Dynamic expression analysis in mutants with temporal expression data

Vinay Jethava [3], Torbjorn Karfunkel[1], Gautham Vemuri[2]*; Chiranjib Bhattacharyya [3], Devdatt Dubhashi[1]

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

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation Microarrays have become a routine tool in many biological labs, resulting in large amounts of datasets containing transcription information in different mutants grown under many conditions. The data from a microarray experiment is a snapshot of the transcription at the time of measurement. Therefore, the conventional methods (log-fold changes, clustering algorithms, etc) have successfully identified transcriptional regulation. Applying these methods to time-series gene expression data inherently assumes that the data points are independent and therefore, losing the temporal evolution of gene expression. Furthermore, many of these methods do not consider the underlying interaction network of genes or proteins they encode for, that plays a key role in the expression of the genes.

Results We present the results on a synthetic dataset

Availability: Source code and dataset available at:

Contact: goutham@chalmers.se

1 INTRODUCTION

Microarrays have replaced the conventional method of determining the expression of a few genes with the ability to measure the true transcriptome rapidly [REFS]. They present a snapshot of the expression of the genes at the time of measurement. Given that there are thousands of genes in a genome, microarrays result in a vast amount of data. The conventional methods to analyze data have been (needs very brief elaboration of the methods here) a.Simple log-fold changes in the expression of genes b.Cluster analysis (PCA, hierarchical clustering, k-means, etc 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 interest in measuring temporal gene expression is increasing again, since dynamic data can potentially reveal causal relationships between the genes [This should be one of the deliverables from our analysis]. In order to extract such hidden information from time-series data, dedicated methods have to be developed. It is only recently that such methods to analyze temporal gene expression data have begun to be developed [REFS] - ...(a) EDGE, (b) STEM, (c) KELLER , (d) others that I missed. The fundamental feature of these methods is ... (i) not take into account the regulatory interactions, (ii) not a true time-series, (iii).. In this article, we present a method (we need to give it a name !!!) that takes into account the above drawbacks by ..

We applied this method to identify the dynamic transcriptional "programs" in different mutants of Saccharomyces cerevisiae during the gradual transition from glucose-limited growth to ammonia-limited growth. During growth in glucose-limited chemostats, S. cerevisiae oxidizes glucose to biomass and CO2, while ammonia-limitation induced partial fermentative behavior, as indicated by the production of ethanol. Gene expression under either of these conditions has been extensively studied during static (steady-state) growth [REFS], rapid shift to excess glucose availability [REFS] as well as gradual transitions [REFS]. The central regulatory protein that governs the adaptation of metabolism and development processes during the transition is the Snf1 kinase [REFS]. The activity of this kinase is prevalent during energy stress, when it repressed energetically expensive reactions such as (synthesis of lipids, proteins, etc) and promotes the reactions that produce ATP (glucose metabolism and oxidative phosphorylation) [REFS]. The Snf1 kinase is activated by phosphorylation by any of the three kinases, Elm1, Sak1 or Tos3 [REFS] (Figure 1). The exact role of these three kinases is still a matter of debate.

Gautham: I will elaborate more in the next round of iteration. The new method allowed us to dissect the role of these kinases at the transcriptional level.

2 METHODS

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

2.1 Known network

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.

*to whom correspondence should be addressed

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 [8]:

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

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.

2.1.1 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

$$w_e^s(t) = w_e(t)\Gamma_e^s \tag{2}$$

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

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

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 [7] 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.

2.1.2 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 [4, 5] 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)[2] for MAP problem for each edge, e,[1]

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

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:

$$\begin{aligned} o_e^t(l) &= P(x_e^{1:S}(t)|w_e(t) = w_l) && (7) \\ &= \frac{1}{Z}\prod_{s=1}^{S} P(x_e^s(t)|w_e(t) = w_l) && (8) \\ &= \frac{\exp\left\{-w_l\left(\sum_{s=1}^S x_i^s(t)x_j^s(t)\Gamma^s(i,j)\right)\right\}}{\sum_{l=1}^{\mathcal{W}}\exp\left\{-w_l\left(\sum_{s=1}^S x_i^s(t)x_j^s(t)\Gamma^s(i,j)\right)\right\}} && (9) \end{aligned}$$

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

$$\begin{aligned} f_e^{t+1}(m) &= P(x_e^{1:S}(1:t), w_e(t+1) = w_m|Q_e^{(n)}) && (10) \\ &= o_e^{t+1}(m)\sum_{l=1}^{\mathcal{W}} f_e^t(l)q_e^{(n)}(l,m) && (11) \\ b_e^t(l) &= P(x_e^{1:S}((t+1):T)|w_e(t) = w_l, Q_e^{(n)}) && (12) \\ &= \sum_{m=1}^{\mathcal{W}} q_e^{(n)}(l,m)o_e^{t+1}(m)b_e^{t+1}(m) && (13) \end{aligned}$$

and the joint probability as

$$\begin{aligned} \xi_e^t(l,m) &= P(w_e(t,t+1) = (w_l,w_m)|x_e^{1:S}(1:T), Q_e^{(n)}) && (14) \\ &\propto f_e^t(l)q_e^{(n)}(l,m)o_e^{t+1}(m)b^{t+1}(m) && (15) \end{aligned}$$

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.

2.1.3 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

$$\begin{aligned} P(\vec{q}_l|\vec{\theta}_l) &\sim Dir(q_{l1}, \ldots, q_{l\mathcal{W}}; \theta_{l1}, \ldots, \theta_{l\mathcal{W}}) && (16) \\ &= \frac{1}{B(\vec{\theta}_l)}\prod_{m=1}^{\mathcal{W}} q_{lm}^{\theta_{lm}-1} && (17) \end{aligned}$$

where $\vec{\theta}_l = [\theta_{l1}, \ldots, \theta_{l\mathcal{W}}]$ and $B(\vec{\theta}_l)$ is the multinomial beta function [3].

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

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

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

2.1.4 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,

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

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

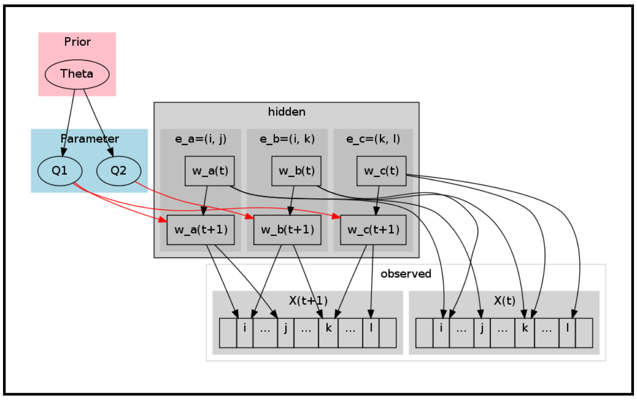

Fig. 1. Single functional classification model. Here, the edges a, c belong to evolution class 1, while edge b belongs to evolution class 2. X_t is the observed gene expression level at time t; $w_e(t)$ is the hidden variable denoting interaction strength for edge e at time t; Q_i are the evolution characteristic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

2.2 Mixture Model

Significant progress has been made towards identifying the functional roles of the genes, resulting in a heirarchical classification of genomes based on their functional roles [6]. We seek to incorporate this information in our inference algorithm. [EXPAND ON GENE CLUSTER IDEA]. We explore the relationship between functional categories and the temporal evolution characteristics of the genes which fall in the same functional 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 [1, 5] 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,

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

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:

$$\begin{aligned} f_e^t(m, h) &= P(x_e^{1:S}(1:t), w_e^t = w_m, y_e^t = h | \Psi_e^{(n)}) && (22) \\ &= P(x_e^{1:S}(t) | w_m) \sum_{w_l} \sum_{h'} \Big[P(y_e^t = h | \alpha^{(n)}) \\ &\times P(w_m | w_e^{t-1} = w_l, y_e^{t-1} = h') \times f_e^{t-1}(l, h') \Big] && (23) \\ &= o_e^t(m) \sum_{l=1}^{\mathcal{W}} \sum_{h'=1}^{H} f_e^{t-1}(l, h') \alpha_h^{(n)} q_{h'}^{(n)}(l, m) && (24) \\ b_e^t(m, h) &= P(x_e^{1:S}((t+1):T) | w_e(t) = w_m, y_e^t = h, \Psi_e^{(n)}) && (25) \\ &= \sum_{w_l} \sum_{h'} \Big[P(x_e^{1:S}(t+1) | w_e^{t+1} = w_l) b_e^{t+1}(l, h') \\ &\times P(w_e^{t+1} = w_l | w_m, y_e^t = h) P(y_e^{t+1} = h' | \alpha^{(n)}) \Big] && (26) \\ &= \sum_{m=1}^{\mathcal{W}} \sum_{h'=1}^{H} q_h^{(n)}(m, l) o_e^{t+1}(l) \alpha_{h'}^{(n)} b_e^{t+1}(l, h') && (27) \end{aligned}$$

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

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

subject to the constraints

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

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

2.2.1 Domain knowledge: We incorporate the effect of the functional classification of genes [6] on the mixture components, $\vec{\alpha}_e$, for an edge, e, by

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

using a dirichlet prior of the form:

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

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:[3]

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

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

3 EXPERIMENTS

We present the experiments performed on synthetic and actual datasets in this section.

3.1 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.

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 2 shows the F-scores for the experiments done with multiple number of strains. Figure **??** shows the results by employing the factorial assumption compared against the results obtained using the standard HMM.

4 CONCLUSION

REFERENCES

[1]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.

[2]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.

[3]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.

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

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

[6]Hans-Werner Mewes, Kaj Albermann, Klaus Heumann, S. Lieb, and Friedhelm Pfeiffer. Mips: a database for protein sequences, homology data and yeast genome information. *Nucleic Acids Research*, 25(1):28–30, 1997.

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

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

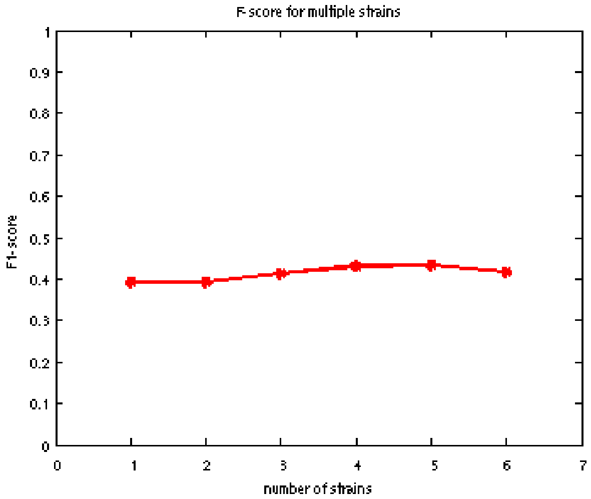

Fig. 2. F-score for the synthetic dataset for multiple strains. We observe that increase in the number of strains provides more information about the activation strengths in the original network, which is visible in the slight increase in the F1-scores. However, since genes are being knocked out in each of the strains, the resulting is not equivalent to i.i.d. samples.

[3] can be expanded lagrangian partial derivative