

Learning Regulatory Networks from Sparsely Sampled Time Series Expression Data

Anshul Kundaje

Dept. of Electrical Engineering
Columbia University

Tony Jebara

Machine Learning Laboratory
Dept. of Computer Science
Columbia University

Omar Antar

Dept. of Molecular Genetics
Albert Einstein College of Medicine

Christina Leslie*

Dept. of Computer Science
Center for Computational Biology and
Bioinformatics (C2B2)
Columbia University

{abk2001, osa2001, jebara, cleslie}@cs.columbia.edu

Abstract

We present a probabilistic modeling approach to learning gene transcriptional regulation networks from time series gene expression data that is appropriate for the sparsely and irregularly sampled time series datasets currently available. We use a clustering algorithm based on statistical splines to estimate continuous probabilistic models for clusters of genes with similar time expression profiles and for individual genes. Using the learned models, we present a novel mutual information score for causal edges between pairs of clusters and between pairs of genes corresponding to a given time lag δ . This score computes dependency between expression values as continuous quantities rather than discretizing them. We present empirical results on time series data for the yeast cell cycle, using randomization trials to determine statistically significant candidate network edges and the Chow-Liu graph learning algorithm to learn the network structure, to obtain a dynamic model of cell cycle regulation. Biological validation of the inferred network suggests that our method can learn a meaningful, higher-level view of regulatory networks from sparse time series data.

Keywords: gene expression; gene regulatory networks; time series; machine learning.

1 Introduction

An exciting new area of research in computational biology is the problem of learning models of gene transcriptional regulation networks from microarray hybridization data. A number of recent papers – including work of Pe’er *et al.* [8], Hartemink *et al.* [4], and Ong *et al.*

*Corresponding author. Email: cleslie@cs.columbia.edu. Address: 1214 Amsterdam Avenue, Mail Code 0401, New York, NY 10027. Telephone: 212-939-7043. Fax: 212-666-0140.

[7] – have approached this problem with the formalism of Bayes nets (also called graphical models), where each gene expression level is modeled as random variable corresponding to a node in a directed acyclic graph G . The main focus of research has been to learn structure of the graphical model G from observations of the joint distribution of gene expression levels.

Microarray data presents difficult challenges to the sophisticated probabilistic methods. The data is noisy, and while there are thousands of random variables (genes) to model, there are at best only a few hundred joint observations. All papers cited above discretize the data prior to Bayes net modeling, making the learning procedure more robust to sparse and noisy data but losing information in the process. Since learning a network involving all the genes is not possible with current datasets, the goals of learning must be more modest: for example, finding small features of the network that are statistically significant [8] or scoring small candidate models involving a few genes [4]. A more subtle difficulty of the standard Bayes net model is that the directed edges of the graph G do not in general imply causal interaction between genes. If knockout data is available, some causal edges can be learned by modeling the knockout experiments as *interventions* in the graph [8]. A simpler approach to directly learning causality is to use time series expression data and a dynamic probabilistic model. Here, the structure learning problem consists of learning edges across the *time slice*, representing a time lag of δ , from pairs of experiments at times $(t, t + \delta)$; the forward direction of time implies causality. However, time series expression datasets are even more sparse and may be irregularly sampled over the time course. Ong *et al.* try to perform structure learning of a dynamic Bayes net directly from a small dataset [7] and report some of the difficulties of this approach.

In this paper, we present a new approach for learning the structure of a dynamic network model from sparsely sampled time series expression data. Since we have many genes but relatively few time points, we can obtain a better-supported probabilistic model for *clusters* of genes with similar expression profiles across the time course rather than trying to model individual genes. Therefore, a node in our dynamic model represents the expression level for a cluster of similarly-behaving genes rather than a single gene. We learn assignments of genes to clusters as well as statistical spline models for both genes and clusters using an expectation maximization approach as presented by Bar-Joseph *et al.* [1]. With the statistical spline approach, we avoid discretizing the data, and we can learn well-supported, continuous and time-varying probabilistic models for the cluster expression levels from sparse and irregularly sampled time series data. Using the cluster probability models, we present a novel mutual information score for detecting dependence of cluster variables across a time slice, which we use as the basis for our network structure learning. Our mutual information score measures dependencies between *continuous* models for the cluster expression variables rather than discretized empirical probability distributions. We present results on yeast cell-cycle data [3], where we use randomization trials to obtain a threshold for a statistically significant network edges and learn a sparse graphical model based on the Chow-Liu algorithm for tree learning [2].

Previous research combining clustering with network inference include work of Toh and Horimoto [10], who build a partial correlation coefficient matrix based on averaged expression profiles, and Mjolsness *et. al* [6], who model regulation of aggregate genes using analog neural network dynamics.

While cluster-to-cluster interactions may seem less interesting than gene-to-gene interactions, in our results on yeast cell-cycle data, we find that the clusters consist of genes with similar function; by using gene ontology annotations for the genes assigned to a cluster, we can attribute a clear biological meaning to many of the clusters. We also find that

the cluster-to-cluster edges in our network model reflect known interactions from yeast biology. Thus our approach appears successful in learning meaningful, higher-level regulatory behavior from sparse time series data in this example.

2 Cluster and Gene Models using Statistical Splines

We begin by learning a continuous, time-varying probabilistic model for the expression levels of “clusters” of genes with similar expression profiles over a time course of microarray experiments. We learn both a parameterized model for the cluster profiles and a **soft assignment** of genes to clusters with a clustering technique introduced by Bar-Joseph *et al.* [1], based on the statistical spline model given in James and Hastie [5].

INITIAL HERE

We model the true expression level for gene i , belonging to cluster j as a function of time as $(s_1(t) \cdots s_q(t)) (\boldsymbol{\mu}_j + \boldsymbol{\gamma}_{ij})$, where $s_1(t), \dots, s_q(t)$ are **spline basis functions**, $\boldsymbol{\mu}_j$ is the **mean vector of spline coefficients for cluster j** and $\boldsymbol{\gamma}_{ij}$ is the **gene-specific variation vector of spline coefficients**. We assume that each experimental observation is subject to Gaussian noise, with error $\epsilon \sim N(0, \sigma^2)$. Therefore, for a vector of observations for gene i at times t_1, \dots, t_m , we have:

$$\mathbf{Y}_i = \begin{pmatrix} Y_i(t_1) \\ \vdots \\ Y_i(t_m) \end{pmatrix} = \begin{pmatrix} s_1(t_1) & \cdots & s_q(t_1) \\ \vdots & \vdots & \vdots \\ s_1(t_m) & \cdots & s_q(t_m) \end{pmatrix} \left[\begin{pmatrix} \mu_j^1 \\ \vdots \\ \mu_j^q \end{pmatrix} + \begin{pmatrix} \gamma_{ij}^1 \\ \vdots \\ \gamma_{ij}^q \end{pmatrix} \right] + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{pmatrix}.$$

We use expectation maximization (EM) on our dataset to learn parameters σ , $\boldsymbol{\mu}_j$, $\boldsymbol{\gamma}_{ij}$, as well as the **$q \times q$ covariance matrix Γ_j** for the vectors $\boldsymbol{\gamma}_{ij}$ and the posterior probability $p(j|i)$ of gene i belonging to cluster j , given the data. Here we follow the **EM formulation** described in [1], but we modify the algorithm with **deterministic annealing** to avoid converging to poor **local maxima of the complete log likelihood function**. Details of the EM procedure are given in the appendix.

When most of the genes satisfy the condition that $\max_j p(j|i)$ is close to 1, we can use the j that maximizes $p(j|i)$ as the **hard cluster assignment** of gene i . This leads to continuous probabilistic models for both *cluster expression profiles* and *gene expression profiles* that are functions of time. Specifically, our model gives the following probability distribution for (observed) expression level Y_i at time t of gene i with cluster assignment j

$$P(Y_i|t) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left(-\frac{1}{2\sigma^2} \left(Y_i - \sum_{k=1}^q s_k(t) (\mu_j^k + \gamma_{ij}^k) \right)^2 \right)$$

For the cluster models, let X_j represent the “cluster” expression random variable. Then we can write $X_j(t) = (s_1(t) \cdots s_q(t)) (\boldsymbol{\mu}_j + \boldsymbol{\gamma}_j) + \epsilon$ where $\boldsymbol{\gamma}_j \sim N(0, \Gamma_j)$ and $\epsilon \sim N(0, \sigma^2)$. We get one-dimensional cluster mean value of $E(X_j|t) = (s_1(t) \cdots s_q(t)) \boldsymbol{\mu}_j = \sum_{k=1}^q s_k(t) \mu_j^k$

and variance $\text{Var}(X_j|t) = (s_1(t) \cdots s_q(t)) \Gamma_j \begin{pmatrix} s_1(t) \\ \vdots \\ s_q(t) \end{pmatrix} + \sigma^2$. Letting $\theta_j^2(t) = \text{Var}(X_j|t)$

(scalar variance), we get the following **probabilistic model for the cluster random variable**:

$$P(X_j|t) = \frac{1}{\sqrt{2\pi}\theta_j(t)} \exp \left(-\frac{1}{2\theta_j^2(t)} \left(X_j - \sum_{k=1}^q s_k(t) \mu_j^k \right)^2 \right)$$

3 Mutual Information from Continuous Models

The recovered spline parameters determine a curve and a distribution over cluster expression levels (as well as individual genes) over a continuous range of time. The availability of this temporal curve distribution permits us to perform a quantitative measurement of the statistical independence between one temporal expression profile a given cluster and the temporal expression profile of another. The pairwise measure of the independence is performed through a mutual information computation, as described below.

We model the joint distribution of two clusters, X_a and X_b , at time t and separated by timelag δ as instantaneously independent: $P(X_a, X_b|t, \delta) = P(X_a|t)P(X_b|t + \delta)$. We integrate with respect to time to obtain an estimated model of the joint distribution of cluster expression variables X_a and X_b separated by timelag δ :

$$P(X_a, X_b; \delta) = \frac{1}{t_{\text{final}} - t_{\text{initial}}} \int P(X_a|t)P(X_b|t + \delta)dt.$$

We can think of this integral as taking a continuous uniform mixture with respect to the time variable. Similarly, we can integrate to define $P(X_a)$ and $P(X_b)$. Now we define a **mutual information score** on **the edge across the timeslice** with timelag δ from X_a to X_b as

$$I(X_a, X_b; \delta) = \int P(X_a, X_b; \delta) \log \frac{P(X_a, X_b; \delta)}{P(X_a)P(X_b; \delta)} dX_a dX_b.$$

The mutual information is the **Kullback-Leibler** distance between the joint probability distribution $P(X_a, X_b; \delta)$ and the product of the independent distributions. Therefore, the score gives an indication of *how far* the distributions of X_a and X_b are *from being independent* when we consider pairs of values separated by time lag δ . A high mutual information score $I(X_a, X_b; \delta)$ gives evidence for a *causal edge* from X_a to X_b . Two fully independent cluster expression distributions (separated by time lag δ) would have a mutual information of 0. We present our method for learning network structure from pairwise mutual information scores in section 4.2.

To approximate the integral for $I(X_a, X_b; \delta)$, we replace the continuous mixture with respect to time by a discrete sum over T evenly spaced time points in expressions, so that all the probability distributions become discrete mixtures of Gaussians. We can then sample from the Gaussian components of $P(X_a, X_b; \delta)$ in standard Monte-Carlo fashion to obtain our approximation:

$$\begin{aligned} I(X_a, X_b; \delta) &\approx \frac{1}{T} \sum_{l=1}^T \int P(X_a|t_l)P(X_b; t_l + \delta) \log \frac{P(X_a, X_b; \delta)}{P(X_a)P(X_b; \delta)} dX_a dX_b \\ &\approx \frac{1}{T} \sum_{l=1}^T \left(\frac{1}{N} \sum_{\substack{N \text{ points } (X_a, X_b) \\ \sim P(X_a|t_l)P(X_b; t_l + \delta)}} \log \frac{\frac{1}{T} \sum_j P(X_a|t_j)P(X_b|t_j + \delta)}{\left(\frac{1}{T} \sum_j P(X_a|t_j)\right) \left(\frac{1}{T} \sum_j P(X_b|t_j + \delta)\right)} \right) \end{aligned}$$

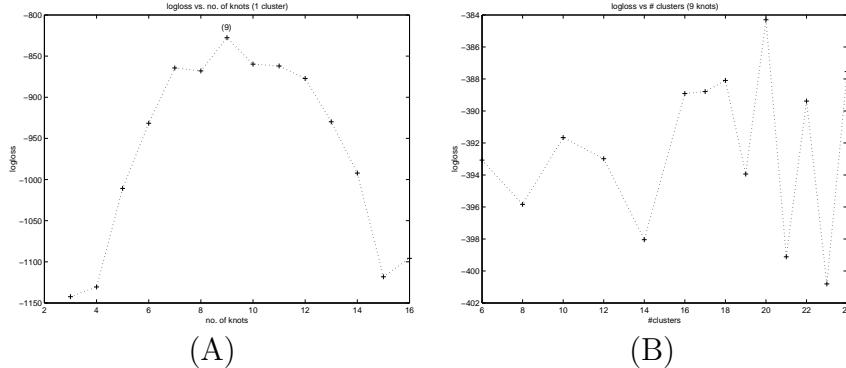


Figure 1: **Log loss curves for cross-validation.** The log loss function for 1 cluster, with the number of knots varying (plot A) and for 9 knots with the number of clusters varying (plot B).

4 Experiments: Yeast Cell Cycle

Spellman *et al.* [3] have listed a set of 799 genes of *Saccharomyces cerevisiae* that they found were cell-cycle regulated. We applied our method to time series data (alpha-pheromone experiment) of 791 genes, which consists of 18 time points sampled every 7 minutes and represents two full cell cycles. We omitted 8 genes due to missing time points. Note that this dataset is not the *cdc15* dataset, whose data is a concatenation of even and odd time points.

4.1 Model Selection by Cross-Validation

Following [1], we used natural cubic splines, where the size q of the spline basis is equal to the number of knots. We used evenly spaced knots and determined optimal value of knots and clusters using the log-loss function on a hold-out set of genes

$$\log \text{ loss} = \sum_{\text{genes } i \text{ in hold-out set}} \log \left(\sum_{\text{clusters } j} p_j P(\mathbf{Y}_i | Z_i = j, \boldsymbol{\mu}_j, \Gamma_j, \sigma^2) \right)$$

where the parameters for the cluster j model are as defined in the Appendix. This cross-validation checks generalization of the learned cluster models to the held-out test set genes.

We ran the clustering algorithm on 90% of the dataset (leaving out a randomly selected 80 genes) and calculated the log loss values on the remaining 10%, based on the training model for different number of knots and clusters. Figure 1 (plot A) shows the log loss as a function of the number of knots when we fixed the number of clusters to one. The peak corresponds to a value of 9 knots. Similar results were found for other cluster values as well ranging from 2 to 24 (results not shown). Hence, 9 knots was chosen as the best value for the number of knots. We similarly calculated the log loss varying the number of cluster from 1 to 24 using a constant value of 9 knots, performing trials on 3 randomly selected hold-out sets for each model (Figure 1, plot B). Based on averaged log-loss results, we found that the best model used 20 clusters.

We also compared the quality of interpolation using our spline-based model with linear interpolation by leaving out time points for all the genes in the dataset and calculating the total RMS error. The spline-based approach was always found to have a lower RMS error

value. Using 20 clusters and 9 knots with the central time point eliminated for all genes, the RMS error using the spline model was 8.610 as compared to 9.346 using the linear model.

4.2 Network Learning from Mutual Information Scores

Using 9 knots and 20 clusters, we ran the EM clustering algorithm using deterministic annealing (see Appendix) on the full dataset of 791 genes several times **using different random cluster centers as starting points**. We observed a consistent clustering of genes and selected the model that gave us the **best complete log likelihood**. Figure 2 (A) gives an example of the learned spline trajectories for one of the clusters.

The pairwise mutual information between clusters for different values of time lag δ of 0 to 50 minutes (in steps of 2 minutes) was calculated using the method in section 3. In order to capture changes in graph structure over the time course, we calculated these mutual information scores over windows corresponding to our **biological** prior knowledge of approximate timing of the phases of the yeast cell cycle, as observed in the experiment [3]. Specifically, in order to capture cluster to cluster interactions across a phase transition, we calculated mutual information over windows corresponding to two consecutive phases of the cell cycle. Since the time course actually represents two full cell cycles, we calculate the mutual information for each phase transition “time slice” over a pair of corresponding time intervals, where one interval comes from each cell cycle. Windows for the transitions were chosen as follows: M/G1-G1 transition, 1-28 and 59-92 minutes; G1-S, 10-43 and 73-105 minutes, S-G2, 24-50 and 86-113 minutes; and G2-M, 38-63 and 101-119 minutes. (See Section 4.3 for a summary of cell cycle phases.)

To obtain a threshold for the mutual information scores prior to any graph learning, we conducted a randomization experiment. Keeping the membership of each cluster the same, we shuffled the time values (order of experiments) of each cluster independently to eliminate any causality and ran the EM algorithm on this new randomized dataset, one cluster at a time, to obtain model parameters for each cluster separately. The mutual information between the clusters based on this randomised model was then calculated for the same time intervals as above. We chose a threshold for each time interval such that and the largest mutual information score was higher than 99.5% of the randomized mutual information scores.

For our structure learning, we were interested in obtaining a multi-layer dynamic network structure where different layers model different time periods of the cell cycle, as opposed to a single-layer network, where the structure is assumed constant across the entire time course. We used a 5-layer model, where each layer contains 20 nodes corresponding to clusters, and mutual information scores derived from time windows corresponding to phase transition gave edge scores for candidate edges across the 4 time slices. All below-threshold edges (as determined by the randomization trials) were removed from the candidate edge set, and only edges between successive layers were allowed. Given the pruned candidate edge set, we used the Chow-Liu algorithm [2], for each transition separately, to learn a maximal weight spanning tree (across each time slice) in order to assemble the network. The algorithm simply involves adding the edge in the graph with largest mutual information while also checking to ensure that no cycles are created in the graph. (Note that cycles can occur over several time slices when we assemble the multi-stage model.) Tree structures are appropriate for pairwise mutual information calculations, and in the case of standard (non-dynamic) Bayes net models, Chow-Liu learns the maximum likelihood tree structure. However, we are primarily using Chow-Liu to enforce biologically reasonable sparsity, and we do not

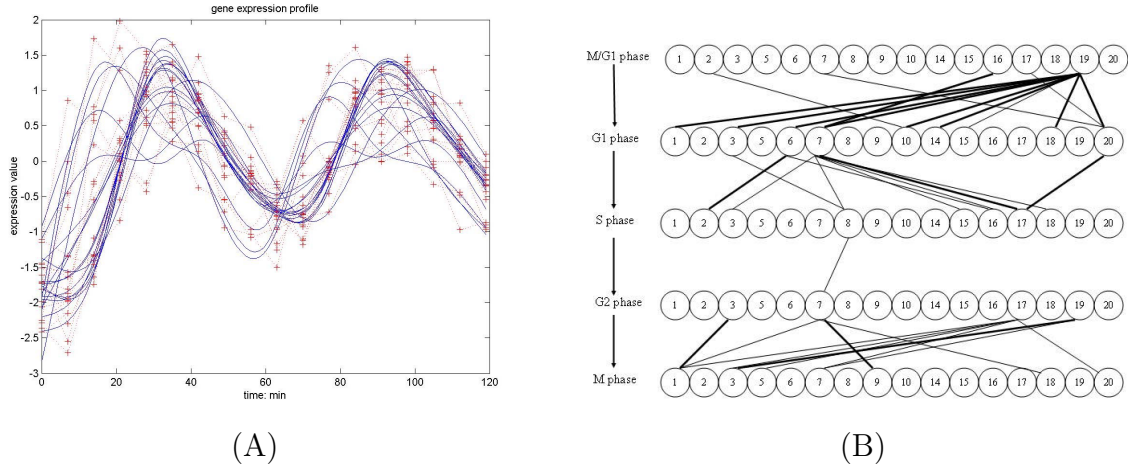


Figure 2: **Network results for yeast cell cycle.** (A) Example: spline models for genes in Cluster 7. (B) Learned multi-phase network structure.

use the edge directions implied by the tree structure but rather the direction of forward time. To learn non-tree structures or more subtle local structures (such as head-to-head edges with weak individual mutual information scores), one must perform multi-information calculations between sets of genes. Algorithms to find optimal graphs based on these scores may be plagued by local maxima and approximations.

For each candidate edge, we used the δ value that resulted in the best mutual information score but eliminated edges for which $\delta < 4$ or $\delta > 35$ minutes maximized mutual information, since mutual information for a very short time lag is likely an artifact of the model or coincidence of shape rather than evidence of causality, and long time lags are not biologically relevant. Using these scores with the Chow-Liu algorithm, we obtained the graph in Figure 2 (B). Biological validation of our learned network structure is given in the next section. Note that different edges can correspond to different causal time lags, which might be a useful indication of the time scale of the kinematics of the causal interactions.

4.3 Biological Validation

The cell cycle clock in *S. cerevisiae* has been empirically categorized into four phases: G1 (gap1), S (synthesis), G2 (gap2), and M (mitosis). Synthesis refers to the synthesis of DNA to complete the duplication of the cell’s chromosomes. Mitosis refers to the final separation of the replicating cell into two daughter cells. The gaps in between these phases are called G1 and G2, but important biological regulation or check points occur at those steps to ensure the orderly progression through the next step in the cell cycle, such as ensuring the genomic DNA is undamaged, or the correct number of chromosomes has been synthesized. A model of the transcriptional control of genes by phase has been shown in a review of the literature [9]. Analysis of these genes and their regulatory targets reveals that genes during one phase of the cell cycle contribute to the regulation of key transcription factors that induce genes in the next phase of the cell cycle, driving the cell cycle and forming a fully connected regulatory circuit.

To interpret our learned network in terms of cell cycle biology, we began by attaching significant biological meaning to our clusters. We assigned each gene to the cluster for

No. (Size)	General GO description ($p < .05$)	Phase	Min p-value GO description
1 (11)	budding, cytokinesis	M/G1	cell wall organization and biogenesis (0.00746)
2 (132)	DNA repair and replication	S	DNA repair (0.00300) ¹
3 (6)	mating	M/G1	regulation of transcription, mating-type specific (0.0228)
4 (10)	cell wall metabolism (budding)	G1	carbohydrate metabolism (0.00334)
5 (13)	cell wall metabolism (budding)	G1	methionine metabolism (5.51E-06)
6 (19)	DNA replication initiation	M/G1	pre-replicative complex formation and maintenance (1.89E-07)
7 (17)	chromatin	S	chromatin assembly/disassembly (1.14E-12)
8 (51)	mitosis	M	proton transport (0.00392)
9 (16)	cell cycle	G1/S, G2/M	regulation of CDK activity (0.000769)
10 (8)	mating	M/G1	reproduction (6.16E-08)
11 (164)	mitosis	M	microtubule-based process (3.76E-05)
12 (221)	budding	M/G1	exocytosis (0.00278)
13 (37)	cell wall metabolism (budding)	G1	carboxylic acid metabolism (7.31E-06)
14 (9)	chromatin	S	phosphate metabolism (0.000658)
15 (15)	energy pathways	S?	iron transport (0.00101)
16 (13)	nitrogen metabolism	M?	one-carbon compound metabolism (0.000258)
17 (25)	DNA replication	S	mitotic cell cycle (5.67E-07)
18 (10)	cell communication (mating)	M/G1	signal transduction (0.00297)
19 (7)	cytokinesis, mating	M/G1	cytokinesis, completion of separation (0.000407)
20 (7)	mating	M/G1	signal transduction during conjugation with cellular fusion (2.40E-05)

Figure 3: **GO annotations for clusters.** Cluster number, size (number of genes), general GO annotation, and GO term with minimum p-value for the cluster.

which it had maximum posterior probability of membership. We considered the complete set of Gene Ontology (GO) biological process annotations or terms for our genes available at the Saccharomyces Genome Database (SGD). For each GO term for genes within a specific cluster, we computed a p-value from the tail of hypergeometric distribution. The p-value estimates the probability that, within the specific cluster, the term would appear at random at least as many times as it does, based on the prevalence of the term for genes in the entire dataset. We give a general GO biological assignment to a cluster based on those GO annotations with p-values of less than 0.05 in Figure 3, and we also list the GO annotation with minimum p-value associated to each cluster. Several of the clusters appear to be dominated by genes regulated in a phase or transcription factor specific manner. Clusters 3, 6, 10, 18, 19, and 20 are associated with the mating or replication initiation process in M/G1 phase. Clusters 1, 4, 5, 12 and 13 are associated with the budding process in late G1 phase. Clusters 2, 7, 14, and 17 are associated with the DNA replication or chromatin maintenance in the S phase. Clusters 8 and 11 are associated with the mitosis or cytokinesis in M phase. Clusters 9, 15, and 16 have questionable assignments to a specific phase. No clusters have a sound association with the G2 phase, the shortest phase in the cell cycle. A complete list of genes for each cluster and with p-value ranked GO annotations will be available at <http://www.cs.columbia.edu/compbio>.

Phases of the cell cycle are annotated on Figure 2 (B), with bold edges representing relationships that are either supported or inferred in the literature [9]. One can observe several connections supported by the literature, but also several that are not. These unsupported connections may be artifacts of the spline interpolation, poor biological representation of the cluster, or previously unknown interactions/function of the genes within the cluster not represented by GO.

Those edges supported by the literature include two general types of regulatory relationships. Some mutual information edges appear to represent genetic relationships approximately within one cell cycle phase (or including part of its adjacent phase). Other mutual information edges appear to represent transitions between cell cycle phases. Examples of biologically supported transition edges found in the M/G1 and G1 phases are mitosis genes in M phase preceding induction of histone genes in late G1/S (16-7), mating genes in M/G1 preceding induction of budding, histone, chromatin, and cyclin genes in G1/S phases (19-1, 19-7, 19-14, 19-18, respectively). In the G1 and S phases, one can see preceding interactions

between DNA replication initiation genes in M/G1 and DNA replication and repair genes in S (6-2), and mating genes in M/G1 and DNA replication genes in S (20-17). Finally in the case of G2 and M phases, one finds preceding interactions between histone genes and cyclin genes of the G2/M transition (7-9). Other biologically supported edges represent intra-phase genetic relationships.

5 Discussion and Future Work

We have presented a framework for learning a regulatory network from sparsely sampled time series gene expression data using a continuous probabilistic model of temporal cluster expression and a novel mutual information score. Our method avoids discretizing the expression values, thus allowing a more sensitive mutual information score. The mutual information scores were used with a **tree-learning algorithm** to produce a biologically plausible multi-stage temporal network model for the yeast cell cycle.

While the current dataset has presented interesting results, many other time series datasets are becoming available and can be studied with the method developed here. Furthermore, experiments with other graph structures are of interest. **Here, a Chow-Liu algorithm was constrained to learn multi-stage sub-graphs.** This structure learning is appropriate for the pairwise mutual information scores that were computed between the various clusters. However, higher order mutual information measures may be used which capture more subtle dependencies in the data. However, unlike Chow-Liu and pairwise mutual information, these higher order variants typically involve intractable learning algorithms and careful approximation strategies must be investigated. We are exploring means to characterize the nature of the edges in our graphical model, such as applying scoring functions that will inform us of the strength and sign of the interactions between nodes in the model. Finally, we note that our method also produces continuous probabilistic models for individual genes. In combination with other data types that inform regulatory interactions (for example, genome location analysis and binding motif data), these individual gene models could be used to learn a finer and fuller structure of the cell cycle regulatory network.

Acknowledgments: We would like to thank Chris Wiggins, Harmen Bussemaker, Jiri Zavadil, and Erwin Böttinger for useful conversations and Dimitris Anastassiou for providing additional computational resources. CL is partially supported by NIH grant LM07276-02.

Appendix: Annealed EM-based Clustering

For convenience, we outline the expectation maximization clustering algorithm from [1] which we modify with deterministic annealing. For each gene i , both the **cluster assignment Z_i** and the **gene-specific variation γ_{ij}** from cluster mean μ_j are treated as hidden variables, using the model

$$\begin{aligned}
 & P(\mathbf{Y}_i, \gamma_{ij} | Z_i = j, \mu_j, \Gamma_j, \sigma^2) \\
 &= \frac{1}{(2\pi)^{\frac{m+q}{2}} |\Gamma_j|^{\frac{1}{2}} \sigma^m} \exp(-(\mathbf{Y}_i - S(\mu_j + \gamma_{ij}))^t (\mathbf{Y}_i - S(\mu_j + \gamma_{ij})) / (2\sigma^2)) \exp(-\frac{1}{2} \gamma_{ij}^t \Gamma_j^{-1} \gamma_{ij})
 \end{aligned}$$

In the E-step, we calculate $p(j|i) = \frac{(p_j P(\mathbf{Y}_i, \gamma_{ij} | Z_i = j, \mu_j, \Gamma_j, \sigma^2))^\beta}{\sum_k (p_k P(\mathbf{Y}_i, \gamma_{ik} | Z_i = k, \mu_k, \Gamma_k, \sigma^2))^\beta}$, where p_k are prior probabilities on the cluster assignments and β is the annealing parameter. Still in the E-step,

we calculate expectations

$$\widehat{\boldsymbol{\gamma}}_{ij} = (\sigma^2 \Gamma^{-1} + S^t S)^{-1} S^t (\mathbf{Y}_i - S \boldsymbol{\mu}_j) \quad \text{and} \quad \widehat{\boldsymbol{\gamma}}_{ij}^t \widehat{\boldsymbol{\gamma}}_{ij} = \widehat{\boldsymbol{\gamma}}_{ij}^t \widehat{\boldsymbol{\gamma}}_{ij} + (\Gamma_j^{-1} + S^t S / \sigma^2)^{-1}.$$

In the M-step, we update parameters with

$$\sigma^2 = \frac{\sum_i \sum_j p(j|i) (\mathbf{Y}_i - S(\boldsymbol{\mu}_j + \widehat{\boldsymbol{\gamma}}_{ij}))^t (\mathbf{Y}_i - S(\boldsymbol{\mu}_j + \widehat{\boldsymbol{\gamma}}_{ij}))}{mN}, \quad \Gamma_j = \frac{\sum_i p(j|i) \widehat{\boldsymbol{\gamma}}_{ij}^t \widehat{\boldsymbol{\gamma}}_{ij}}{\sum_i p(j|i)}$$

$$\boldsymbol{\mu}_j = \left(\sum_i p(j|i) S^t S \right)^{-1} \left(\sum_i p(j|i) S^t (\mathbf{Y}_i - S \widehat{\boldsymbol{\gamma}}_{ij}) \right), \quad p_j = \frac{1}{N} \sum_i p(j|i).$$

We start with the annealing parameter $0 < \beta \ll 1$ and gradually let β approach 1 as we iterate E-steps and M-steps. This technique helps the algorithm avoid poor local maxima of the complete log likelihood function.

References

- [1] Z. Bar-Joseph, G. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon. A new approach to analyzing gene expression time series data. *Proceedings of RECOMB 2002*, 2002.
- [2] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.
- [3] T. S. Spellman et al. Comprehensive identification of cell cycle-related genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. of the Cell*, 9:3273–3297, 1998.
- [4] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing*, 2001.
- [5] G. James and T. Hastie. Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society*, 2001.
- [6] E. Mjolsness, T. Mann, R. Casta no, and B. Wold. From coexpression to coregulation: An approach to inferring transcriptional regulation amd gene classes from large-scale expression data. *Advances in Neural Information Processing Systems*, 12:928–934, 2000.
- [7] I. M. Ong, D. Page, and J. D. Glasner. Modelling pathways in *E. coli* from time series expression profiles. *Proceedings of the Tenth International Conference on Intelligent Systems for Molecular Biology*, 2002.
- [8] D. Pe’er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Proceedings of the Ninth International Conference on Intelligent Systems for Molecular Biology*, 2001.
- [9] I. Simon, J. Barnett, N. Hannett, C. Harbison, N. Rinaldi, T. Volkert, J. Wyrick, J. Zeitlinger, D. Gifford, T. Jaakkola, and R. Young. Regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106:697–708, 2001.
- [10] H. Toh and K. Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18(2):287–297, 2002.