Improved Inference of Gene Regulatory Networks through Integrated Bayesian Clustering and Dynamic Modeling of Time-Course Expression Data

Brian Godsey*

Department of Statistics and Probability Theory, Vienna University of Technology, Vienna, Austria

Abstract

Inferring gene regulatory networks from expression data is difficult, but it is common and often useful. Most network problems are under-determined–there are more parameters than data points–and therefore data or parameter set reduction is often necessary. Correlation between variables in the model also contributes to confound network coefficient inference. In this paper, we present an algorithm that uses integrated, probabilistic clustering to ease the problems of under-determination and correlated variables within a fully Bayesian framework. Specifically, ours is a dynamic Bayesian network with integrated Gaussian mixture clustering, which we fit using variational Bayesian methods. We show, using public, simulated time-course data sets from the *DREAM4 Challenge*, that our algorithm outperforms non-clustering methods in many cases (7 out of 25) with fewer samples, rarely underperforming (1 out of 25), and often selects a non-clustering model if it better describes the data. Source code (*GNU Octave*) for BAyesian Clustering Over Networks (*BACON*) and sample data are available at: http://code.google.com/p/bacon-for-genetic-networks.

Citation: Godsey B (2013) Improved Inference of Gene Regulatory Networks through Integrated Bayesian Clustering and Dynamic Modeling of Time-Course Expression Data. PLoS ONE 8(7): e68358. doi:10.1371/journal.pone.0068358

Editor: Alberto de la Fuentes, CRS4, Italy

Received March 26, 2013; **Accepted** June 3, 2013; **Published** July 23, 2013

Funding: The author has no support or funding to report.

Competing Interests: The author has declared that no competing interests exist.

* E-mail: briangodsey@gmail.com

Introduction

Inferring gene regulatory networks from high-throughput gene expression data is a difficult task, in particular because of the high number of genes relative to the number of data points, and also because of the random noise that is present in measurement. Over the last several years, many new methods have been developed to address this problem; a nice review of these can be found in [1]. This review directly compares several different types of approaches by summarizing the correctness of the genetic networks inferred from synthetic (*in silico*) data generated from a known network. Of particular interest are the results of each of the algorithms when applied to the *DREAM4 In Silico Network Challenge* data sets, which includes data types such as “knock-out”, “knock-down”, and time-series data among the sub-challenges. See [2] for more details on the *DREAM* challenges.

Though [3] have had success combining methods in order to infer genetic networks from different types of data simultaneously, here we focus on time-series data and the corresponding methods for network inference. In the review of [7], two types of algorithms seem to outperform the others when applied to time-series data: dynamic Bayesian networks and causal structure identification (CSI) in non-linear dynamical systems (NDSs).

Dynamic Bayesian networks (DBNs) are typically some variation of the basic linear model

$$x_{t+1} = Ax_t + \epsilon_1 \tag{1}$$

$$y_t = x_t + \epsilon_2 \tag{2}$$

where in the context of gene regulatory networks, x_t is the vector of “true” gene expression levels at time t, y_t is a vector of observations of these expression levels, A is a matrix of interaction coefficients, and ϵ_1 and ϵ_2 are random (Gaussian) noise. More information on DBNs and their application to gene regulatory networks can be found in [4] and [5].

The algorithms considered in [1] include a model very similar to that of the basic DBN formulation above, but which exploits conditional [first-order] dependence within nodes of the network, as well as an assumption of relative sparseness, to efficiently infer network structure. This model, from [6] is referred to as *G1DBN* and is available as an *R* package from *CRAN* [7]. The second DBN considered by [1] is that of [8], which adapts a state-space model with inputs to include hidden states, the quantity and values of which are inferred through variational Bayesian learning. This algorithm is referred to as *VBSSM*, as in the review. Causal structure identification (CSI) in non-linear dynamical systems (NDSs) avoids the restriction of linearity when determining network structure, and in the case of [3], which is also considered in the review, latent interaction parameters of a discrete Gaussian process model are inferred using a Bayesian framework. According to [1], both the *G1DBN* and *VBSSM* algorithms performed well on the *DREAM4* data sets, as did the CSI algorithm of [9]. Both DBNs and CSI outperformed ordinary differential equations (ODEs) and models using Granger causality.

Though these results are convincing, there is still room for improvement, and the discussion of optimal methods is still open; in fact, the body of research in the area of gene expression time-series analysis continues to grow quickly. A recent review, [10], outlines the state of the art in gene expression time-series analysis, including much information on clustering methods and software. We can see that, when compared to a similar, earlier review, [11], a considerable amount of work has been done. However, we feel that there is still a branch of time-series data analysis that is under-utilized in gene regulatory network inference. Despite the vast amount of work that has been done on the clustering of gene expression data, much of which deals specifically with time-series, relatively little work has been done on inferring time-dependent interactions between gene clusters or between a gene cluster and an individual gene. Let us briefly discuss clustering methods for time-series data before continuing on to its potential use in inferring gene regulatory networks.

In order to successfully cluster time-series data, we need to utilize the stronger dependencies between data in consecutive time points relative to more distant time points. Quite often, researchers are interested in expression patterns across time; [12] cluster short time-series data around specific pre-determined profiles that may have meaning within the particular experiment. [13] perform a similar cluster analysis of time-series, but instead of using pre-determined expression patterns, they use a hidden Markov model (HMM) to infer dynamics of a limited number of clusters between a small number of states (*e.g.* nine discrete expression levels). [14] take a slightly different approach by clustering genes to inferred profiles, focusing mainly on impulse models in experiments where one might expect peaks in the expression values.

In each of the above papers, it was shown that gene clustering can infer biological meaning, whether co-expression, co-regulation, involvement in particular biological processes, or some other effect. Such information may also be valuable in inferring genetic regulatory networks. [15] and [16] have done work in this direction, combining state-space models and clustering heuristics for simultaneous, integrated inference. However, both of these are demonstrated on data containing hundreds of genes which are clustered or grouped into a low (fewer than 20) number of clusters/modules and subsequently the large cluster size prevents any meaningful conclusions about regulatory interactions between specific genes.

In this paper, we describe a fully Bayesian model of gene cluster interaction, and we demonstrate that probabilistic gene clustering in conjunction with a dynamic Bayesian network can aid in the inference of gene regulatory networks, even in the *DREAM4* data sets, where no clusters were explicitly included. It achieves this by potentially reducing–in a fully Bayesian manner–the parameter space and helping solve the problem of solution identifiability in under-defined, noisy data models such as are common in gene expression analysis. The algorithm presented here is a variational Bayesian hybrid of a DBN and a Gaussian mixture clustering algorithm, both of which have been shown to infer meaningful solutions to their respective problems [8,17], and which we show can work even better in tandem. We call this algorithm *BAyesian Clustering Over Networks* (*BACON*). *BACON* is built specifically to simultaneously consider multiple data sets based on the same network, such that for each data set, expression states are inferred independently, but that cluster membership and regulatory dynamics are assumed to be constant for all data from the given network, regardless of the particular data set. This gives more accurate results than a heuristic combination of interaction rankings based on the various time-series for each of the *DREAM4* networks.

Methods

In this paper we introduce an algorithm called *BACON*, which is a variational Bayesian algorithm that combines a Gaussian mixture clustering model with a DBN. However, before we give the specific formulation of our model, it may be helpful first to look at a simple case where integrated clustering can help infer gene regulatory networks, even if no "true" clusters are present.

A simple example

Assume, as an illustration, that we have a three genes, X, Y, and Z and that we have time-series expression data for each of them, such that the observed expression levels of these at time t are given by x_t, y_t, and z_t, respectively, for time points $t \in \{1,\ldots,T\}$. Let us, for simplicity's sake, assume that we are concerned only with potenial regulators of gene Z, and that X and Y are the only two candidates. Furthermore, we assume a simple linear model of dynamics, in which z_{t+1} is assumed to be a noisy observation of the dot/inner product of the vector of two interaction coefficients a_x and a_y with the vector of x_t and y_t, namely:

$$z_{t+1} = (a_x, a_y)(x_t, y_t)' + \epsilon \tag{3}$$

Note that this is a simple linear model on three variables, where all interaction coefficients except a_x and a_y are set to zero. It is a special case of the standard linear model, given in equation 1, where for this example we treat the observations as the true expression values, we attempt only to infer a_x and a_y, and for illustration purposes we ignore all other possible interaction coefficients. When attempting to infer a_x and a_y under a Bayesian framework, we make the following assumptions:

$$z_{t+1} \sim \mathcal{N}((a_x, a_y)(x_t, y_t)', \lambda) \tag{4}$$

$$(a_x, a_y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}) \tag{5}$$

where λ is a precision parameter (inverse of variance), $\mathbf{m}$ is the prior mean of the multivariate normal distribution, and $\mathbf{L}$ is a 2×2 precision matrix (inverse of the covariance matrix).

Given these assumptions, the data x_t, y_t, and z_t, and the precision parameter λ (fixed), the estimated posterior distribution for (a_x, a_y) under a variational Bayesian framework (see [?] and [?] for a detailed explanation) is multivariate normal, with mean $\hat{\mathbf{m}}$ and precision $\hat{\mathbf{L}}$, such that

$$\hat{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda} + \sum_{t=1}^{T-1} \lambda \begin{bmatrix} x_t^2 & x_t y_t \\ x_t y_t & y_t^2 \end{bmatrix} \tag{6}$$

$$\hat{\boldsymbol{\mu}} = (\boldsymbol{\mu} + \sum_{t=1}^{T-1} \lambda \begin{bmatrix} x_t & y_t \end{bmatrix}) \hat{\boldsymbol{\Lambda}}^{-1} \tag{7}$$

Under some conditions, such inference works quite well, but if the expression profiles for X and Y are highly correlated (or negatively correlated), then the determinant of $\hat{\boldsymbol{\Lambda}}$ approaches zero, and the diagonal elements of $\hat{\boldsymbol{\Lambda}}^{-1}$ (the estimated variances of a_x and a_y) approach infinity. Such a problem can be overcome with a strong prior for a_x and a_y, but this is usually not desireable since

Table 1. Algorithm results comparison for the DREAM4 networks.

	Algorithm	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5
AUROC	*ACON*	**0.82**	**0.67**	0.72	0.81	0.88
	BACON (no clustering)	**0.82**	**0.67**	0.72	0.81	0.88
	G1DBN	0.73	0.64	0.68	**0.85**	**0.92**
	VBSSM	0.73	0.66	**0.77**	0.80	0.84
AUPR	*BACON*	**0.42**	0.36	**0.51**	0.49	0.57
	BACON (no clustering)	**0.42**	0.36	**0.51**	0.49	0.57
	G1DBN	0.37	0.34	0.45	**0.69**	**0.77**
	VBSSM	0.38	**0.41**	0.49	0.46	0.64

The area under the receiver operating characteristic (AUROC) curve and area under precision-recall (AUPR) curve for each of the five data sets. Here, we included *BACON* without clustering in order to establish that the plain DBN algorithm is generally as good as the other two DBN algorithms. The scores for *G1DBN* and *VBSSM* were taken from [7]. The best score for each data set is shown in bold.

doi:10.1371/journal.pone.0068358.t001

typically $\boldsymbol{\mu}$ is set to zero (as in [?]), and a high prior precision $\boldsymbol{\Lambda}$ merely pulls the estimate $\hat{\boldsymbol{\mu}}$ towards zero, and potentially decreases the statistical significance of the inferred interaction parameters. Thus, we are faced with a decision between strong priors or very high variances of posterior parameter estimates.

If the x_t and y_t are highly correlated, and if they are likewise correlated with z_{t+1}, then we might be able to say with near certainty that either X or Y regulates Z, but we could not say which one. This may be acceptable on a small scale, but would be difficult in a gene expression time-series experiment with hundreds of genes and thousands of putative interaction coefficients and covariances. It could be interesting to optimize the choice of a set of, for example, ten gene interactions, with respect to the probability of at least one of them being verifiable in an independent evaluation. But, this would be difficult for experiments with large numbers of genes, and so typically only the individual variances are considered when calculating the statistical significance of the estimated interaction parameter values. Estimating, potentially, thousands of interaction parameters is very difficult in dynamic gene expression time-series analysis because, for example in the basic linear model given in equation 1, there are generally many possible values for the transition matrix, each of which could produce the data. In other words, a given gene expression time-series could be reproduced by many different linear combinations of other [lagged] time-series. A particular case of this is when two potentially regulating genes have highly correlated expression profiles, which, as we have shown above, can cause some difficulty in inference.

Here, we propose that clustering genes and inferring the dynamics of the clusters can help avoid the case in which highly correlated gene profiles inhibit interaction inference. In our example, if genes X and Y have highly correlated expression profiles, then for weak priors the precision estimate in equation 6 is nearly singular, and thus by treating X and Y as two contributors to the same dynamic quantity, we avoid this particular singularity problem altogether. Then, a standard method (DBN or similar) could more easily infer that both X and Y (as one cluster) are likely regulators of Z. If, when creating a list of the most likely individual gene-gene interactions, we simply assign all the inferred interaction coefficients for a cluster to each of its members, we can obtain a ranking of interaction pairs that is comparable to the ranking obtained from a standard DBN.

It may seem, at first, that passing along inferred interaction coefficients to all cluster members would create many false positives. However, if clusters include–by definition–highly correlated expression profiles, then if a cluster appears to be a good potential regulator of a gene, all of the cluster's members must also have profiles that generally indicate potential regulation, and in the absence of clustering, it would be difficult to identify the best interaction parameters. This is true whether or not any or all of the concerned genes are actually verifiable regulators, and thus clustering together correlated expression profiles–regardless of the biological meaning of the clustered genes–could improve inference. For instance, in our example, the presence of gene Y (if highly correlated with X) adversely affects the identification of X as a regulator of Z, a problem that can be avoided if X and Y are treated as members of the same cluster. In a data set with hundreds of genes, the chance of having at least one pair of highly correlated expression profiles is rather large. Of course, we must be careful in our construction of clusters and their dynamics, but as we show, Bayesian inference provides the means to select a number of clusters, to assign cluster membership, and to estimate cluster interaction parameters in an optimal way. We describe this below.

Model

Given K clusters and G genes, we assume that the cluster expressions $\mathbf{F_t}$ for time points $t\in\{1,...,T\}$ follow the standard linear dynamics

$$\mathbf{F_{t+1}} = \mathbf{SF_t} + \mathbf{S_c} + \epsilon \tag{8}$$

where $\mathbf{F}_t$ is a vector of length K, $\mathbf{S}$ is a $K\times K$ transition matrix, $\mathbf{S_c}$ is a column vector of length K, and ϵ is vector of Gaussian noise. The k^{th} element of $\mathbf{F_t}, f_{kt}$, is the expression of cluster k at time t. The vector $\mathbf{S_c}$ represents linear trends in cluster expression levels over the time points, and its inclusion in the model prevents such trends from being confused for interactions in similarly trending clusters.

The expression of gene g at time t is given by μ_{gt}, and the membership of gene g to cluster k is given by the k^{th} element of the indicator vector ξ_g. Each gene g belongs to exactly one cluster k, and so ξ_g contains a single 1 in in the k^{th} element and zeros elsewhere. The n^{th} observation/replicate of μ_{gt} is x_{gtn}. The corresponding prior distributions are:

$$x_{gtn} \sim \mathcal{N}(\mu_{gt}, \lambda) \tag{9}$$

$$\mu_{gt} \sim \mathcal{N}(\xi \mathbf{F_t}', \xi_g \gamma_t') \tag{10}$$

$$\mathbf{F_t} \sim \mathcal{N}(\mathbf{S}\mathbf{F_{t-1}} + \mathbf{S_c}, \Sigma) \tag{11}$$

where the λ is a technical precision (inverse of variance) representing the measurement errors, assumed to be independent, γ_t is a vector of precisions of length K, and Σ is a $K \times K$ precision matrix which we require to be diagonal, as in [8], so that in the posterior distribution estimates, the rows of $\mathbf{S}$ are independent. We also formulate our other prior distributions as in [8]: for the elements of $\mathbf{F}_1$, $\mathbf{S}$, and $\mathbf{S_c}$, we use zero-mean normal distribution priors whose precisions we iteratively update to maximize the marginal likelihood estimate (discussed below). Likewise, for the hyper-parameters of the gamma distribution priors we assume for the elements of the precisions λ, γ_t, and Σ. For ξ_g, we use a uniform prior distribution over the K possible clusters.

For multiple time-course data sets from the same gene regulatory network, as we have in the *DREAM4 Challenge* data sets we use in this paper, we infer all of the parameters separately for each of the series, except for the dynamics parameters $\mathbf{S}$ and $\mathbf{S_c}$ and the membership indicator vectors ξ_g, which are shared and inferred simultaneously for all time-series from the given network.

Inference

To estimate the parameters of our model, we use a variational Bayesian algorithm analagous to those described in [18] and [19], which has been previously used to fit a DBN to gene expression time-series in [8], as well as a Gaussian mixture model for gene clustering in [17].

In short, the algorithm used in this paper estimates the posterior parameter distribution $P(\boldsymbol{\theta}|\mathbf{D})$ given the data $\mathbf{D}$ using a factorable distribution $Q(\boldsymbol{\theta}) = Q(\theta_1)Q(\theta_2)\ldots Q(\theta_n)$ whose factors can be iteratively updated so that with each update, $Q(\boldsymbol{\theta})$ becomes a better approximation for $P(\boldsymbol{\theta}|\mathbf{D})$, as measured by the Kullback-Leibler divergence between the two. We have chosen conjugate prior distributions for each of the parameters we estimate, and therefore the posterior distribution estimate $Q(\theta_i)$ for each parameter is of the same form as its prior, and the parameters of these distributions are updated iteratively according to variational Bayesian inference, as in equations 6 and 7.

We fit the model using 10 starts with randomized initial parameter values, and with a range of cluster numbers less than or equal to the number of genes in the data set (in the case of the *DREAM4* data, $k \in \{5,6,\ldots,10\}$) and then accept the model that has the highest estimated marginal likelihood. Accepting the model with the maximum marginal likelihood is simpler than combining all models based on their likelihoods, when in fact it is rare for a second, different model to have a likelihood close enough (i.e. a log likelihood within 3 or 4) to the best model for it to make a significant impact on the interaction rankings.

We are concerned primarily with the transition matrix $\mathbf{S}$ and the membership indicators ξ_g; using posterior estimates for these, we can rank directed gene-gene interactions by their statistical strength. Specifically, for each directed cluster pair interaction $i \rightarrow j$ ($i \neq j$), we calculate the posterior mean estimate for element (i,j) of $\mathbf{S}$ divided by its posterior standard deviation, assign this value to all possible directed pairs within the two clusters, and we rank by largest absolute value.

The *Octave* code implementing this algorithm–available at: http://code.google.com/p/bacon-for-genetic-networks-takes approximately 40 minutes on a single core of a 1.2 GHz processor for a single random start and a given number of clusters. Multiple starts and different numbers of clusters can be run in parallel; see the code for more details.

Data

We used the *DREAM4 In Silico Network Challenge* data sets to evaluate the performance of our model. See [2,20–22] for more details on the *DREAM* challenges. We utilized only the 10-gene time-series data, which consists of five simulated networks. For each of the networks, there are five time-series experiments, each with 20 time points. No simulated technical replicates were included, but random noise was added. The list of actual, "gold standard", interactions was provided after the official challenges ended.

Results

For each of the five data sets, each corresponding to a single gene regulatory network, we inferred the network using all

Table 2. Results of BACON on individual DREAM4 time series.

	Time-series	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5
AUROC	1	0.68 (**0.71**)	0.61 (0.61)	0.64 (0.64)	0.62 (0.62)	0.57 (0.57
	2	**0.75** (0.66)	0.70 (0.70)	**0.68** (0.66)	0.77 (0.77)	0.64 (0.64)
	3	**0.67** (0.61)	0.62 (0.62)	**0.65** (0.60)	**0.60** (0.58)	0.65 (0.65)
	4	**0.61** (0.53)	0.66 (0.66)	0.59 (0.59)	**0.60** (0.60)	**0.74** (0.66)
	5	**0.66** (0.63)	0.64 (0.64)	0.59 (0.59)	0.76 (0.76)	0.78 (0.78)
AUPR	1	0.24 (**0.39**)	0.24 (0.24)	0.32 (0.32)	0.19 (0.19)	0.23 (0.23)
	2	**0.42** (0.27)	0.34 (0.34)	0.28 (**0.30**)	0.26 (0.26)	0.32 (0.32)
	3	**0.30** (0.19)	0.21 (0.21)	**0.24** (0.19)	0.15 (**0.16**)	0.24 (0.24)
	4	**0.24** (0.16)	0.38 (0.38)	0.21 (0.21)	**0.24** (0.18)	**0.27** (0.23)
	5	**0.22** (0.19)	0.20 (0.20)	0.17 (0.17)	0.34 (0.34)	0.33 (0.33)

For each of five individual time-series in each of the five data sets, the area under the receiver operating characteristic (AUROC) curve and area under precision-recall (AUPR) curve. For each time series, we give two of each score, one for *BACON* with clustering and one for *BACON* without clustering (in parentheses). The higher of the two scores appears in bold. If the two scores are identical, neither is in bold.

doi:10.1371/journal.pone.0068358.t002

available time-series (five each) and used the inferred interactions and the known gold standard to calculate the area under the receiver operating characteristic (AUROC) curve and the area under precision-recall (AUPR) curve, as in [1]. Table 1 gives the AUROC and AUPR for *BACON* both with and without clustering, as well as the corresponding scores from the *G1DBN* and *VBSSM* models, as reported in [1]. *BACON* gives an AUROC score better than both *G1DBN* and *VBSSM* in two out of five data sets– likewise for the AUPR scores–and is comparable to the other two algorithms in the remaining data sets. Given that *BACON* without clustering compares favorably with other algorithms, and that *BACON* with clustering gives the exact same results as *BACON* without clustering (the inferred number of clusters in each case was 10, the number of genes), we conclude that both versions of *BACON* give satisfactory results for these data sets.

However, the *DREAM4* time-series data are not typical; a single time-series with 20 time points is somewhat uncommon in practice (most experiments have 10 or fewer time points), and five independent time-series for the same gene network would be extremely rare. Thus, we subsequently consider each of the time-series individually, in order to see if an even more under-determined problem (only 20 data points for each of the 10 genes instead of 100) favors the model version with clsutering. We show in Table 2 the AUROC and AUPR of the 25 individual time-series (five from each of five data sets) for the *BACON* model both with and without clustering.

In many cases, the with-clustering and without-clustering scores were identical–i.e. 10 clusters is optimal–but in several other cases, fewer clusters gave a higher marginal likelihood score, and the corresponding AUROC and AUPR were indeed better, more often than not. Specifically, for 15 of the 25 time-series, *BACON* with clustering performed identically to the version without, but in seven cases, the version with clustering gave higher scores for both AUROC and AUPR. In only one case, the without-clustering version outperformed the with-clustering version in both AUROC and AUPR. These tallies are summarized in Table 3. Clearly, for smaller data sets such as a single time series, there is some benefit to be had from clustering the genes, when compared to non-clustering DBNs.

Table 3. Results comparison: with vs without clustering.

		Higher AUPR		
		with clustering	**equal**	**without**
	with clustering	7	0	2
Higher AUROC	equal	0	15	0
	without	0	0	1

Among the five individual time-series in each of the five data sets (25 total time series), here we give a tally of how many times *BACON* with clustering outperformed *BACON* without clustering, or vice versa, or if the AUROC and AUPR scores are equal.

doi:10.1371/journal.pone.0068358.t003

Discussion

Inferring gene regulatory networks from expression data is not usually easy, but it is common and often useful. Because of the under-determined nature of the problem–there are more parameters than data points–some reduction of the parameter set is often necessary in order to reach any meaningful conclusion at all. Sometimes, we can accomplish this through heuristic methods and decisions about which data are more important prior to the main statistical analysis. Other times, this is not desirable. In this paper, we present a probabilistic model of time-series gene expression with an integrated, theoretically sound method of parameter space reduction. We have described its implemetation and use, including a simple analytically-tractable example in which clustering is advantageous to network inference even if no "true" cluster exists, and if we are not at all concerned with cluster membership.

Many of the expectations we had for the Bayesian model turned out to be true. In particular, we expected the model to favor clustering mainly in data sets with few samples; in fact, the model preferred (via the likelihood function) not to cluster when we included all data for each network (100 samples, 20 from each of five time-series), but elected to cluster for 10 of the 25 separate time series (20 samples each). Likewise, because of the under-determined nature of network inference, we also expected the clustering model to perform better than a model without clustering if there are fewer samples. This also proved true; of the 10 time-series for which the model's marginal likelihood was highest for less than 10 clusters, seven were indeed better than without clustering (when comparing both AUROC and AUPR scores), and only one proved worse.

We believe that probabilistic clustering could be very useful in gene network inference, though there are disadvantages. For one, the computational time is generally much higher when clustering. This is due to the need to do model fits for a range of possible cluster numbers. For the purposes of this paper, in addition to doing the 10 random starts for the non-clustering model version, we do 10 random starts for the cluster quantities we wish to consider. Of course, the algorithm is much faster for smaller cluster numbers, as the size of the parameter of primary interest, the interaction/transition matrix, varies with the square of the number of clusters. It would likely be beneficial, in the case of very large data sets, to use a sequential or iterative search over the number of clusters, rather than use the exhaustive search method as we have here, but we leave that for a future publication.

In summary, we have shown that there are benefits to be had by clustering genes as part of a network inference algorithm. The potential for significant correlation among genes is high in typical time-series data sets, particularly those with few samples. The algorithm we have presented here, which we call *BAyesian Clustering Over Networks* (*BACON*), can help avoid the negative consequences of inter-gene correlation for the purposes of network inference. In our tests, the algorithm outperformed its non-clustering version in 7 out of 25 time-series from the *DREAM4 Challenge*, under-performing only once, and most often electing to disregard clusters when the data did not support it. Therefore, we feel that there are significant benefits of using probabilistic clustering to aid in the inference of gene regulatory networks.

Source code (*GNU Octave*), more information about the software for BAyesian Clustering Over Networks, (*BACON*) and sample data can be found at: http://code.google.com/p/bacon-for-genetic-networks.

Author Contributions

Analyzed the data: BG. Contributed reagents/materials/analysis tools: BG. Wrote the paper: BG.

References

1. Penfold CA, Wild DL (2011) How to infer gene networks from expression profiles, revisited. Inter-face Focus 1: 857–870.
2. Prill RJ, Saez-Rodriguez J, Alexopoulos LG, Sorger PK, Stolovitzky G (2011) Crowdsourcing Network Inference: The DREAM Predictive Signaling Network Challenge. Sci Signal 4: mr7+.
3. Greenfield A, Madar A, Ostrer H, Bonneau R (2010) DREAM4: Combining Genetic and Dynamic Information to Identify Biological Networks and Dynamical Models. PLoS ONE 5: e13397+.
4. Kim SY, Imoto S, Miyano S (2003) Inferring gene networks from time series microarray data using dynamic Bayesian networks. Brief Bioinform 4: 228–235.
5. Husmeier D (2003) Sensitivity and specificity of inferring genetic regulatory interactions frommicroarray experiments with dynamic Bayesian networks. Bioinformatics 19: 2271–2282.
6. Lèbre S (2009) Inferring Dynamic Genetic Networks with Low Order Independencies. Statistical Applications in Genetics and Molecular Biology 8: 1–38.
7. R Development Core Team (2009) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
8. Beal MJ, Falciani F, Ghahramani Z, Rangel C, Wild DL (2005) A Bayesian approach to recon-structing genetic regulatory networks with hidden factors. Bioinformatics 21: 349–356.
9. Klemm SL (2008) Causal Structure Identi_cation in Nonlinear Dynamical Systems. Master's thesis, Department of Engineering, University of Cambridge, Cambridge, UK.
10. Bar-Joseph Z, Gitter A, Simon I (2012) Studying and modelling dynamic biological processes using time-series gene expression data. Nat Rev Genet 13: 552–564.
11. Bar-Joseph Z (2004) Analyzing time series gene expression data. Bioinformatics 20.
12. Ernst J, Nau GJ, Bar-Joseph Z (2005) Clustering short time series gene expression data. Bioinfor-matics (Oxford, England) 21 Suppl 1: i159–168.
13. Schliep A, Schonhuth A, Steinhoff C (2003) Using hidden Markov models to analyze gene expression time course data. Bioinformatics 19: i255–i263.
14. Sivriver J, Habib N, Friedman N (2011) An integrative clustering and modeling algorithm for dynamical gene expression data. Bioinformatics 27: i392–i400.
15. Hirose O, Yoshida R, Imoto S, Yamaguchi R, Higuchi T, et al. (2008) Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models. Bioinformatics 24: 932–942.
16. Shiraishi Y, Kimura S, Okada M (2010) Inferring cluster-based networks from differently stimulated multiple time-course gene expression data. Bioinformatics 26: 1073–1081.
17. Teschendorff AE, Wang Y, Barbosa-Morais NL, Brenton JD, Caldas C (2005) A variational Bayesian mixture modelling framework for cluster analysis of gene-expression data. Bioinformatics 21: 3025–3033.
18. Beal MJ (2003) Variational algorithms for approximate Bayesian inference. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London. Available: http://www.cse.buffalo.edu/faculty/mbeal/thesis/. University at Buffalo website. Accessed 2013 May.
19. Winn JM (2003) Variational Message Passing and its Applications. Ph.D. thesis, St Johns College, Cambridge, Cambridge, England.
20. Prill RJ, Marbach D, Saez-Rodriguez J, Sorger PK, Alexopoulos LG, et al. (2010) Towards a Rig-orous Assessment of Systems Biology Models: The DREAM3 Challenges. PLoS ONE 5: e9202+.
21. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, et al. (2010) Revealing strengths and weaknesses of methods for gene network inference. PNAS 107: 6286–6291.
22. Marbach D, Schaffter T, Mattiussi C, Floreano D (2009) Generating Realistic In Silico Gene Net-works for Performance Assessment of Reverse Engineering Methods. Journal of Computational Biology 16: 229–239.