BIOINFORMATICS

Vol. 00 no. 00 2010
Pages 1–6

NET GEM: Network Embedded analysis of Temporal Gene Expression using Mixtures

Vinay Jethava [1], Torbjorn Karfunkel[2], Chiranjib Bhattacharyya [1], Devdatt Dubhashi[2],Goutham N. Vemuri[3]*

[1]Computer Science and Automation Department, Indian Institute of Science, Bangalore, INDIA
[2]Department of Computer Science, Chalmers University of Technology, Göteborg, SWEDEN
[3]Systems Biology, Department of Chemical and Biological Engineering, Chalmers University of Technology, Göteborg, SWEDEN

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation Microarrays have become a routine tool in biological enquiry, geared to measure global gene expression in response to genetic or environmental perturbations. The outcome is a vast amount of data, for which many statistical methods have been developed. These methods identify condition-dependent transcriptional regulation, but are not suited to analyze time-series data. Furthermore, these methods ignore the effect of the underlying interaction network of genes or proteins. Substantial amount of additional information on the interaction dynamics could be obtained if time were to be treated appropriately in the context of the interaction network.

Results We present NET GEM, an algorithm that models temporal gene expression data using Hidden Markov Models, within the constraints of an underlying network. We used NET GEM to identify the dynamics of interactions in wild-type Saccharomyces cerevisiae or its isogenic mutant during continuously changing nutritional environment. NET GEM identified significant changes in the interactions between [to be filled]

Availability: The source code for NET GEMM is available from http://www.website [will fill in before submission. The website mentions having a permanent website. I propose http://129.16.106.142/ where we collect all the tools. We should include a readme.txt that should have info on input files needed, their formatting, choosing parameters, output file description and their interpretation]

Contact: goutham@chalmers.se

1 INTRODUCTION

Microarrays have become a routine tool in the biological inquiry of transcriptional regulation. Gene expression microarrays present a snapshot of the transcriptional profile of all the genes at the time of measurement, resulting in a vast amount of data. Current methods to analyze the data are geared to identify genes that have differential expression in response to genetic or environmental perturbations [REFS]. There are numerous statistical methods to cluster genes, and all of them have been enjoyed varying degree of success in discovering new mechanisms of transcriptional regulation. [(Torbjorn needs to write brief intro to the different methods to cluster genes and theor drawbacks, keeping in mind this is not a review of the methods. I can think of PCA, k-means, Self Organized Maps, supervised learning methods such as Support Vector Machines, anything else I missed. All this needs to be less than 100 words] All these methods have been able to identify the condition-specific gene expression and map the transcription regulatory network by identifying binding sites of the promoters [REFS]. The general idea of all these methods is to cluster genes that have similar expression profile, based on the assumption that co-expressed genes are likely to be co-regulated. An inherent drawback with the presently available methods is that they are not well suited to analyze temporal gene expression data. Time course data on gene expression provides substantial insight into the dynamics of transcriptional regulation, such as sequential events in invoking transcriptional response, any time lag in the process and relate the amplitude of the signal to different perturbations [general REF on time series]. Since time series data have a natural temporal ordering, using conventional methods to analyze time series data will fail to capture the internal structure of the data. Moreover, these methods also fail to consider that observations closer in time are likely to be closer than temporally distant observations[- this is why it is important to include timepoints into the analysis]. Unfortunately, the traditional framework of time series analysis cannot be used to analyze temporal gene expression data due to the small number of observations (time points), owing to cost and/or biological limitations. Recognizing these drawbacks, there has been a growing interest to develop dedicated algorithms that address these drawbacks [REFS]. [Torbjorn to review (less than 100 words) EDGE, STEM, KELLER, TARM, and others that I included in another document. This can also go into the intro of Manuscript 3]

Hidden markov models (HMMs) capture the one-way ordering of time such that observations at any time point are dependent on the previous values [REFS]. HMMs were used to analyze temporal gene expression data [Vinay to review (less than 50 words) previous analysis of microarray data using HMM, only some of the references are provided]. These algorithms do not consider the underlying

*to whom correspondence should be addressed

network topology of regulation. In this article, we present network-embedded analysis of temporal gene expression using mixture models (NET GEM). This method takes into account the functional catogories of the interacting genes during the inference procedure.

[Vinay to complete the sentence]

We applied NET GEM to publicly available time-series gene expression data in Saccharomyces cerevisiae. The available of a highly curated interaction network for this organism makes it an ideal platform for testing the method. We selected two time-series datasets in which the nutritional environment changed with time, one without any genetic perturbations and one with a deletion in the SFP1 transcription factor. The first dataset consists of expression of genes during the gradual transition from carbon starvation to nitrogen starvation in a D-stat under aerobic or anaerobic conditions (Farzadfard et al., 2010). Almost a fourth of the genome underwent transcriptional changes in response to the transition. The dominant transcription factor that brought about these changes was Sfp1, which is known to assimilate signals from the environment and coordinates growth with metabolism (Marion et al., 2004). The second dataset measures the temporal changes in gene expression upon sudden exposure of a strain of S. cerevisiae in which Sfp1 was deleted to glucose (Cipollina et al., 2009). Using these datasets, NET GEM identified [dynamics of interactions between genes, functional categories, etc - Vinay to complete the sentence]

2 METHODS

We describe our approach to the problem of inference of the temporally evolving interactions in an underlying network.

2.1 Dataset

Temporal gene expression datasets were downloaded from Gene Expression Omnibus using accession numbers XXXXX and XXXXX. The two datasets were obtained using Affymetrix platform. The first dataset contained the expression profiles of the genes in S. cerevisiae during the transition from carbon limitation to nitrogen limitation under aerobic or anaerobic conditions. The transition was achieved by gradual increment of glucose availability in the feed to the cells, while keeping the nitrogen concentration constant in a D-stat (Farzadfard et al., 2010). Beyond a certain concentration of glucose, nitrogen became the limiting nutrient. The cells underwent changes related to growth rate as well as metabolism. Analysis of genes whose expression significantly changed indicated that Sfp1 transcription factor played a dominant role in the bringing out the response to transition. In the interest of coherence, we chose a dataset that contains the temporal gene expression profiles in sfp1 deletion mutant and its isogenic reference at different time points after pulsing steadily growing cells with glucose. The data was measured at six time points after the pulse. These data were analyzed using conventional methods, assuming that all time points are independent.

2.2 Construction of the interaction network

The yeast interaction network was constructed using data from previously published datasets. Interactions between proteins that occurred in at least two independent datasets were considered. These interactions were downloaded from BIND [1], MIPS [7], MINT [13], DIP [12] and BioGRID [10] and literature data (5-6 references). The construction of this high-confidence network was described in detail previously (Musigkain et al., 2010). The transcriptional regulatory network (interactions between transcription factors and genes) was downloaded directly from YEASTRACT [11]. The two networks were combined and the nature of interactions was not distinguished for the analysis.

2.3 Known network

We assume that the base underlying network of interactions is known as a graph $G = (V, E)$. Under different conditions, some of the edges are switched on or off, or, more generally set at various levels of activation, $\mathcal{W}$. Also, the same edge may be active in one strain and not in others at any given time point. Thus, we model the state of the network by activation levels, $\mathbf{w}^s(t) = \{w_e^s(t)\}_{e\in E}$, where $w_e^s(t)$ is the activation level of the edge e at time t in strain s.

We use the notation $x_e^s(t)$ to denote the expression levels for genes, i and j, consisting the edge, $e = (i, j) \in E$, for strain, s, at time t. Similarly, $x_e^{1:S}(t_a : t_b)$ denotes the observations for gene expression levels for edge, $e = (i, j)$, over the set of strains, $\{1, \ldots, S\}$; for the time interval, $\{t_a, (t_a + 1), \ldots, t_b\}$.

The observed gene expression levels, $\mathbf{x}^s(t)$, for an strain s at time t are modeled as an Ising system [9]:

$$P\left(\mathbf{x}^s(t)|\mathbf{w}^s(t)\right) = \frac{1}{Z}\exp\left(-\sum_{(i,j)\in E} w_e^s(t)x_i^s(t)x_j^s(t)\right) \quad (1)$$

We assume that the weights evolve according to the markov chain, i.e., $P(\mathbf{w}^s(t+1) = \mathbf{w}_{t+1}|\mathbf{w}^s(t) = \mathbf{w}_t, \mathbf{w}^s(t-1) = \mathbf{w}_{t-1}, \ldots) = P\left(\mathbf{w}^s(t+1) = \mathbf{w}_{t+1}|\mathbf{w}^s(t) = \mathbf{w}_t\right)$ for each strain. However, the strains in the given problem are just slightly altered networks where a few genes have been knocked out of the network. Therefore, most of the network remains the same across strains with only the "close" neighbourhood of the knocked out genes being affected. Thus, if one looks at a "far" edge, e_{far}, the activation strength, $w_{e_{far}}(t)$, should be the same across strains the gene expression data for the edge strains, $x_{e_{far}}^{1:S}(t)$, should be like i.i.d. samples, generated with the same activation strength, $w_{e_{far}}(t)$. In the following discussion, we present a heuristic method which incorporates the ideas mentioned above into the inference problem.

2.3.1 Strain Damping Heuristic We assume that the weights corresponding to the reference strain $\mathbf{w}(t)$ evolve according to a Markov law given by a matrix Q, where $Q(l, m) = P(\mathbf{w}(t+1) = \mathbf{w}_m|\mathbf{w}(t) = \mathbf{w}_l)$ with the property that $\sum_m Q(l, m) = 1$ for all the initial states $\mathbf{w}_l$. For other strains, we assume that the corresponding values are just slightly perturbed; thus

$$w_e^s(t) = w_e(t)\Gamma_e^s \quad (2)$$

The perturbing parameters Γ_e^s are determined deterministically from the underlying network G by

$$\Gamma^s(i, j) = (1 - \gamma_i^s)(1 - \gamma_j^s) \quad (3)$$

where $\gamma_i^s \in [0, 1]$ is a label determined by how far the gene i is in the underlying network to one of the genes knocked out in strain s. We note that the deterministic nature of the damping implies that all strains evolve similarly, i.e., $Q^s = Q\ \forall\ s$. This allows us to incorporate the information for gene expression levels in the different strains while learning the temporal evolution characteristics.

There is a tradeoff between using more sophisticated conditional probability models $p(\mathbf{w}^s(t)|\mathbf{w}^0(t))$ involving more parameters to be learnt and the limited amount of experimental data.EXPAND FURTHER?

We compute the damping factor, γ_i^s, for the genes as follows: If the gene, i, is knocked out in strain s, then we label it as $\gamma_i^s = 0$. Now, we diffuse the labels across the graph such that $\gamma_i^s = \frac{1}{d(i)}\beta\sum_{j\in N(i)}\gamma_j^s$, i.e., the damping factor at a node is the average of the damping factors at its neighbours.

Intutitively, while $\Gamma_e = 0$ for an edge directly incident to one of the knocked out genes, the perturbation gradually damps out with distance from the knocked out gene and for an edge e far away from one of the knocked out genes, $\Gamma_e \approx 1$.

Thus, the problem is a simple HMM [8] with $\mathbf{x}^{1:S}(t)$ and $\mathbf{w}(t)$ as the observation and the hidden variable at time t, and $Q = P(\mathbf{w}(t+1)|\mathbf{w}(t))$ the unknown parameter to be learnt. We note that an application

of the standard forward-backward algorithm to compute the probability distribution over the weight states requires $O(\mathcal{W}^{N_e}T)$ computations, where, $\mathcal{W}$ is the number of possible discrete states for an edge activation strength, N_e is the total number of edges, and T is the time period for which observations are made. This is prohibitively expensive and in the following section, we outline an approximation to solve this problem.

2.3.2 Factorial approximation As noted in the previous section, applying the standard forward backward algorithm is prohibitively expensive for moderate sized graphs. So, we make the simplifying assumption that the weights are evolving *independent* of each other. This leads to the factorial approximation [5, 6] for weight distribution, i.e.,

$$\hat{P}(\mathbf{w}^t) = \prod_{e\in E} P_e(w_e^t) \tag{4}$$

$$\hat{P}(w_e^{t+1} = w_l | w_e^t = w_m) = q_e(l,m) \tag{5}$$

Now, the parameter to be learnt is the transition matrix Q_e for each edge, $e = (i,j) \in E$. We solve the Expectation Maximization (EM)[3] for MAP problem for each edge, e,[1]

$$\begin{aligned} \text{E-step:} \quad & \mathcal{L}(Q_e; Q_e^{(n)}) = E_{w_e^|}[\ln P(\mathbf{x}_e^{1:S}(1:T), \mathbf{w}_e(1:T)|Q_e)] \\ \text{M-step:} \quad & \hat{Q}_e^{(n+1)} = \arg\max_{Q_e}(\ln P(Q_e) + \mathcal{L}(Q_e; Q_e^{(n)})) \end{aligned} \tag{6}$$

where $W_e^|$ is the conditioned variable, $w_e(1:T)|\mathbf{x}_e^{1:S}(1:T), Q_e^{(n)}$ and $Q_e^{(n)}$ is the MAP estimate for the transition probability, Q_e, at the n^{th} iteration of the algorithm. We assume that the data for strain, s, is independently generated based on the Ising model in (1) with the weights that are damped versions (2) of the weights in the original strain. This leads to the observation model specified as:

$$o_e^t(l) = P(x_e^{1:S}(t)|w_e(t) = w_l) \tag{7}$$

$$= \frac{1}{Z}\prod_{s=1}^{S} P(x_e^s(t)|w_e(t) = w_l) \tag{8}$$

$$= \frac{\exp\left\{-w_l\left(\sum_{s=1}^{S} x_i^s(t)x_j^s(t)\Gamma^s(i,j)\right)\right\}}{\sum_{l=1}^{\mathcal{W}}\exp\left\{-w_l\left(\sum_{s=1}^{S} x_i^s(t)x_j^s(t)\Gamma^s(i,j)\right)\right\}} \tag{9}$$

The update equations for computing the forward and backward probability distributions are given as:

$$f_e^{t+1}(m) = P(x_e^{1:S}(1:t), w_e(t+1) = w_m|Q_e^{(n)}) \tag{10}$$

$$= o_e^{t+1}(m)\sum_{l=1}^{\mathcal{W}} f_e^t(l)q_e^{(n)}(l,m) \tag{11}$$

$$b_e^t(l) = P(x_e^{1:S}((t+1):T)|w_e(t) = w_l, Q_e^{(n)}) \tag{12}$$

$$= \sum_{m=1}^{\mathcal{W}} q_e^{(n)}(l,m)o_e^{t+1}(m)b_e^{t+1}(m) \tag{13}$$

and the joint probability as

$$\xi_e^t(l,m) = P(w_e(t,t+1) = (w_l, w_m)|x_e^{1:S}(1:T), Q_e^{(n)}) \tag{14}$$

$$\propto f_e^t(l)q_e^{(n)}(l,m)o_e^{t+1}(m)b^{t+1}(m) \tag{15}$$

Often, there is domain knowledge available which can be incorporated in the form of prior distribution. For example, one may know that most of the edges are inactive about 50% of the time.

2.3.3 Dirichlet prior We model the transition probabilities matrices as dirichlet distributions, such that the prior on the transition probabilities matrix, Q, given the parameter, Θ, is

$$P(\vec{q_l}|\vec{\theta_l}) \sim Dir(q_{l1}, \ldots, q_{l\mathcal{W}}; \theta_{l1}, \ldots, \theta_{l\mathcal{W}}) \tag{16}$$

$$= \frac{1}{B(\vec{\theta_l})}\prod_{m=1}^{\mathcal{W}} q_{lm}^{\theta_{lm}-1} \tag{17}$$

where $\vec{\theta_l} = [\theta_{l1}, \ldots, \theta_{l\mathcal{W}}]$ and $B(\vec{\theta_l})$ is the multinomial beta function [4].

This leads to the update equation for the MAP estimate for transition probabilities, $q(l,m)$, obtained by the maximization step in (6) as

$$q_e^{(n+1)}(l,m) = \frac{(\theta_{lm}-1) + \sum_{t=1}^{T-1}\xi_e^t(l,m)}{\sum_m(\theta_{lm}-1) + \sum_{t=1}^{T-1}\sum_{m=1}^{\mathcal{W}}\xi_e^t(l,m)} \tag{18}$$

2.3.4 Cluster Similarity: We can often group the network interactions into categories based on domain knowledge about the functional classification of genes. For example, one might model the genes that participate in sugar metabolism as one component, while treating genes involved in DNA synthesis as another component. This allows us a simplification that we need to consider only evolution over the *components*(or *clusters*), A_k of edges, which are parameterized by the cluster transition, Q_k for cluster A_k.

Then, the update equations for the transition probability matrix, Q_k, for the cluster, A_k are as follows,

$$q_k^{(n+1)}(l,m) = \frac{(\theta_{lm}^{(k)}-1) + \sum_{e\in A_k}\sum_{t=1}^{T-1}\xi_e^t(l,m)}{(\theta_{ij}^{(k)}-1) + \sum_{e\in A_k}\sum_{t=1}^{T-1}\sum_m\xi_e^t(l,m)} \tag{19}$$

where $\theta^{(k)}$ is the dirichlet parameter matrix for cluster, A_k.

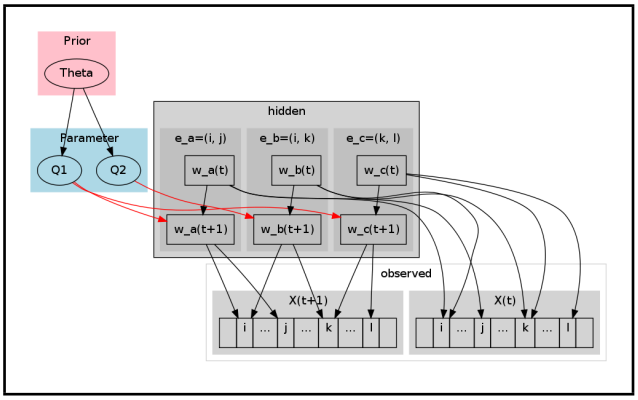

Fig. 1. Single functional classification model. Here, the edges a, c belong to evolution class 1, while edge b belongs to evolution class 2. X_t is the observed gene expression level at time t; $w_e(t)$ is the hidden variable denoting interaction strength for edge e at time t; Q_i are the evolution characteristic for class i; and θ is the prior based on domain knowledge.

Figure 1 shows the graphical model corresponding to the known network model with cluster similarity. We present an extension of the current model to handle multiple functional categories for genes.

2.4 Mixture Model

This section presents our approach to incorporating the functional categories of genes in our analysis. Since, genes belong to multiple categories, a mixture model is a naturally suited model to handle the influence of multiple functional categories in the inference procedure. This allows us to explore the relationship between functional categories and the temporal evolution characteristics of the genes which fall in the same functional

[1] The quality of the factorial approximation is discussed in the appendix

category. We now define the problem concretely. There are H possible gene categories. Each gene can be a member of one or more hierarchical classes, $\mathcal{C} = \{C_1, \ldots, C_H\}$, where the hierarchical class C_h is characterized by evolution matrix, Q_h. The evolution probability matrix, Q_e, for each edge, $e \in E$, is given as

$$Q_e = \sum_{h=1}^{H} \alpha_{e,h} Q_h \tag{20}$$

where $\alpha_{e,h}$ denotes the influence of hierarchical class C_h in the edge, e, such that $\sum_h \alpha_{e,h} = 1$ for all edges $e \in E$. We define the random variable $\mathbf{y}^{1:T} = \{y_e^t\}$ for all edges, $e \in E$, and times, $t = \{1, \ldots, T\}$ where y_e^t denotes the component from which the evolution characteristics are chosen at time t for edge e such that the event $y_e^t = h$ implies that $P(w_e^{t+1} = w_m | w_e^t = w_l, y_e^t = h) = q_h(l, m)$.

We now outline the expectation maximization procedure [2, 6] which iteratively learns the unknown quantity, $\Psi = \{Q_h, \alpha_{e,h}\}$, for $h \in \mathcal{H} = \{1, \ldots, H\}$ and $e \in E$ where Q_h is the class evolution probability matrix for class, C_h, and $\alpha_{e,h}$ is the mixing proportion for edge, e and class C_h; and $\Omega = \{\mathbf{y}^{1:T}, \mathbf{w}^{1:T}\}$ is the hidden variable. Let $\Psi^{(n)} = \{\alpha_{e,h}^{(n)}, Q_h^{(n)}\}$ be the estimates at the n^{th} iteration. Then,

$$\begin{array}{lrcl} \text{E-step:} & \mathcal{L}(\Psi; \Psi^{(n)}) & = & E_{\Omega|}\left[\ln P(\mathbf{x}^{1:S}(1:T), \Omega(1:T) | \Psi(1:T))\right] \\ \text{M-step:} & \hat{\Psi}^{(n+1)} & = & \arg\max_{\Psi}(\ln P(\Psi) + \mathcal{L}(\Psi; \Psi^{(n)})) \end{array} \tag{21}$$

where $\Omega|$ is the conditioned variable $(\mathbf{w}^{1:T}, \mathbf{y}^{1:T} | \mathbf{x}^{1:S}(1:T), \Psi^{(n)})$.

The factorial approximation in the previous section allows us to compute the probability distribution over the edges independently. [2] The observation model, $o_e^t(l) = P(x_e^{1:S}(t) | w_e^t = w_l)$, remains unchanged as in (7)-(9). The forward iterates, $f_e^t(l, h)$ and backward iterates, $b_e^t(l, h)$ can be computed as follows:

$$\begin{aligned} f_e^t(m,h) &= P(x_e^{1:S}(1:t), w_e^t = w_m, y_e^t = h | \Psi_e^{(n)}) && (22) \\ &= P(x_e^{1:S}(t) | w_m) \sum_{w_l} \sum_{h'} \Big[P(y_e^t = h | \alpha^{(n)}) \\ &\times P(w_m | w_e^{t-1} = w_l, y_e^{t-1} = h') \times f_e^{t-1}(l, h') \Big] && (23) \\ &= o_e^t(m) \sum_{l=1}^{\mathcal{W}} \sum_{h'=1}^{H} f_e^{t-1}(l,h') \alpha_h^{(n)} q_{h'}^{(n)}(l,m) && (24) \end{aligned}$$

$$\begin{aligned} b_e^t(m,h) &= P(x_e^{1:S}((t+1):T) | w_e(t) = w_m, y_e^t = h, \Psi_e^{(n)}) && (25) \\ &= \sum_{w_l} \sum_{h'} \Big[P(x_e^{1:S}(t+1) | w_e^{t+1} = w_l) b_e^{t+1}(l,h') \\ &\times P(w_e^{t+1} = w_l | w_m, y_e^t = h) P(y_e^{t+1} = h' | \alpha^{(n)}) \Big] && (26) \\ &= \sum_{m=1}^{\mathcal{W}} \sum_{h'=1}^{H} q_h^{(n)}(m,l) o_e^{t+1}(l) \alpha_{h'}^{(n)} b_e^{t+1}(l,h') && (27) \end{aligned}$$

The conditional probability $P(\Omega_e^t = (w_l, h), \Omega_e^{t+1} = (w_m, h') | \mathbf{x}_e^{1:S}(1:T), \Psi^{(n)})$ denoted by $\xi_e^t(l, m, h, h')$ can be computed as

$$\xi_e^t(l,m,h,h') \propto f_e^t(l,h) \alpha_{h'}^{(n)} q_h^{(n)}(l,m) o_e^{t+1}(m) b_e^{t+1}(m,h') \tag{28}$$

The likelihood term, $\mathcal{L}(\Psi; \Psi^{(n)})$, in (21) can be expressed in terms of the conditioned edge probabilities, ξ_e^t, in (28) as

$$\mathcal{L}(\Psi; \Psi^{(n)}) = \sum_{e \in E} \sum_{t=1}^{T-1} \mathbf{E}_{\xi_e^t}[\ln q_h(l,m) + \ln \alpha_{e,h'}] \tag{29}$$

subject to the constraints

$$\sum_m q_h(l,m) = 1 \quad \forall h \tag{30}$$

$$\sum_h \alpha_{e,h} = 1 \quad \forall e \tag{31}$$

2.4.1 Domain knowledge: We incorporate the effect of the functional classification of genes on the mixture components, $\vec{\alpha}_e$, for an edge, e, by using a dirichlet prior of the form:

$$P(\vec{\alpha}_e) \sim Dir(\alpha_{e,1}, \ldots, \alpha_{e,H}; \gamma_{e,1}, \ldots, \gamma_{e,H}) \tag{32}$$

with the prior parameter, $\gamma_{e,h}$, for the edge, $e = (i, j)$, of the form

$$\gamma_{e,h} = \begin{cases} \gamma_p & \text{if genes } i \text{ or } j \text{ in class } h \\ \gamma_o & \text{otherwise} \end{cases} \tag{33}$$

The maximization step in (21) can be done separately for $q_h(l, m)$ and $\alpha_{e,h'}$ independently. We use the priors in (16) and (32)-(33), and the constraints in (30)-(31) to obtain the following update equations:

$$\alpha_{e,h'}^{(n+1)} = \frac{(\gamma_{e,h'} - 1) + \sum_{t=1}^{T-1} \sum_{l,m,h} \xi_e^t(l,m,h,h')}{\sum_{h'} (\gamma_{e,h'} - 1) + \sum_{t=1}^{T-1} \sum_{l,m,h,h'} \xi_e^t(l,m,h,h')} \tag{34}$$

$$q_h^{(n+1)}(l,m) = \frac{(\theta_{lm} - 1) + \sum_e \sum_{t=1}^{T-1} \sum_{h'} \xi_e^t(l,m,h,h')}{\sum_m (\theta_{lm} - 1) + \sum_e \sum_{t=1}^{T-1} \sum_{m,h'} \xi_e^t(l,m,h,h')} \tag{35}$$

3 EXPERIMENTS

We present the experiments performed on synthetic and actual datasets in this section. We compare the factorial weights and the mixture model against a standard implementation of HMM which learn the evolution characteristic for the system jointly in 3.1 and present the results on a synthetic dataset in 3.2.

3.1 Validation

We choose a synthetic graph with $N = 5$ nodes (genes) and $H = 10$ functional classes. The genes are randomly assigned classes such that each node is a member of $N_{av} = 1.5$ classes on average. The weights take possible values in $\{-1, 1\}$, and the evolution characteristic, Q_e for each weight is a mixture based on the interacting genes. We compare our methods to estimate the overall transition probability matrix, Q_{large}, with the results obtained from an standard implementation of HMM[3] for $N = 20$ random trials.

Figure 2 shows the comparison between different methods with increasing length of the sequence, T. We note that the standard HMM requires longer sequences to get comparable results with factorial and mixture model approach. Figure 3 compares the methods with increasing number of strains present (with at most 1 gene knocked out). We note that the performance of the factorial weights assumption shows a slight improvement with increasing number of strains.

3.2 Synthetic dataset

We generate a "random" graph $G = (V, E)$ with C major components, $\{A_1, \ldots, A_C\}$ with input parameters p_i and p_c, where p_i is the probability, of an edge between two vertices in the same component, and p_c is the probability of an edge between two vertices in different components.

[2] Part of this section may be moved to appendix.

[3] http://people.cs.ubc.ca/ murphyk/Software/HMM/hmm.html

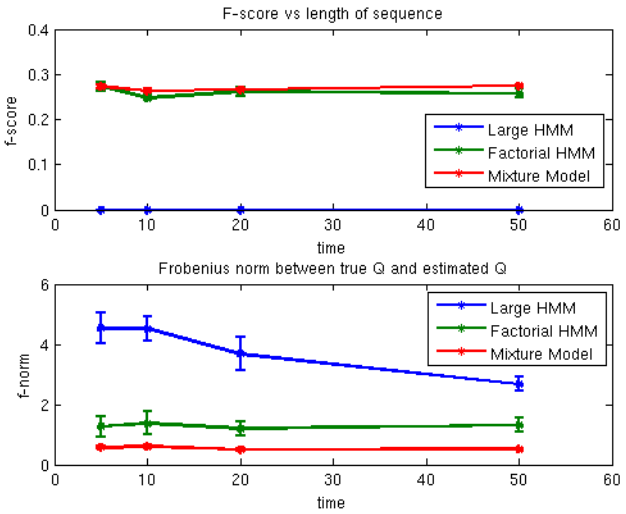

Fig. 2. (a) F-score for the estimated weight evolution vs the true weight evolution sequence with increasing length of sequence, T. (b) Frobenius norm of the difference between the true evolution characteristics for the system and the estimated characteristics.

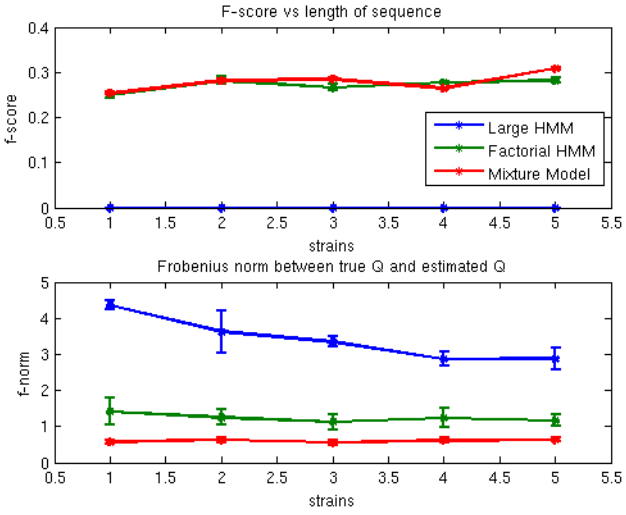

Fig. 3. (a) F-score for the estimated weight evolution vs the true weight evolution sequence with increasing number of strains. (b) Frobenius norm of the difference between the true evolution characteristics for the system and the estimated characteristics.

The activation level, $w_e(t)$ defined on an edge, e, belonging to component, A_k, is a markov chain with transition probability matrix Q_k. The problem is the estimation of the unknown transition probability matrices Q_k for each component, A_k.

We use a noisy dirichlet prior for the estimation as follows:

$$\Theta^{(k)} = Q_k + \mathcal{N}(0, \sigma^2 I) \tag{36}$$

The experiment is conducted for 20 trials with a graph of size $N = 50$ and number of components, C chosen randomly between 2 and 10. Figure 4 shows the F-scores for the experiments done with multiple number of strains.

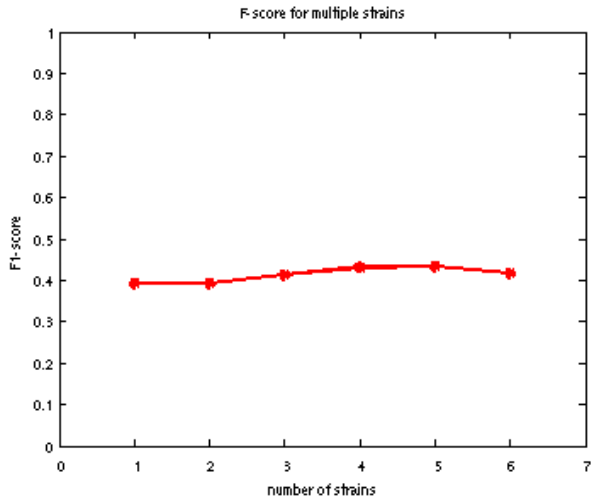

Fig. 4. F-score for the synthetic dataset for multiple strains. We observe that increase in the number of strains provides more information about the activation strengths in the original network, which is visible in the slight increase in the F1-scores. However, since genes are being knocked out in each of the strains, the resulting is not equivalent to i.i.d. samples.

4 CONCLUSION

REFERENCES

[1]G. D. Bader, D. Betel, and C. W. Hogue. Bind: the biomolecular interaction network database. *Nucleic acids research*, 31(1):248–250, January 2003.

[2]Jeff Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1998.

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

[4]Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis, Second Edition (Texts in Statistical Science)*. Chapman & Hall/CRC, 2 edition, July 2003.

[5]Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Machine Learning*, 29(2-3):245–273, 1997.

[6]Geoffrey J. Mclachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, 1 edition, November 1996.

[7]H. W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkötter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nucleic acids research*, 30(1):31–34, January 2002.

[8]L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[9]Le Song, Mladen Kolar, and Eric P. Xing. Keller: estimating time-varying interactions between genes. *Bioinformatics*, 25(12), 2009.

[10]C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Res*, 34(Database issue), January 2006.

[11]M. C. Teixeira, P. Monteiro, P. Jain, S. Tenreiro, A. R. Fernandes, N. P. Mira, M. Alenquer, A. T. Freitas, A. L. Oliveira, and I. Sá-Correia. The yeastract database: a tool for the analysis of transcription regulatory associations in saccharomyces cerevisiae. *Nucleic Acids Res*, 34(Database issue), January 2006.

[12]Ioannis Xenarios, Danny W. Rice, Lukasz Salwinski, Marisa K. Baron, Edward M. Marcotte, and David Eisenberg. Dip: the database of interacting proteins. *Nucl. Acids Res.*, 28(1):289–291, January 2000.

[13]A. Zanzoni. Mint: a molecular interaction database. *FEBS Letters*, 513(1):135–140, February 2002.