hat{\mathbf{v}}^{(i+1)}$, the innovation precisions:

$$\hat{\mathbf{v}}^{(i+1)} = \left(\sum_{r=1}^{R} \sum_{t=1}^{T} (\mathbf{y_{tr}} - \hat{C}\mathbf{x_{tr}}^{(i)} - \hat{D}\mathbf{y_{t-1,r}})^2 / (RT - 1) \right)^{-1} \tag{7}$$

   where $\hat{C}$ and $\hat{D}$ are the posterior means of C and D, given $\tilde{\psi}^{(i+1)}$ and $\mathbf{x}^{(i)}$

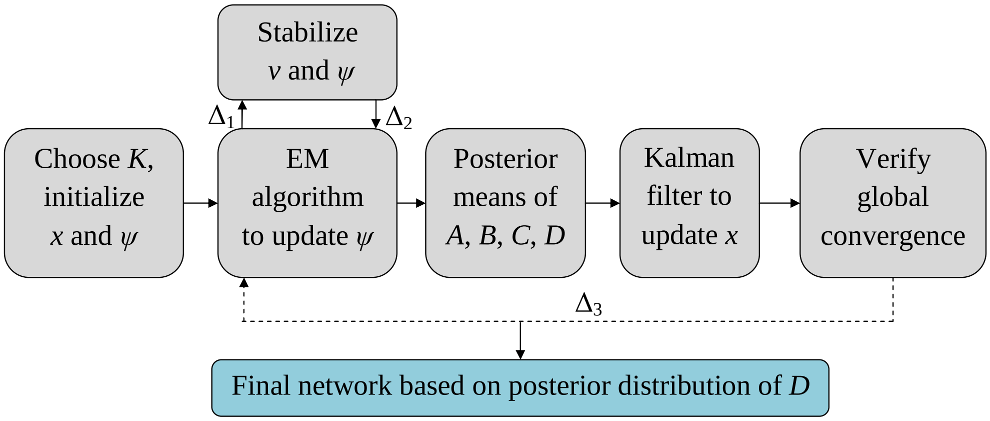
Stabilize
v and *ψ*
Δ_1
Δ_2
Choose *K*, initialize *x* and *ψ*
EM algorithm to update *ψ*
Posterior means of *A*, *B*, *C*, *D*
Kalman filter to update *x*
Verify global convergence
Δ_3
Final network based on posterior distribution of *D*

Figure 3: Visual representation of EBDBN algorithm. The algorithm starts by choosing K, as described in Section 3.1, and initializing hidden states $\mathbf{x}$ and hyperparameters ψ. Next, the EM-like algorithm described in Section 3.3 is used to update ψ, with a sub-loob to stabilize gene precisions $\mathbf{v}$ and convergence criteria Δ_1 and Δ_2 as noted in Section 3.3.1. The posterior means of matrices A, B, C, and D are then computed given the current estimates $\hat{\psi}$ and $\hat{\mathbf{x}}$. Using these posterior means, the Kalman filter and smoother described in Section 3.2 are used to update the hidden states, and global convergence of parameters is checked using criterion Δ_3. If convergence is met, the network topology is inferred based on the posterior distribution of D; otherwise, the algorithm returns to the EM-like algorithm step.

3. Run the EM algorithm to fine-tune the hyperparameter estimates with $\mathbf{x}^{(i)}$, holding $\hat{\mathbf{v}}^{(i+1)}$ constant. When convergence is reached (using stopping criterion Δ_2), set the final hyperparameter estimate at $\hat{\psi}^{(i+1)}$

3.4 Empirical Bayes Dynamic Bayesian Network method

We discuss the importance of different convergence criteria that are used throughout the EBDBN algorithm (Figure 3), as well as the procedure for choosing "significant" edges once the algorithm has terminated.

3.4.1 Convergence Criteria

There are three convergence criteria used in the implementation of the EBDBN algorithm. The first two, Δ_1 and Δ_2, are used to determine convergence of the initial and fine-tuning runs of the EM algorithm in the estimation of ψ, as noted in Section 3.3.1. The third, Δ_3, is used to determine global convergence of the EBDBN algorithm. In general, for parameter values at the i^{th} and $(i+1)^{\text{th}}$ iteration, these convergence criteria are a distance metric of the form

$$D = \max_{\psi=\{\alpha,\beta,\gamma,\delta\}} \sqrt{\frac{\sum(\psi^{(i+1)} - \psi^{(i)})^2}{\sum(\psi^{(i)})^2}} \tag{8}$$

corresponding to the maximum of these distances for $\psi = \{\alpha, \beta, \gamma, \delta\}$. That is, when all of the values ψ change very little from one iteration to the next (whether within a run of the EM algorithm or in the larger loop of the full algorithm), approximate convergence is declared and estimation is considered to be complete.

Unfortunately, determining the convergence properties of the algorithm is difficult since simulating plausible data based on a state space model with arbitrary fixed hyperparameter values (i.e., the precisions of the state space matrices A, B, C, and D) rather than fixed parameter values (i.e., the state space matrices themselves) is difficult. Consequently, it is unclear how to ascertain whether convergence can always be attained. Preliminary, smaller-scale simulations indicated relatively good convergence behavior, and thresholds chosen based on these early simulations ($\Delta_1 = 0.15$, $\Delta_2 = 0.05$, and $\Delta_3 = 0.05$) seemed to perform well in later simulations.

3.4.2 Edge Selection

Since the posterior distributions of the elements of D (the matrix representing direct gene-to-gene interactions) are Gaussian, we can compute the standard z-statistic for normally distributed variables for each edge. Edges whose distributions lie far above the zero point are interpreted as activations, while those far below the zero point as inhibitions. Consequently, to decide which edges are present in the network, we can use the standard thresholds $(1.96, 2.58, 3.30)$ for $(95, 99, 99.9)$% confidences, respectively.

4 Simulations

The importance of simulation studies for methods of network inference has been stressed by several authors, including Husmeier (2003), Rogers and Girolami (2005), and Rice et al. (2005). In the following simulations, we consider a wide range of comparison criteria, including the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve (AUCROC), the sensitivity, specificity, and positive predictive value at a given threshold, and the computational time. Network inference methods that perform well on one of these criteria often underperform in others (Wessels et al., 2001); consequently, we consider a more global perspective on performance in order to thoroughly understand the advantages and limitations of each method.

Further, we compare the performance of several similar methods on simulated data. In these simulations, the EBDBN method outlined in Section 3 is compared to two other pre-established methods developed to infer gene networks from temporal gene expression data: the Variational Bayes State Space Model (VBSSM) of Beal et al. (2005) and the Vector Auto-Regressive model (VAR) of Opgen-Rhein and Strimmer (2007). The former incorporates hidden states in a state space model under a Bayesian framework, while the latter uses an auto-regressive model without hidden states. In addition, we also consider the EBDBN method where no hidden states are estimated, denoted EBDBN(-); this amounts to a simple auto-regressive model (similar to the VAR) where D is estimated with the EM-like algorithm of Section 3.3, and no Kalman filter/smoother step is applied. We refer to the EBDBN method with hidden states incorporated as EBDBN(x) in general, and EBDBN($K = k$) for a particular hidden state dimension k.

For these simulations, let TP denote the number of true positives, FP represent the number of false positives, and FN and TN denote the number of false and true negatives, respectively. We define the following three criteria as follows: a) Sensitivity = TP/(TP + FN), b) Specificity = TN/(TN + FP), and c) Positive Predictive Value (Precision) = TP/(TP + FP).

4.1 Model-Based Simulations

We first simulate gene expression data based on an autoregressive model (which is the same model for all of the methods under consideration). While this model undoubtedly oversimplifies the dynamics of real microarray data, it is a useful starting point that allows for straightforward simulation with a known network structure.

To simulate data, we choose P = 53 genes, and vary both the number of replicates R and the number of time points T to be $\{5, 10, 15\}$. The $P \times P$ matrix D, representing the direct gene-gene interaction matrix, is simulated such that 10% of its elements follow either a $U(-1, -0.2)$ or $U(0.2, 1)$ distribution and

90% of its elements equal 0. We initialize the gene expression values at time $t = 1$ to be $\mathbf{y_t} \sim N(0, 0.1)$ and simulate subsequent times $t = 2, ..., T$ following an AR(1) model, such that $\mathbf{y_t} \sim N(D\mathbf{y_{t-1}}, 0.1)$. We then remove the 3 genes with the most outward connections, denoted $\{\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \tilde{\mathbf{y}}_3\}$, from the dataset. This set of genes play the role of the $K = 3$ hidden states. The goal in these simulations is to infer the remaining structure of the gene network, $D^\star = D \setminus \{\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \tilde{\mathbf{y}}_3\}$ using only the 50 genes in the "observed" data $\mathbf{y}^\star = \mathbf{y} \setminus \{\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \tilde{\mathbf{y}}_3\}$.

First, consider the results for the AUCROC, which can be interpreted as the average probability that a randomly chosen edge is correctly characterized as present or not, assuming any given edge is *equally likely* to be present or not (Beal et al., 2005). The simulation results are presented in Figure 4. Several trends are apparent: first, for small datasets (that is, few replicates and time points), all methods perform poorly. For the EBDBN(x), EBDBN(-), and VBSSM, increasing either R, T, or both improves performance; interestingly, for the VAR method this is true when R is increased while the inverse holds for increasing T. Furthermore, the VBSSM outperforms all methods considered here, although the margin of difference decreases considerably for larger datasets. Comparing the EBDBN(x) and EBDBN(-) methods, the results indicate that the latter outperforms the former only for data with few time points (T = 5).

To obtain a more complete perspective on the performance of the models, we also consider the sensitivity, specificity, and positive predictive value for given threshold values for the z-score, as well as the computational time required, shown in Figure 5. While the EBDBN(x), EBDBN(-), and VBSSM all seem to be comparable in terms of sensitivity and AUC, the VBSSM and VAR methods perform best in terms of specificity and positive predictive value. For the model-based simulations, the VBSSM and VAR are more stringent in edge selection, resulting in a lower rate of *false* positives (although in the case of the VAR, this also results in a much lower rate of *true* positives). The calculation time for each method is also worth considering. The VBSSM required about 12 minutes for algorithm convergence, while for the EBDBN(x), EBDBN(-), and VAR this time was about 5 1/2 minutes, 2 seconds, and 1 second, respectively. All simulations were run on a dual-processor Dell PowerEdge 1850 (quad-core 2.8 GHz Intel (R) Xeon (TM)) with 12GB RAM, running Red Hat Enterprise Linux 5.3 Server x86_64.

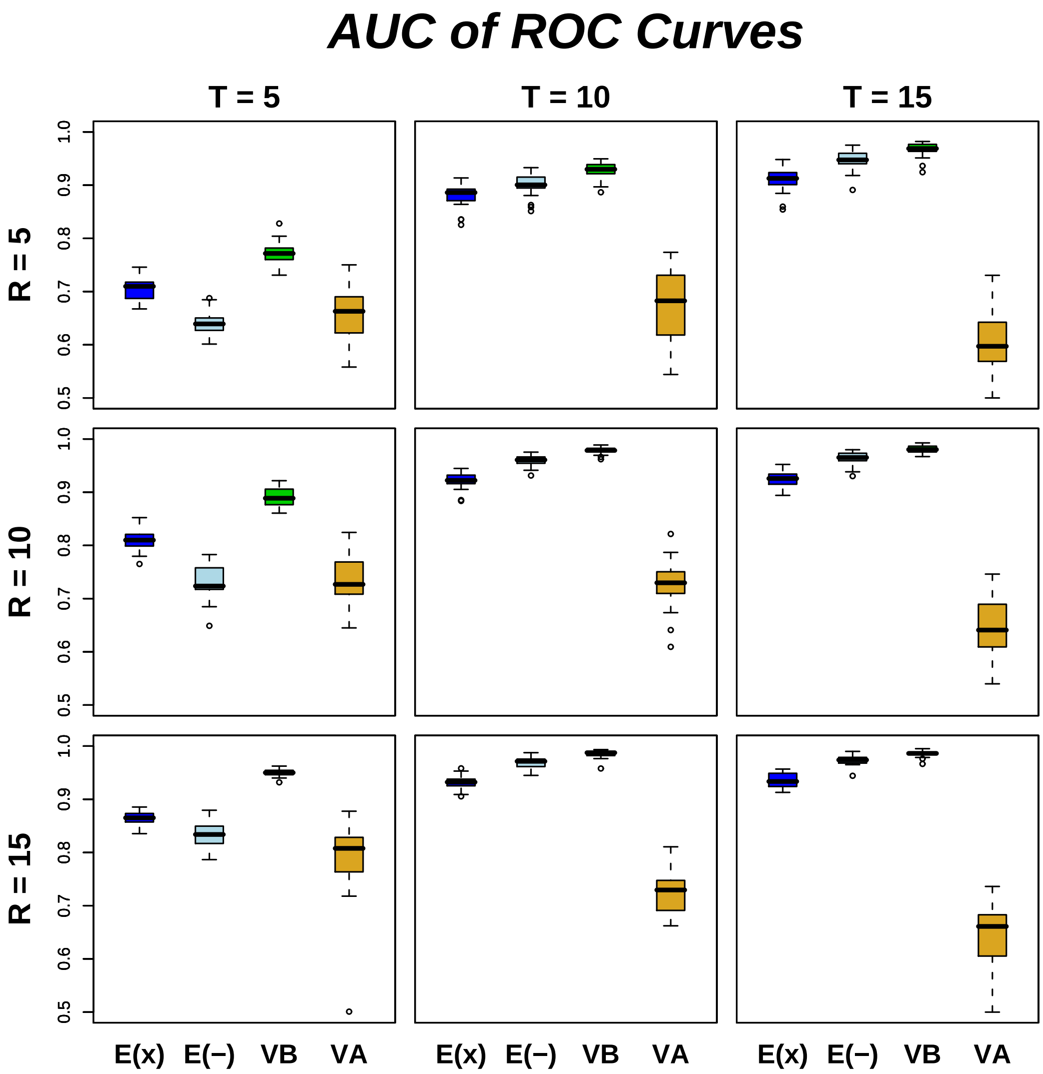

Figure 4: AUC of the ROC curve of model-based simulations, by number of replicates and number of time points simulated in the data. Each row of the graphical matrix corresponds to the number of replicates (R = 5, 10, 15) and each column to the number of time points (T = 5, 10, 15), with 25 datasets simulated per evaluation. Within each individual plot, the methods represented (from left to right) are as follows: E(x) = EBDBN method with hidden states (dark blue), E(-) = EBDBN method without hidden states (light blue), VB = Variational Bayes method (green), and VA = VAR method (yellow).

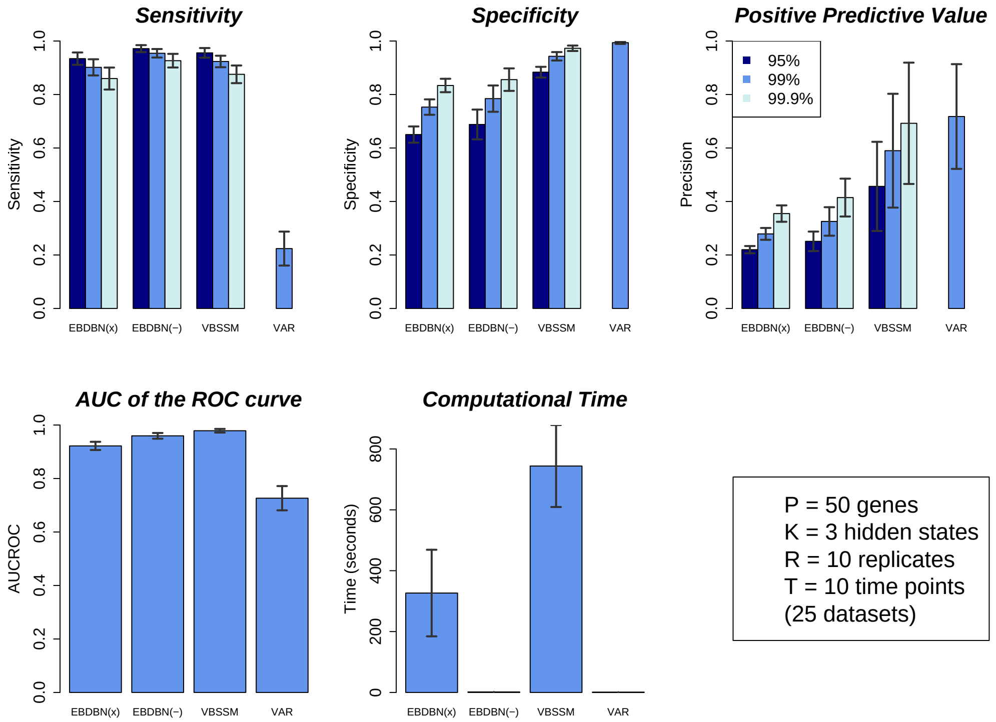

Figure 5: Additional comparison criteria for model-based simulations, for data with $R = 10$ replicates and $T = 10$ time points for all methods, using a cutoff for the z-scores of $\{95, 99, 99.9\}$% for the EBDBN(x), EBDBN(-), and VBSSM. A cutoff of 80% is used for the local false discovery rate in the VAR method, as suggested by Opgen-Rhein and Strimmer (2007).

4.2 Data-Based Simulations

One of the challenges of relying on simulations to compare different methods for network inference is that results based on data simulated under the same model as a particular method (as is the case for all the methods under consideration in Section 4.1) are undoubtedly over-optimistic. For this reason, it is also important to consider "realistic" simulations of well-known systems based on ordinary differential equations, which more closely approximate real microarray datasets. One such set of simulations by Zak et al. (2001, 2003) appears to be particularly promising, as it has already been used for this purpose by several other authors (including Husmeier, 2003; Beal et al., 2005).

The Zak et al. (2001, 2003) data include 10 genes that interact with one another via a set of transcription factor proteins, bound promoters, and a ligand

DOI: 10.2202/1544-6115.1513

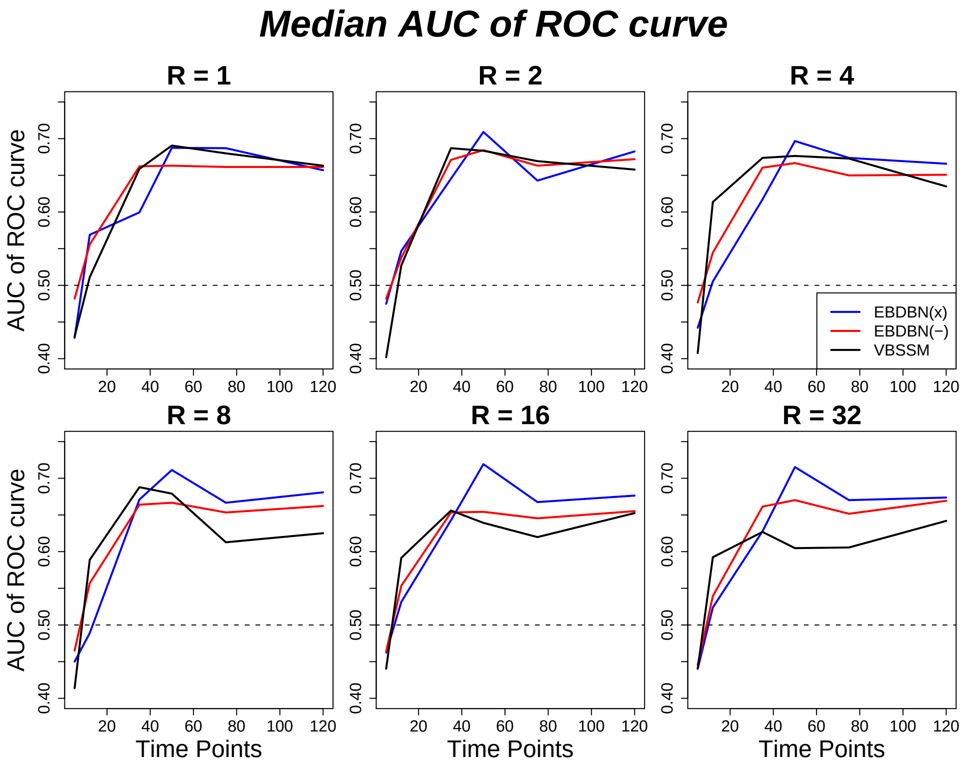

Figure 6: Median AUC of the ROC curve for the EBDBN(x), EBDBN(-), and VBSSM, by number of replicates R and time points T. The horizontal dotted line in each graph represents an AUC of 0.5, corresponding to a random-guess classifier.

input, organized into regulatory motifs taken from the biological literature. A set of *in silico* simulations in Matlab, carried out by integrating the ordinary differential equations describing the model, yields a dataset of 55 variables over 500 time points. We follow Beal et al. (2005) and use only the observations for the 10 genes for network inference (thus leaving 45 variables as “hidden states”). We construct datasets of length $T = \{5, 12, 35, 50, 75, 120\}$ by subsampling equally spaced time points, and we artificially create replicates of size $R = \{1, 2, 4, 8, 16, 32\}$ by adding Gaussian noise at each time point to the log-ratios of the observed gene expression data. The VAR method is not included in the results below, as the small number of genes (10) in the dataset did not allow for accurate estimation of the local fdr needed for edge selection.

The results of the median AUCROC for the EBDBN(x), EBDBN(-), and

VBSSM are shown in Figure 6. In these figures, we see that data simulated with only 5 time points yield poor performance for all methods under consideration; this is alleviated by increasing the number of time points, although the improvement seems to plateau at around 50 or 75 time points. This is likely due to the artificial nature of the time point subsampling; that is, the same interval of time is covered by data sampled with 75 and 120 time points, albeit in greater detail in the latter case. The benefits of ever-finer sampling are thus eventually overrun by the additional noise incurred by the extra data points.

Unsurprisingly, a similar phenomenon is observed with the number of replicates, which were also artificially created by adding Gaussian noise. As with the subsampled time points, the creation of additional artificial replicates tends to add noise to the data without contributing additional information about the network dynamics. For data with 4 or more replicates, the VBSSM appears to outperform the EBDBN, but only for datasets with less than 35 time points; this relationship is reversed for a larger number of time points. Finally, note that the AUCROC values observed in Figure 6, on the order of 45% to 70%, are much lower than those observed in Figure 4, which were on the order of 60% to 98%. This is undoubtedly due to the more realistic, and thus more complicated, biological relationships simulated here.

We also consider the performance of the EBDBN and VBSSM on additional comparison criteria, shown in Figure 7. Here, the EBDBN(x), EBDBN(-), and VBSSM perform comparably in terms of sensitivity and positive predictive value, while the EBDBN(x) and EBDBN(-) outperform the VBSSM in terms of specificity and AUC of the ROC curve. There is also a considerable difference in computational time, as the VBSSM required about 26 minutes on average to converge, while the EBDBN(x) and EBDBN(-) required only a little over 1 minute and 1 second, respectively.

5 Application

We compare the results of the EBDBN(x), EBDBN(-), VBSSM, and VAR methods using real microarray data. Although several time-series microarray datasets are publicly available in organisms such as *Saccharomyces cerevisiae* (Lee et al., 2004), few microarray experiments include the necessary level of replication and time points to effectively infer network structure. One exception arises from a study of T-cell activation, with gene expression measured on 58 genes over 10 time points on 44 replicates (see Rangel et al., 2004). Genes were pre-selected for modulation following the activation, meaning that genes with

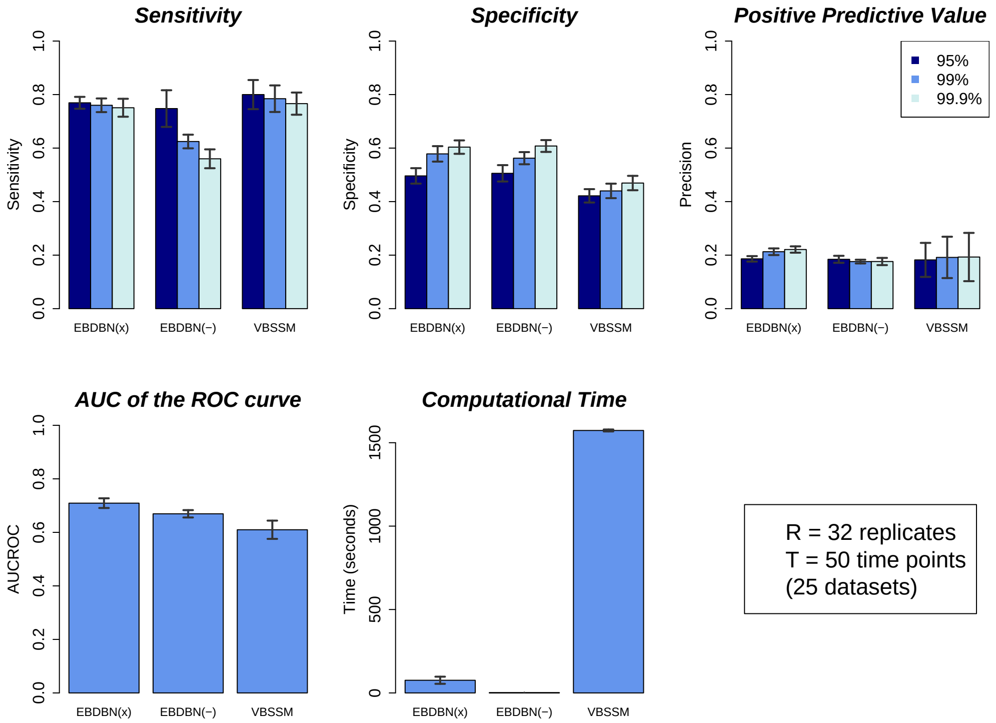

Figure 7: Additional comparison criteria of data-based simulations, for data with $R = 32$ replicates and $T = 50$ time points, using a cutoff for the z-scores of $\{95, 99, 99.9\}$% for the EBDBN(x), EBDBN(-), and VBSSM.

weak expression or low reproducibility across replicates were removed from the dataset. The data were log-transformed and quantile normalized in a preprocessing step. The expression measurements and gene descriptions may be found in the R package `GeneNet` of Schäfer et al. (2006) on CRAN (R Development Core Team, 2009).

In previous work, the hidden state dimension was chosen to be K = 9 and K = 14 (Rangel et al., 2004; Beal et al., 2005). Recall that the maximum number of nonzero singular values taken from the decomposition of the block-Hankel matrix $\mathbf{H}$ is $T - 1$, or in this case, 9. After applying the decomposition of $\mathbf{H}$, the hidden state dimension is chosen to be K = 4. We apply the EBDBN($K = 4$) and EBDBN(-) methods for 10 different sets of initial values for hyperparameters, hidden states, and gene variances. A 99.9% cutoff is used as a threshold for the z-scores of the edges, and only edges identified in 80% of the runs are retained for the final network structure. For the VBSSM and VAR methods,

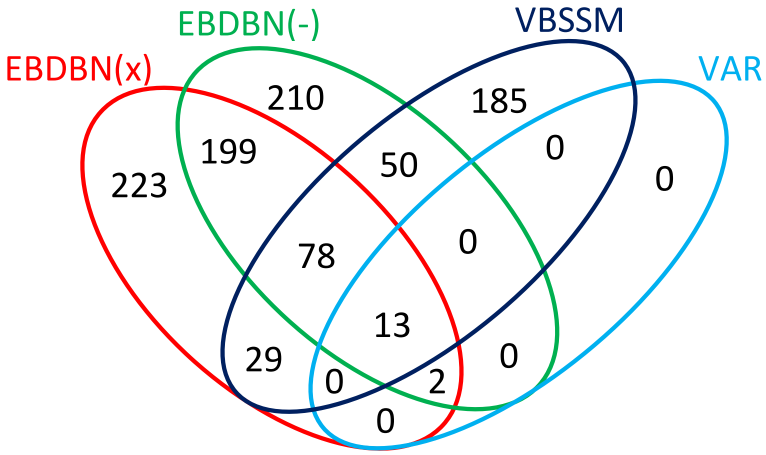

Figure 8: Four-way Venn Diagram of total edges identified by the EBDBN(x), EBDBN(-), VBSSM, and VAR.

Table 1: Positive, negative, and total edges found in T-cell data.

Method	# Positive	# Negative	Total Edges (%)
EBDBN(x)	435	109	544 (16.2)
EBDBN(-)	338	214	552 (16.4)
VBSSM	233	122	355 (10.6)
VAR	9	6	15 (0.4)

cutoff values of 99.9% for the z-scores and 80% for the local fdr correction are used, respectively.

The effect of each gene-gene interaction is determined by the sign of the significant elements of the D matrix from Equation (1). That is, activatory relationships are represented by positive elements of D, and inhibitory relationships by negative elements. The number of edges (positive, negative, and total) found by each method is displayed in Table 1. The overlap of significant edges is displayed in a four-way Venn diagram in Figure 8. Only 13 edges are selected by all four methods under consideration, largely due to the small number (15) of edges identified by the VAR method. There are 93 edges identified by at least three of the methods, and 371 by at least two of the methods. Several edges are identified by only a single method; there are 223, 210 and 185 such identified edges for the EBDBN($K = 4$), EBDBN(-), and VBSSM, respectively.

The sub-network identified by both the EBDBN($K = 4$) and VBSSM is

DOI: 10.2202/1544-6115.1513

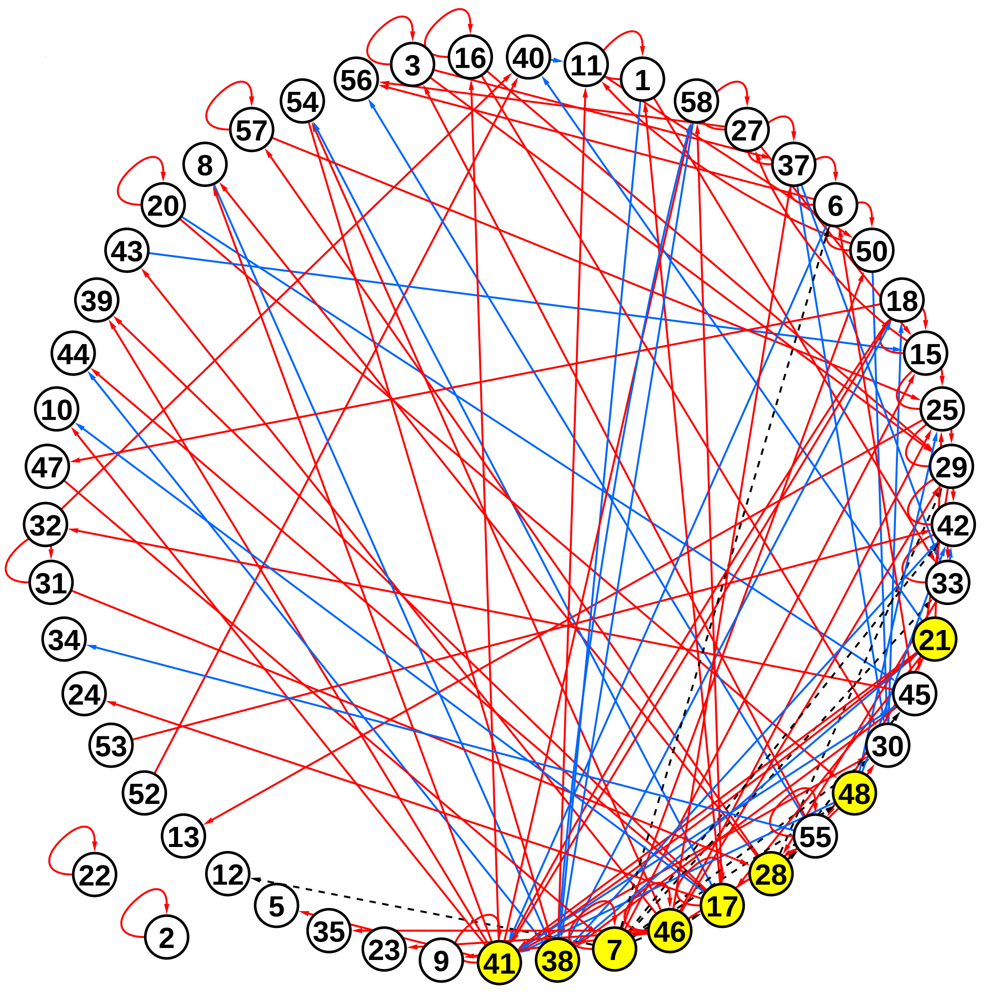

Figure 9: Edges inferred by both the EBDBN($K = 4$) and VBSSM methods for the T-cell activation data (Rangel et al., 2004). Nodes represent genes, blue solid lines inhibitory regulations, red solid lines activatory regulations, and black dotted lines edges with ambiguous regulation. Yellow nodes have five or more regulatory interactions with other genes, indicating important genes in the network topology.

Table 2: Description of important genes in network topology.

Gene #	Description
7	CD69 antigen (p60, early T-cell activation antigen)
17	Survival of motor neuron 1, telomeric
21	Cyclin C
28	Early growth response 1
38	Caspase 4, apoptosis-related cysteine protease
41	GATA-binding protein 3
46	Interleukin 2 receptor, gamma (severe combined immunodeficiency)
48	Myeloperoxidase

made up of 120 edges involving 51 genes. Of these edges, 26 represent inhibitions, 84 represent activations, and 10 have an ambiguous regulatory effect (i.e., the two methods do not agree with respect to the type of interaction). The structure of this “consensus network” is visualized using the software application Cytoscape (Shannon et al., 2003), shown in Figure 9. Eight genes (7, 17, 21, 28, 38, 41, 46, 48) have a high degree of connectivity with other genes, with 5 or more outward-directed edges. These genes appear to be key players in the network and could potentially be avenues of interest for future research (descriptions of these genes can be found in Table 2).

In work by Rangel et al. (2004), the gene FYB (gene 1) is found to occupy a crucial position in the graph, since it is involved in the highest number of outward connections. In the consensus network found above, although it does not figure as prominently, FYB is present and directly connected to three other genes (genes 17, 33, and 38). An interleukin receptor gene, IL-2Rγ (gene 7), is one of the most connected genes, with nine outward connections and one feedback loop with itself.

A portion of the sub-network found by Beal et al. (2005) representing the interaction between two proto-oncogenes of the Jun protein family, Jun-B (gene 54) and Jun-D (gene 11), is also found in this consensus network (see Figure 10). This is of interest, as Jun-B and Jun-D are thought to negatively regulate cell growth and inhibit programmed cell death, and thus are at the center of mechanisms controlling apoptosis and proliferation. Both sub-networks seem to support this hypothesis, as Jun-B appears to regulate the apopototic gene Caspase-4 (gene 38), and at least indirectly, Caspase-8 (gene 18). Despite their differences, both sub-networks support a central pathway in which Jun-B activates Caspase-4, which in turn activates Jun-D. These results agree with the

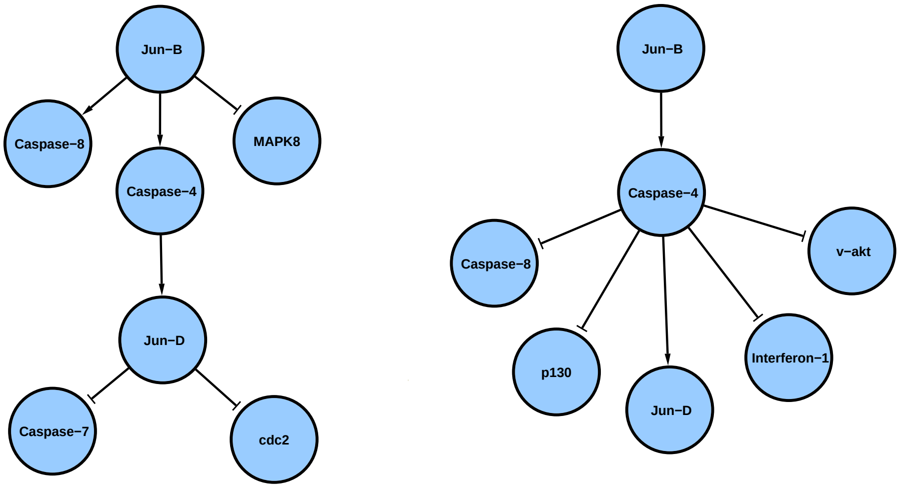

Figure 10: Subnetwork found representing the interaction between genes in the Jun protein family and genes involved in programmed cell death. (Left) Subnetwork proposed in Beal et al. (2005). (Right) Subnetwork proposed by consensus of VBSSM and EBDBN($K = 4$).

current literature, and suggest an important role for Jun-B in T-cell activation and tumor promotion (see Gurzov et al., 2008).

6 Discussion

We have presented an algorithm to infer the structure of gene regulatory networks using an empirical Bayes estimation procedure for the hyperparameters of a linear feedback state-space model. This work serves as a complement to the VBSSM of Beal et al. (2005), with the advantage of a straightforward, EM-like estimation procedure and improved computational speed. The proposed method performs comparably to previously published methods on model-based data when a minimum number of replicates (≈ 10) and time points (≈ 10) are measured.

Inferring the structure of gene regulatory networks is an intrinsically difficult task, given the complexity of network topology, the small number of replicates and time points available in real data, and the noise implicit in microarray expression measurements. As a necessary simplification, the proposed method deals solely with first-order linear dynamics in the state and observation equations of the state-space model. Although this is a good approximation to the general nature of complicated biological systems, more realistic models of gene regulatory interactions would undoubtedly capture complex relationships, such as nonlinear or higher-order interactions, more effectively. In addition, the applicability of state space models to the task of network inference remains uncertain, particularly due to the necessity of determining the hidden state dimension. In this work, we used the singular value decomposition of the block-Hankel matrix for model selection. We are aware that the procedure is ad hoc and are working on an improvement as a point of future research.

We have stressed the importance of using simulation studies to compare the performance of different network inference methods, but the current work illustrates some of the challenges inherent in doing so in this context. Model-based simulations, though an over-simplified version of the complexities of gene regulatory networks, offer greater flexibility in the choice of network parameters (e.g., number of genes, number of hidden states, percentage of edges present, amount of noise, etc.). Although the more realistic data-based simulations better represent the dynamics of gene regulatory networks, they present much smaller datasets (only 10 genes and 1 replicate) than would typically be used for network inference. As such, the development of a set of stochastic, time-course benchmark datasets for comparisons among methods is a necessary addition to the growing collection of network inference techniques.

It is somewhat surprising that although the EBDBN and VBSSM resemble each other quite closely, over 60% of the interactions identified by each one (64% for the former, 68% for the latter) are not identified by the other in the real microarray data considered here. This seems to highlight one of the major stumbling blocks for network inference methods: different methods, even those very similar to one another, often yield very different results for the same data. For this reason, we introduce the concept of a "consensus network", that is, one in which several different network inference methods are in agreement on the significance of a particular edge. Although somewhat unorthodox from a frequentist statistical point of view, this type of compromise (also referred to as "model averaging") may be more meaningful than applying a single method for network inference, at least until microarray data can be feasibly collected for a large number of replicates and time points. The further development of such "consensus networks" is a focus of our current research.

References

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics 21*, 243–247.

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Genetics Reviews 8*, 450–461.

Aoki, M. and A. Havenner (1991). State space modeling of multiple time series. *Econometric Reviews 10*(1), 1–59.

Basso, K., A. A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, and A. Califano (2005). Reverse engineering of regulatory networks in human B cells. *Nature Genetics 37*(4), 382–390.

Beal, M. J., F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild (2005). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics 21*(3), 349–356.

Beal, M. J., J. Li, Z. Ghahramani, and D. L. Wild (2005). *Introduction to Systems Biology*, Chapter ”Reconstructing transcriptional networks using gene expression profiling and Bayesian state space models”. Humana Press (Springer).

Bilmes, J. A. (1997). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and Hidden Markov models. Technical Report ICSI-TR-97-021, University of Berkeley.

Brandman, O. and T. Meyer (2008). Feedback loops shape cellular signals in space and time. *Science 322*, 390–395.

Bremer, M. and R. W. Doerge (2009). The KM-algorithm identifies regulated genes in time series expression data. *Advances in Bioinformatics (in press)*.

Bremer, M. M. (2006). *Identifying regulated genes through the correlation structure of time dependent microarray data*. Ph.D. dissertation, Purdue University, West Lafayette, IN USA.

Cao, J. and H. Zhao (2008). Estimating dynamic models for gene regulation networks. *Bioinformatics 24*(14), 1619–1624.

Dempster, A., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological) 39*(1), 1–38.

D'haeseleer, P., S. Liang, and R. Somogyi (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics 16*(8), 707–726.

Friedman, N. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology 7*(3/4), 601–620.

Gurzov, E. N., L. Bakiri, J. M. Alfaro, E. F. Wagner, and M. Izquierdo (2008). Targeting c-Jun and JunB proteins as potential anticancer cell therapy. *Oncogene 27*, 641–352.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics 19*(17), 2271–2282.

Husmeier, D., R. Dybowski, and S. Roberts (Eds.) (2005). *Probabilistic Modeling in Bioinformatics and Medical Informatics*. Springer.

Kaiser, H. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement 20*(1), 141–151.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering 82*, 35–45.

Leclerc, R. D. (2008). Survival of the sparsest: robust gene networks are parsimonious. *Molecular Systems Biology 4*(213).

Lee, I., S. V. Date, A. T. Adai, and E. M. Marcotte (2004). A probabilistic functional network of yeast genes. *Science 306*, 1555–1558.

Murphy, K. and S. Mian (1999). Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA.

Opgen-Rhein, R. and K. Strimmer (2007). Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *BMC Bioinformatics 8(Suppl 2)*.

Perrin, B.-E., L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alche Buc (2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics 19*(Suppl. 2), ii138–ii148.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rangel, C., J. Angus, Z. Ghahramani, M. Lioumi, E. Southeran, A. Gaiba, D. L. Wild, and F. Falciani (2004). Modeling T-cell activation using gene expression profiling and state-space model. *Bioinformatics 20*(9), 1361–1372.

Rice, J. J., Y. Tu, and G. Stolovitzky (2005). Reconstructing biological networks using conditional correlation analysis. *Bioinformatics 21*, 765–773.

Rogers, S. and M. Girolami (2005). A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics 21*(14), 3131–3137.

Schäfer, J., R. Opgen-Rhein, and K. Strimmer (2006). Reverse engineering genetic networks using the GeneNet package. *R News 6*, 50–53.

Schäfer, J. and K. Strimmer (2005). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics 21*(6), 754–764.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Shannon, P., A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker (2003). Cytoscape: A software environment for integrated model of biomolecular interaction networks. *Genome Research 13*, 2498–2504.

Wessels, L., E. Van Someren, and M. Reinders (2001). A comparison of genetic network models. *Pacific Symposium on Biocomputing 6*, 508–519.

Wu, F., W. Zhang, and A. Kusalik (2004). Modeling gene expression from microarray expression data with state-space equations. *Pacific Symposium on Biocomputing 9*, 581–592.

Zak, D. E., F. J. I. Doyle, G. Gonye, and J. S. Schwaber (2001). Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data. *Proceedings of the 2nd International Conference on Systems Biology*, 231–238.

Zak, D. E., G. E. Gonye, J. S. Schwaber, and F. J. Doyle (2003). Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: Insights from an identifiability analysis of an in silico network. *Genome Research 13*, 2396–2405.

Zou, M. and S. D. Conzen (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics 21*(1), 71–79.