BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–8

Continuous Hidden Process Model for Time Series Expression Experiments

Yanxin Shi [a], Michael Klustein [b], Itamar Simon [b], Tom Mitchell [a], Ziv Bar-Joseph [a*]

[a] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213, USA,
[b]Department of Molecular Biology, Hebrew University Medical School, Jerusalem, Israel, 91120

ABSTRACT

Motivation: When analyzing expression experiments researchers are often interested in identifying the set of biological processes that are up or down regulated under the experimental condition studied. Current approaches, including clustering expression profiles and averaging the expression profiles of genes known to participate in specific processes, fail to provide an accurate estimate of the activity levels of many biological processes.

Results: We introduce a probabilistic Continuous Hidden Process Model (CHPM) for time series expression data. CHPM can simultaneously determine the most probable assignment of genes to processes and the level of activation of these processes over time. To estimate model parameters CHPM uses multiple time series datasets and incorporates prior biological knowledge. Applying CHPM to yeast expression data, we show that our algorithm produces more accurate functional assignments for genes compared to other expression analysis methods. The inferred process activity levels can be used to study the relationships between biological processes. We also report new biological experiments confirming some of the process activity levels predicted by CHPM.

Availability: A Java implementation is available at $http : \backslash\backslash www.cs.cmu.edu\backslash \sim yanxins\backslash chpm$

Contact: zivbj@cs.cmu.edu

1 INTRODUCTION

The Gene Ontology (GO) maps genes to a collection of biological processes and functions. When analyzing microarray expression data researchers often discuss their results in the context of these processes identifying those that are up or down regulated in the condition studied (Newman and Weiner, 2005). This is especially true for time series expression data where the goal is often to determine not only the biological processes that are activated or repressed but also the temporal relationships between these processes (Ramakrishnan *et al.*, 2005). This practice was recently shown to be an effective way to analyze expression data. Tan *et al.* (2003) concluded that while the set of genes determined to be over or under expressed in a specific study may depend on the specific microarray platform used, the set of biological processes identified is often in good agreement across different platforms.

To identify the various biological processes involved, researchers often use one of several clustering algorithms to group genes according to their expression profiles. These clusters are then analyzed to find enriched GO terms, and the results are displayed for each cluster in decreasing significance (Segal and Koller, 2002). An alternative approach starts with the set of genes known to be assigned to each of the processes, and plots their average expression profile to determine the activity level of a process in the condition studied (Smid and Dorssers, 2004). Both of these methods are less than ideal for identifying and quantifying the set of biological processes involved in a specific response. Clustering, while very useful for grouping co-expressed genes, is an unsupervised method which fails to take advantage of prior knowledge regarding which genes are known to be associated with specific processes. As a result, the discovered gene clusters contain genes from many different biological processes, and in many cases genes known to be in the same process are assigned to different clusters (Gibbons and Roth, 2002). On the other hand, beginning with annotations derived from GO, averaging expression levels for each GO process ignores the fact that many genes are associated with multiple biological processes. Averaging expression values from such genes introduces influences from other processes and can artificially increase or decrease the recovered profile for a specific process. In addition, even for genes in the same process some may be required in higher quantity than others and so simple averaging may not be the optimal representation. Of course, this approach also suffers when the information in GO is imperfect or incomplete.

To solve these problems we present the Continuous Hidden Process Model (CHPM). CHPM models the observed gene expression levels as being generated by a combination of multiple biological processes whose activity levels vary over time. We represent the activity levels of processes at each time point as hidden values of random variables which are linked over time. Associated with each process is a set of genes whose expression levels depend both on the activity level of the process, and on a gene-specific weight parameter. We use GO to determine priors for process-gene associations. Using a large collection of time series expression datasets our algorithm utilizes a Kalman filter (Murphy, 2002) to infer the values of the hidden process nodes and to learn the parameters of the model.

The model learned from the data can be used for several purposes. Gene-process associations and the weights of these associations can be used to make predictions regarding the function of genes and to recover the set of biological processes that accounts for the expression profiles. New biological experiments confirmed some of the predictions made by our algorithm regarding process

[*]To whom correspondence should be addressed

activity levels. Using processes activity levels we can also determine temporal relationships between biological processes.

1.1 Related work

While most clustering methods are completely unsupervised, a few clustering methods incorporate gene functions as prior knowledge (Fang *et al.*, 2006; Huang and Pan, 2006). However, these methods still aim to group genes based on expression levels. The resulting clusters often contain genes from several different functional categories making it hard to recover a profile for a specific category. Similarly bi-clustering (Cheng and Church, 2000; Tanay *et al.*, 2002) may also result in several enriched categories for each cluster. SVD can decompose genes into a group of orthogonal "eigen-genes" (Alter *et al.*, 2000). Unlike our method, the biological meaning of each of these "eigen-genes" is not clear and thus they require a post-processing step to infer functional annotations and to determine the significance with which we can associate a gene with a specific function.

Several tools allow users to determine the set of enriched GO categories in a list of genes (Khatri and Draghici, 2005). Many of these methods can also be used to visualize the average expression of genes assigned to a specific category. While these are useful for many applications, they differ from our method in that they rely on the deterministic assignment of genes to categories. Thus, the resulting profiles do not account for multiple functional annotations for a specific gene and also ignore the fact that some genes may be more important to a specific process than others.

Supervised learning algorithms were proposed for annotating genes with new biological functions (Huttenhower *et al.*, 2006; Barutcuoglu *et al.*, 2006). In these methods GO categories are used as class labels and a discriminative model is learned for each of these categories. Unlike these methods we use a temporal generative model to decompose expression data. This allows us to recover not only functional assignments but also the hidden process profile for the category. In addition, it allows our method to deconvolve the relative contributions of several simultaneously active biological processes associated with the same gene.

Segal *et al.* (2003) proposed a method to decompose gene expression to infer activity levels of several abstract processes and simultaneously predict the gene-process association. While their goal is similar to ours, the two methods differ in several important ways. First, unlike our method Segal *et al.* do not use prior knowledge about gene functions. As a result, the inferred processes are not directly associated with a process defined in GO. Another difference is the temporal model we employ. Segal *et al.* treat each experiment as independent (enabling them to use both static and time series data) while our method is restricted to time series data. The advantage of this restriction is that it helps constrain the set of profiles we recover and can help overcome noise as we discuss in Methods. Finally, the running time of the algorithm presented in Segal *et al.* (2003) is exponential in the number of processes. While this allows for a better search, it is not appropriate for the problem we consider in this paper that aims at determining the hidden profiles of more than 100 processes.

In the context of gene regulation Nachman *et al.* (2004) presented a temporal model to infer activity levels of transcription factors (TFs) from time series expression data. Their model is similar to ours in the use of a hidden Markov chain in the transition model. However, it differs in other aspects. Unlike Nachman *et al.* we focus on biological processes rather than regulatory interactions. This leads to differences in our search strategy due to the non-negativity constraint we place on the relationship between genes and processes. In addition, we use prior knowledge as part of our likelihood score whereas Nachman *et al.* use it only to initialize their model.

The term Hidden Process Model (HPM) was introduced by Hutchinson *et al.* (2006) in the context of decoding hidden processes from fMRI data. There are key differences between our method and that of Hutchinson *et al.*. First, in Hutchinson's HPMs model process activities is either "on" or "off." In contrast, we model the activity level of a process with a continuous value. Second, Hutchinson *et al.* do not utilize any prior knowledge associating observed features with hidden processes. Third, Hutchinson *et al.* allow optional input about their timings, whereas we do not assume any prior knowledge regarding the timing of these processes.

2 METHODS

In this section we first elaborate our biological experiments procedure and then introduce the Continuous Hidden Process Model (CHPM) and an associated EM algorithm to learn the model parameters and to infer the activity levels of hidden biological processes. To learn the model CHPM uses time-series microarray expression data from multiple conditions and a set of functional assignments from GO. The learned model assigns a global weight (0 or higher) for the contribution of each process to the expression level of each gene. These weights can be used for inferring functions for unknown genes. The resulting model can also be used to infer the activity levels of the biological processes at each of the time points of a new experiment.

2.1 Budding index experiments

Yeast cells (s288c background strains) were grown to early logarithmic phase (OD 600 nm 0.2-0.5) and H_2O_2 in a final concentration of 0.3 mM was added to the cell culture. 100 cells were sampled every 10 min, between 0 and 120 min, fixed (1% formaldehyde) and observed under a light microscope. Cells were observed to determine bud size and the fraction of cells with no bud, small bud (smaller than one half of the yeast cell) or a large bud was documented. In Figure 4(c) we present the results of one of these experiments. In that figure we annotate the no bud fraction as G1, small bud as S and large bud as G2/M. See supplementary material for complete results.

2.2 Continuous Hidden Process Model

Suppose we have m genes whose expression levels are measured at a series of time points under a variety of experimental conditions (datasets), and suppose we have n biological processes. A Continuous Hidden Process Model (CHPM) defines a probability distribution over time series of gene expression levels in terms of a set of biological processes with unobserved time-varying activity levels. We use a CHPM to estimate which genes are associated with each biological process, with what weights, and to estimate the hidden activity level of each biological process over time.

Let $G_{i,d,t}$ represent the expression level of gene i in dataset d at time t. Similarly, let $P_{j,d,t}$ denote the *activity level* (i.e., intensity) of biological process j in dataset d at time t. Each gene may be associated with zero, one, or several biological processes. Let $w_{i,j}$ denote the non-negative weight with which gene i is associated with biological process j ($w_{i,j} = 0$ if gene i is not associated with process j). A CHPM models the observed expression level for gene i at any time point as the linear superposition of contributions from each of its associated biological processes. More precisely:

$$G_{i,d,t} \sim \begin{cases} \mathcal{N}(0\ ,\ \alpha_d^2) & if\ gene\ i\ is\ not\ associated \\ & with\ any\ biological\ process \\ \mathcal{N}(\sum_{j=1}^{n} w_{i,j} P_{j,d,t}\ ,\ \beta_d^2) & otherwise \end{cases} \tag{1}$$

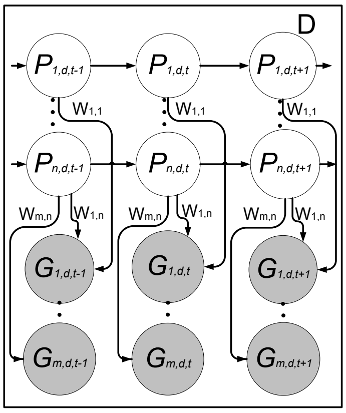


Fig. 1. A graphical model representation of the CHPM. Observed variables are shaded. $P_{j,d,t}$ is the (hidden) activity level of process j at time point t in dataset d. $G_{i,d,t}$ is the observed expression level for gene i at time point t in dataset d. The edge from biological process j to gene i exists if and only if gene i is associated with process j, i.e. $w_{i,j} > 0$, where $w_{i,j}$ represents the weight for each edge. D plates correspond to the D datasets.

As can be seen in Equation 1, the expression profile of a gene over time is a noisy realization of the weighted sum of the profiles of processes with which this gene is associated. At each point in time the expression level is modeled using a Gaussian distribution whose variance is either α_d^2 or β_d^2, depending on whether the gene is believed to be associated with at least one of the biological processes (i.e., depending on whether for gene i the weights $w_{i,j}$ are zero for all j). If the gene is associated with at least one process, then the variance β_d^2 is intended to capture simple noise in the observed expression level. If the gene is not associated with any of the processes under consideration, it may be the case that the gene is participating in some process that is not included in the model. The α_d^2 variance captures this possibility in addition to the usual observational noise.

The evolution over time of activity levels for each biological process is modeled as a hidden Markov chain (see Figure 1). The activity level of the process at time point t (i.e. $P_{j,d,t}$) is dependent on the activity level of this same process at time point $t-1$ (i.e. $P_{j,d,t-1}$). This dependency is modeled as a Gaussian random walk, i.e., $P_{j,d,t} \sim \mathcal{N}(P_{j,d,t-1}, \gamma_d^2)$. The variance γ_d^2 imposes a smoothing effect on the possible change in the process activity level between consecutive time points. The activity level of each process at the very first time point in the d^{th} experimental dataset is modeled as a Gaussian distribution with mean 0 and variance σ_d^2. This dataset-specific variance allows integrating multiple datasets in which the activity levels at the first time point for some processes may differ from 0, for example cell cycle experiments. Figure 1 presents the graphical model used by CHPM.

Note the expression noise parameters α_d^2, β_d^2 are shared across genes within a particular dataset, and the process smoothness term γ_d^2 is shared across processes. However, we estimate different parameter values for each dataset d, to allow for the possibility that noise levels may differ across datasets from different labs using different array platforms. On the other hand, we assume the association between gene i and biological process j is independent of experimental conditions. Therefore, the weight parameters $w_{i,j}$ are shared across all datasets.

2.3 Penalized likelihood function

Given a set of processes, a set of genes, and a collection of gene expression datasets, we train the CHPM by inferring which genes are associated with each process, and by estimating the various CHPM parameters $w_{i,j}$, α_d, β_d, γ_d and σ_d. These estimates are chosen to maximize a penalized complete log-likelihood score subject to the constraint that all weights $w_{i,j}$ be non-negative, and to the constraint that any gene be associated with at most C processes (i.e., that it has at most C non-zero $w_{i,j}$). This second constraint accounts for the fact that it is unlikely for most genes to be associated with more than two processes (only ~10% of yeast genes are associated with more than two processes according to current GO annotations). The constrained penalized log-likelihood score is $Score(\mathbf{o}, \mathbf{h} : W, \theta)$ where $\mathbf{o}, \mathbf{h}$ represent all observed and hidden variables, respectively, and θ includes all model parameters other than the association weights W.

$$\begin{aligned} Score(\mathbf{o}, \mathbf{h} : W, \theta) &= \sum_{d=1}^{D} \log(P(\mathbf{o_d}, \mathbf{h_d} | W, \theta)) - \lambda_1 \sum_{i=1}^{m} \sum_{j=1}^{n} |w_{i,j}| \\ &\quad - \lambda_2 \sum_{i=1}^{m} \sum_{j=1}^{n} \delta(w_{i,j} > 0) \, (E_{i,j}\pi_1 + (1 - E_{i,j})\pi_0) \\ &subject\ to: \ w_{i,j} \geq 0 \quad for\ all\ i, j \\ &\qquad (|\{w_{i,j} | w_{i,j} > 0\}| \leq C) \quad for\ all\ i \end{aligned} \tag{2}$$

Here $\mathbf{o_d}$ and $\mathbf{h_d}$ are the observed expression levels for genes and the unobserved activity levels for biological processes in dataset d, respectively. The score contains two regularization terms. The first one imposes sparsity constraints by limiting the number of non-zero edges. We use an L_1 penalty on the weights leading to selection of a few high weighted edges and setting most other possible gene-process association weights to zero (Tibshirani, 1996). The second term incorporates prior knowledge encoded in GO. $E_{i,j}$ is a binary indicator which is 1 if gene i is annotated with process j in GO and 0 otherwise. $\delta(w_{i,j} > 0)$ is 1 if $w_{i,j} > 0$, and 0 otherwise. π_1 is a penalty term for edges that are supported by GO. π_0 is a similar penalty term for edges that are currently not in GO. Since we are using experimentally validated GO terms and since for a given process j most genes do not belong to j, the penalty for adding an edge not supported by GO is much higher than the penalty for an edge supported by GO, or $\pi_0 >> \pi_1$. Both π_1 and π_0 are user defined and depend of our confidence in the accuracy and completeness of GO annotations. Note we do not penalize for genes assigned by GO but not selected by our model. Such genes may be post-transcriptionally regulated and since our goal is to represent transcriptional models they should not be enforced upon the model. Hence we do not penalize for them.

2.4 Inference and learning for CHPM

To learn the CHPM we use an approximate EM algorithm to attempt to maximize $Score(\mathbf{o}, \mathbf{h} : W, \theta)$. It iteratively performs an E step in which the current model parameters W and θ are used to calculate the expected values of the hidden process activity levels $\mathbf{h}$, followed by an M step in which these activity levels $\mathbf{h}$ are used to re-estimate the model parameters. These two steps are iterated until convergence.

E step: Given all model parameters, we can represent the CHPM using matrix notation. Let $\overrightarrow{P_{d,t}}$ denote an n-dimensional column vector describing the activity of all processes at time point t in dataset d. That is, the j^{th} dimension in this vector is the activity level of process j at time point t in dataset d, i.e. $P_{j,d,t}$. Let $\overrightarrow{G_{d,t}}$ denote the m-dimensional column vector for gene expression levels for this dataset at this time point. That is, the i^{th} dimension in this vector is the expression level of gene i at time point t in dataset d, i.e. $G_{i,d,t}$. Based on our model we can then write:

$$\overrightarrow{P_{d,t}} = \overrightarrow{P_{d,t-1}} + Q_{d,t}, \quad where \ Q_{d,t} \sim \mathcal{N}(\mathbf{0}, \Sigma_{Q_d}); \tag{3}$$

$$\overrightarrow{G_{d,t}} = W \times \overrightarrow{P_{d,t}} + R_{d,t}, \quad where \ R_{d,t} \sim \mathcal{N}(\mathbf{0}, \Sigma_{R_d}); \tag{4}$$

where W is the m-by-n association weight matrix (0 indicates no edge). Here Σ_{Q_d} is a n-by-n diagonal matrix where all diagonal elements are γ_d^2, and determines the probable rate of change of the process activities over time (i.e., $Q_{d,t}$). Similarly, Σ_{R_d} is a m-by-m diagonal matrix where the i^{th} diagonal element is α_d^2 if $w_{i,j} = 0$ for all j and β_d^2 otherwise. It determines the variance in the noise in the observed expression levels (i.e., $R_{d,t}$). All elements in the activity level vector for the first time point follow an independent and identical Normal distribution with mean 0 and variance σ_d^2.

As Equation 3 and 4 show, when the parameters are known the model reduces to the standard Kalman filter (Murphy, 2002) model. Inference in this model can be done efficiently by computing posterior probabilities of the hidden variables $\overrightarrow{P_{d,t}}$ which consist of the $\mathbf{h_d}$ in the *Score* function. The probabilities are all normally-distributed and the computation is tractable because of the conjugacy of the normal distribution.

M step: Given the expected activity levels of biological processes inferred by the E step, we use an approximate algorithm to select new parameters to attempt to maximize the *Score* function in the M step. We can calculate exact solutions for the variance terms γ and σ by zeroing the partial derivatives of the penalized complete log-likelihood of data defined in Equation 2. Fixing the association weights W, we can also estimate the MLE for α and β. See supplementary material for the complete derivation.

Because the *Score* function is constrained to require non-negative $w_{i,j}$ and to allow at most C non-zero weights for each gene, we cannot employ a straightforward process to estimate W. Instead, we first conduct a greedy search to associate processes with each gene, and then solve a constrained optimization problem to obtain estimates for W. To find the optimal set of processes for each gene i, the algorithm first computes the penalized likelihood score in the case that gene i is not associated with any process ($w_{i,j} = 0$ for all j). It next adds one process at a time up to C. Assume we have selected a set of processes c ($|c| < C$). We loop over all processes j where $j \notin c$. For each one we create the set $c_j = c \cup \{j\}$ and solve the following optimization problem which is equivalent to maximizing the penalized complete likelihood score using processes in c_j:

for gene i, minimize:

$$\begin{aligned} F(c_j) &= \frac{1}{D}\sum_{d=1}^{D}\frac{1}{T_d \times \beta_d^2}\sum_{t=1}^{T_d}(G_{i,d,t} - \sum_{j\in c_j} w_{i,j}\hat{P}_{j,d,t})^2 \\ &\quad +\lambda_1 \sum_{j\in c_j}|w_{i,j}| \\ &\quad +\lambda_2 \sum_{j\in c_j}\delta(w_{i,j} > 0)(E_{i,j}\pi_1 + (1-E_{i,j})\pi_0); \end{aligned}$$

subject to:

$$w_{i,j} \geq 0 \quad for\ any\ j \in c_j \tag{5}$$

where T_d is the number of time points in dataset d and $\hat{P}_{j,d,t}$ is the inferred expected activity level of process j at time t in dataset d. All other notations are adopted from Equation 2. This optimization problem is solved using a subspace trust region method (Coleman and Li, 1996).

We choose the process j that minimizes $F(c_j)$ among all processes not in c. If $F(c_j) < F(c)$ we set $c = c \cup \{j\}$ and repeat the above search until c contains C processes. Otherwise we assign to gene i all processes in c using weights computed from the solution to the optimization problem for c. All other weights are set to 0.

2.5 Process selection

The GO database defines a hierarchical structure on the set of biological processes, each of which contains a set of annotated genes. Our goal is to choose a subset of well characterized processes that jointly contain most annotated genes with a small overlap between every pair of selected processes. We are also interested in specific functions (i.e. leaf nodes which convey a more specific function). We thus choose candidate biological processes in the following way. We start by checking leaf processes. If the number of genes associated with a leaf process is more than a pre-defined threshold T this process is selected. Otherwise, we assign the annotated genes to its parent process(es), and delete this leaf (resulting in new leaves). This procedure is repeated until all leaf processes in the current structure are selected. As a post-processing step, we check for overlap between all pairs of selected processes. If any pair has an overlap of over 50% we remove the smaller process.

3 EXPERIMENTS AND RESULTS

We compared the performance of CHPM on both simulated and real expression data with a number of other methods listed below. We note that a number of these methods have very different goals, but here we focus on the goals defined in the introduction (recovering process activity levels and functional assignments of genes to processes).

- Averaging (avg): Process activity levels are obtained by averaging the expression of genes assigned to the process in GO (Ramakrishnan *et al.*, 2005). We rank processes for each gene based on correlation coefficient.
- Singular Value Decomposition (SVD): To predict process assignments genes are first clustered using the eigen-vectors (Alter *et al.*, 2000). We rank GO terms for the genes in each cluster based on their enrichment (using the hypergeometric distribution). Activity levels for enriched biological processes in each cluster are estimated by averaging the expression profiles of genes in the cluster.
- K-means (km) and bi-clustering (bic): As with SVD, genes are first grouped and rankings and profiles are derived using GO enrichment analysis. For bi-clustering we use the implementation in EXPANDER (Shamir *et al.*, 2005) and associate genes with the most significant processes (choosing from all clusters the gene belongs to). We only average expression levels for the time points represented in the cluster.
- Support Vector Machine (SVM): Expression levels are used as feature vectors in a "one-vs.-all" strategy resulting in a multi-class SVM. We used LIBSVM (Chang and Lin, 2001) which outputs a probability estimate for each class based on the distance to the hyperplane. This probability is used to rank processes for each gene. Note that SVM cannot be used directly for process activity inference.

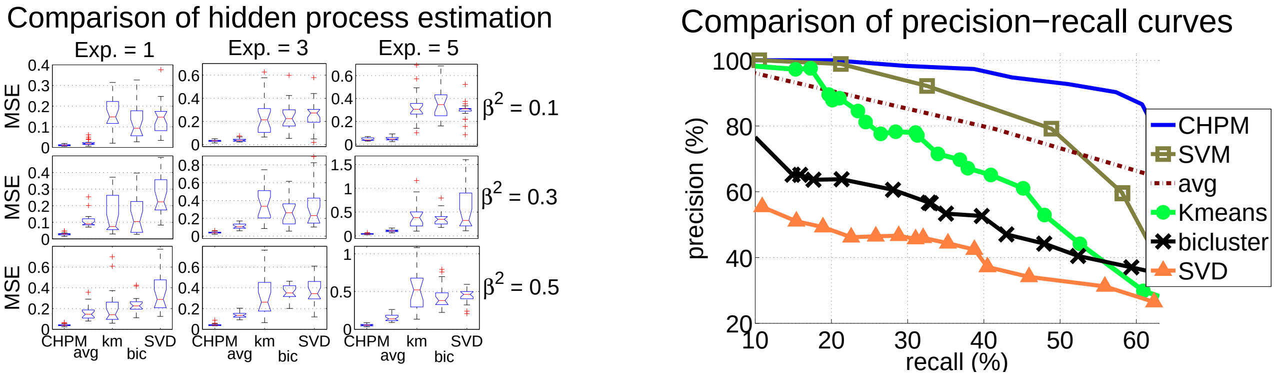
The maximum number C of associated processes for one gene was set to 2 in all experiments below.

3.1 Results on simulated data

Simulated experiment allows a quantitative assessment of model performance based on known underlying activity levels of processes. We first synthesized n biological process activation profiles using a random walk model. Next we randomly generated a number of process-gene association matrices by varying the expected number of processes a gene is associated with. Since none of the other methods model association weights, we intentionally fixed all association weights to be 1. Based on the hidden processes and the association matrix we generated the observed expression values for all genes and added random noise to each time point for each gene.

By varying the noise levels β^2 and the expected number of process-gene associations we can test the influence of various combinations of parameters on the performance of the different methods. For all cases we sampled $n = 100$ processes and $m = 800$ genes. For other parameters, we used the values learned from real data to make the simulation realistic.

Figure 2(a) presents the mean squared error between the true and inferred process activation profiles for each of the methods. The



(a)

(b)

Fig. 2. (a) Mean squared error (MSE) between actual and inferred hidden processes. Rows correspond to a different observation noise level β^2. Columns correspond to different expected number of gene-process associations. Red line is the median. Blue box indicates upper and lower quantiles. The black bars are the range of the MSE. Outliers are plotted by "+". (b) Comparison of precision-recall curves in 8-fold CV.

prior confidence on evidences were set to $\pi_0 = 0.9$ and $\pi_1 = 0.1$ by cross validation. As can be seen, CHPM consistently outperformed all other methods. Among other methods, deterministic averaging (avg) had the best performance. This is partly because all other methods employ prior knowledge only as a post-processing step.

Figure 2(b) shows precision-recall curves of an 8 fold cross validation. In each fold we hid the associations of 100 out of the 800 genes and used all methods to predict the functions for these genes. The precision-recall curves were drawn by increasing the cutoff for the estimated weights, correlation coefficients or significance (depending on the method, see above). Again, CHPM outperformed other baseline methods. In both tasks, clustering and SVD performed poorly. The goal of these methods is to recover patterns in expression data and these seldom correlate with one distinct process.

3.2 Yeast expression data

We applied CHPM to *saccharomyces cerevisiae* microarray time series data collected under 17 experimental conditions including various stresses, cell cycle and DNA damage (see supplementary material for complete list). The number of time points in these datasets ranges from 8 to 24. We first tested our approach using cross-validation. For this, we removed all genes that were not known to participate in any of the 108 processes we modeled (see below). This left 848 genes. To construct the prior evidence matrix E we only used experimentally validated GO annotations. We set the prior knowledge $\pi_0 = 0.9, \pi_1 = 0.1$ indicating our belief in the high quality of these GO assignments.

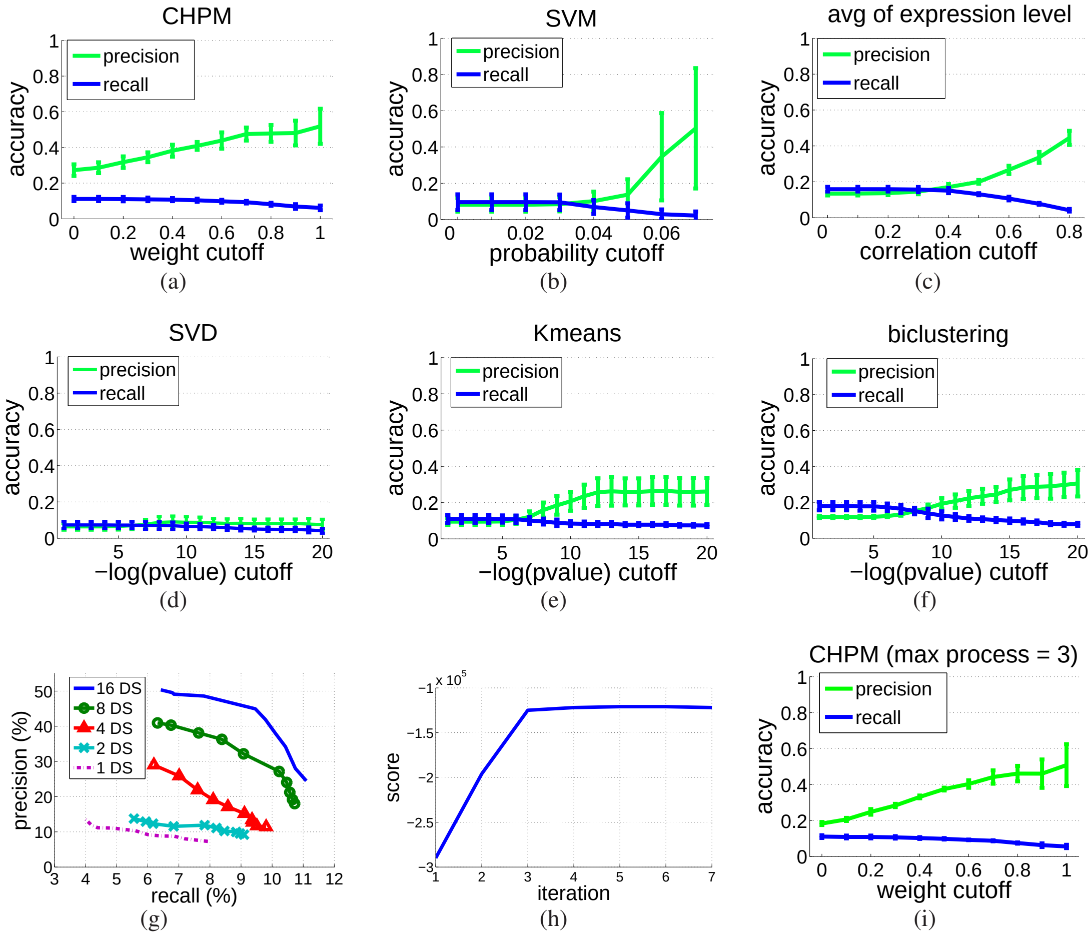
For process selection, we set the threshold T for the minimum number of genes to 25. Applying our selection process to the GO annotations released on 06/06 resulted in 108 biological processes. See supplementary material for complete list.

Predicting gene-process associations: We first tested the ability of CHPM to predict gene-process associations by performing 8-fold cross validation. Figure 3(a-f) plots the precisions and recalls for various weight cutoffs. Similarly, we plot the precisions and recalls for all other methods by varying the cutoff for correlation coefficient or significance.

Since clustering and SVD do not use the gene-process associations as prior knowledge it is not surprising that they did not perform well on this task. For averaging (avg), as the correlation coefficient between a gene and a process increases the precision also increases significantly, though it does not reach the level of the CHPM method. For SVM, as the cutoff for probability increases the precision increases to over 50%. However, the recall decreases rapidly to 2.1%. In contrast, by relying on prior knowledge and by considering the weights of the associations using a generative model, CHPM outperformed all other methods. The precision curve of CHPM increases dramatically when we increase the weight cutoff while the recall does not significantly decrease. Since each gene can only be assigned to up to 2 out of 108 processes, a precision rate of close to 60% is quite impressive. The recall rate is low indicating that more expression data and other sources of data are required for high quality prediction of other genes. Still, the fact that higher weight correlates well with correct functional assignments indicates that the recovered process profiles are a good representation of the underlying profiles.

To test whether more data can improve the performance of our algorithm we measured precision-recall curves using different numbers of datasets. Figure 3(g) shows five curves corresponding to the performance of CHPM with 1, 2, 4, 8 and 16 datasets. Indeed, more datasets improved both precision and recall. Figure 3(h) shows the penalized likelihood scores versus the number of iterations. As can be seen, the score converges quickly, reaching a (local) maximum after three iterations. Note that while this convergence may seem fast, it is a direct result of the fact that we are initializing our model with known GO annotations rather than random initializations that are common in many EM applications.

In Figure 3(i) we present the precision-recall curves for CHPM when setting the maximum number C of processes for each gene to 3. The computation time for on a desktop with 3.2GHz CPU



(a) (b) (c) (d) (e) (f) (g) (h) (i)

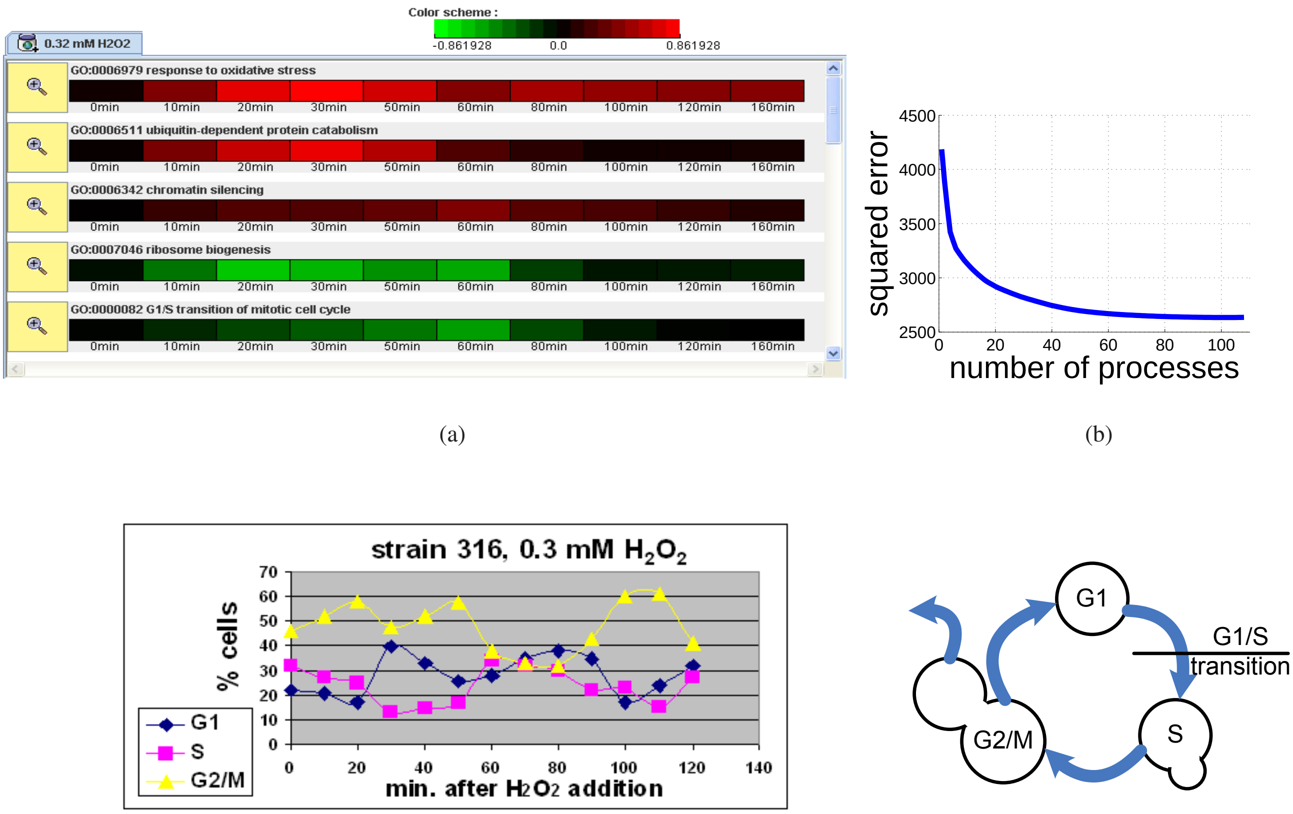
Fig. 3. (a-f) The precision and recall curves and error bars of methods tested using yeast expression data. (g) The precision-recall curves as a function of the number of expression datasets used for CHPM, ranging from 1 to 16 datasets (DS). (h) The penalized likelihood score curve for CHPM versus the number of iterations. (i) Precision and recall curves and error bars for CHPM when setting the maximum number of processes associated with each gene to 3.

and 2.0G RAM increases from approximately 2 to 3 hours when changing from $C = 2$ to $C = 3$. Compared to the case of $C = 2$ (as shown in Figure 3(a)), the recall does not change significantly while the precision slightly drops. Note that most genes in our training data are associated with two or fewer processes in GO which may explain the drop in precision.

Representing expression datasets using process models: For a new expression dataset we can use CHPM (including the learned weight parameters for *all* genes) to infer the subset of processes that are activated or repressed under the condition studied. To choose the significant subset of processes we first infer the activity levels of all processes. Using these levels and the weights we can reconstruct the expression profiles for all genes. Next, we greedily drop one process at a time, minimizing the resulting reconstruction errors for the expression data. We can either set a predefined number of processes to retain or choose based on the average residual error (see Figure 4(b)).

We tested this using the hydrogen peroxide dataset (0.32mM H_2O_2) from Gasch *et al.* (2000). A CHPM learned from 16 other datasets was applied to infer hidden activity levels in the H_2O_2 dataset. Figure 4(b) shows the squared reconstruction error versus the number of processes kept. Figure 4(a) is a screenshot from our software displaying five of the 20 most significant processes. These processes correctly include “response to oxidative stress”, the major process known to be activated under this condition. It also includes “ubiquitin-dependent protein catabolism” which is known to be induced by mild oxidative stress (Gomes-Marcondes and Tisdale, 2002). Repression of ribosomal genes is also well documented under stress condition (Gasch *et al.*, 2000).

Experimental validation of reconstructed profiles: For some biological processes it is possible to carry out experiments that measure phenotypic changes which are assumed to be correlated with the activity of the process. We selected one such process, the “G1/S transition of mitotic cell cycle”. The profile for this process (Figure 4(a)) was predicted to be gradually decreasing (repressed) reaching a low point between 30 and 60 minutes and then recovering to its original (pre-treatment) levels at 120 minutes. A possible explanation for the predicted repression of G1/S genes which is



(a)

(b)

(c)

(d)

Fig. 4. (a) A color screenshot from our software displaying inferred profiles of 5 significant processes for the H_2O_2 experiment. Red represents induced activity, green repressed activity and black no change. The 5 processes are: response to oxidative stress; ubiquitin-dependent protein catabolism; chromatin silencing; ribosome biogenesis and G1/S transition of mitotic cell cycle. (b) Squared error for reconstructed gene expressions as a function of the number of processes kept. (c) Budding index counts. The percentage of “G1”, “S” and “G2/M” cells are plotted at 10 min intervals. (d) Schematic diagram of the yeast cell cycle.

followed by an increase to normal levels is a stress related cell cycle arrest and recovery. To test these predictions we counted the budding index of cells following treatment by H_2O_2. As Figure 4(c) shows, following treatment the percentage of “small bud” (S) cells gradually drops reaching a low point between 30-50 minutes. Combined with the increase in “no bud” (G1) cells at this time interval these findings indicate a G1 arrest. Next, starting at 60 minutes the cell cycle resumes and cells transition from G1 to S. Finally at 120 minutes the percentage of cells in S is close to its pre-treatment percentage.

The minor difference between the timing of these events (a low point at 50 minutes for budding index vs. 60 minutes for the expression data) can be explained by the different yeast strains used, different labs and differences in sampling rates. However, the general trend (slow repression reaching a low value for ∼20 minutes and then recovery) is the same indicating that our model was able to accurately reconstruct the hidden profile for this process.

As mentioned above, a naive method to estimate the activity level of processes is by using the average expression profiles of genes assigned to that process in GO (termed avg above). To compare CHPM’s prediction to avg for the “G1/S transition of mitotic cell cycle” process we have looked at the average expression levels of genes assigned to this process in GO. The results clearly demonstrate the advantage of our generative model. Unlike CHPM which shows a gradual decrease during the first few time points, in the avg result (Figure S1 in Supplementary Material), the first 5 time points (0-50 minutes) are flat. At time point 6 (60 minutes) the avg activity level sharply decreases but it then returns to normal again for time points 7 to 10 (80-160 minutes). In contrast, CHPM shows a more gradual increase which agrees well with an arrest and release behavior. Thus, while the CHPM predictions are supported by the experimental data, the levels reconstructed using the avg method do not always agree with that data.

Exploring process dependencies: An interesting direction in the study of biological processes is exploring the inter-dependencies among these processes. A key challenge for this analysis is the lack of quantitative measurements of the activities for most biological processes. CHPM solves this problem and can thus be used to study these relationships. To test the use of CHPM for this task we inferred the hidden activity profiles for all 108 processes in all 17 experimental conditions. As a post-processing step, the inferred activity levels were discretized into three states: “induced”, “repressed” and “no change”. Next we employed the REVEAL algorithm (Liang, 1998) assuming one time point delay and a maximum fan-in of 2. REVEAL learns inter-slice adjacency matrix given fully observable discrete time series by maximizing the BIC score between parents and child nodes.

In the inferred structure, all processes are connected to themselves (as a result of our Markov model assumptions). However, most processes also contain an incoming edge from a different process. For example, the more general “response to chemical stimulus” was determined to be the parent of “response to DNA damage stimulus” indicating that cells activate a general response which becomes more specific over time. In addition, “maintenance of chromatin architecture” was determined to be a parent of “response to DNA damage”. This might indicate that chromatin maintenance genes identify DNA damage and activate (through a signaling or transcription pathway) the response. See supplementary material for a full list of significant dependencies and supporting references.

4 CONCLUSIONS AND FUTURE WORK

We have presented the Continuous Hidden Process Model (CHPM) and associated algorithms that simultaneously estimate the most probable assignment of genes to biological processes, and the hidden level of activation of each process over time. This CHPM approach integrates data from multiple experiments with prior knowledge of suggested gene-process associations.

Applying our algorithm to yeast we showed that it improves upon current expression based function prediction methods. While function prediction is not the ultimate goal of CHPM, this shows the accuracy of the inferred profiles and weight assignments. The reconstructed profiles agree with current knowledge and can be used to recover the set of important processes for a given experiment. They are also useful for studying relationships between processes. New biological experiments validated the reconstructed profile for one of the processes. CHPM is fully implemented in Java. In the future we would like to apply CHPM to study other organisms including humans, and to enrich the CHPM formalism to directly model dependencies and couplings among different processes.

ACKNOWLEDGEMENT

This work was supported in part by NIH grant NO1 AI-5001 and NSF CAREER award 0448453 to ZBJ.

REFERENCES

Alter, O., Brown, P., and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, **97(18)**, 10101–6.

Barutcuoglu, Z., Schapire, R., and Troyanskaya, O. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, **22(7)**, 830–836.

Chang, C. and Lin, C. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Cheng, Y. and Church, G. (2000). Biclustering of expression data. In *Proceeding of ISMB*, pages 93–103.

Coleman, T. F. and Li, Y. (1996). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, **6**(2), 418–445.

Fang, Z., Yang, J., Li, Y., Luo, Q., and Liu, L. (2006). Knowledge guided analysis of microarray data. *J Biomed Inform.*, **39(4)**, 401–411.

Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D., and Brown, P. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell.*, **11(12)**, 4241–4257.

Gibbons, F. and Roth, F. (2002). Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res.*, **12**, 1574–1581.

Gomes-Marcondes, M. and Tisdale, M. (2002). Induction of protein catabolism and the ubiquitin-proteasome pathway by mild oxidative stress. *Cancer Lett.*, **180(1)**, 69–74.

Huang, D. and Pan, W. (2006). Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, **22**(10), 1259–1268.

Hutchinson, R., Mitchell, T., and Rustandi, I. (2006). Hidden process models. In *Proceedings of ICML*.

Huttenhower, C., Hibbs, M., Myers, C., and Troyanskaya, O. G. (2006). A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, **22**(23), 2890–2897.

Khatri, P. and Draghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, **21**, 3587–3595.

Liang, S. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Proceedings of PSB*.

Murphy, K. (2002). Dynamic bayesian networks: Representation, inference and learning. Ph.D. Thesis, University of California, Berkeley.

Nachman, I., Regev, A., and Friedman, N. (2004). Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, **20 Suppl 1**, I248–I256.

Newman, J. and Weiner, A. (2005). L2L: a simple tool for discovering the hidden significance in microarray expression data. *Genome Biol.*, **6:R81**.

Ramakrishnan, N., Antoniotti, M., and Mishra, B. (2005). Reconstructing formal temporal models of cellular events using the go process ontology. In *Proceeding of Bio-Ontologies SIG Meeting, ISMB*.

Segal, E. and Koller, D. (2002). Probabilistic hierarchical clustering for biological data. In *Proceedings of RECOMB*, pages 273–280.

Segal, E., Battle, A., and Koller, D. (2003). Decomposing gene expression into cellular processes. In *Proceedings of PSB*, volume 8, pages 89–100.

Shamir, R., Maron-Katz, A., Tanay, A., Linhart, C., Steinfeld, I., Sharan, R., Shiloh, Y., and Elkon, R. (2005). EXPANDER - an integrative program suite for microarray data analysis. *BMC Bioinformatics*, **6:232**.

Smid, M. and Dorssers, L. (2004). GO-Mapper: functional analysis of gene expression data using the expression level as a score to evaluate gene ontology terms. *Bioinformatics*, **20(16)**, 2618–2625.

Tan *et al.*, P. (2003). Evaluation of gene expression measurements from commercial microarray platforms. *Nucleic Acids Res.*, **31(19)**, 5676–5684.

Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18(S1)**, S136–S144.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J Royal Statist Soc B.*, **58**(1), 267–288.