BIOINFORMATICS

Vol. 25 ISMB 2009, pages i128–i136
doi:10.1093/bioinformatics/btp192

KELLER: estimating time-varying interactions between genes

Le Song, Mladen Kolar and Eric P. Xing*

School of Computer Science, Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ABSTRACT

Motivation: Gene regulatory networks underlying temporal processes, such as the cell cycle or the life cycle of an organism, can exhibit significant topological changes to facilitate the underlying dynamic regulatory functions. Thus, it is essential to develop methods that capture the temporal evolution of the regulatory networks. These methods will be an enabling first step for studying the driving forces underlying the dynamic gene regulation circuitry and predicting the future network structures in response to internal and external stimuli.

Results: We introduce a kernel-reweighted logistic regression method (KELLER) for reverse engineering the dynamic interactions between genes based on their time series of expression values. We apply the proposed method to estimate the latent sequence of temporal rewiring networks of 588 genes involved in the developmental process during the life cycle of *Drosophila melanogaster*. Our results offer the first glimpse into the temporal evolution of gene networks in a living organism during its full developmental course. Our results also show that many genes exhibit distinctive functions at different stages along the developmental cycle.

Availability: Source codes and relevant data will be made available at http://www.sailing.cs.cmu.edu/keller

Contact: epxing@cs.cmu.edu

1 INTRODUCTION

Many biological networks bear remarkable similarities in terms of global topological characteristics, such as scale-free and small-world properties, to various other networks in nature, such as social networks, albeit with different characteristic coefficients (Barabasi and Albert, 1999). Furthermore, it was observed that the average clustering factor of real biological networks is significantly larger than that of random networks of equivalent size and degree distribution (Barabasi and Oltvai, 2004); and biological networks are characterized by their intrinsic modularities (Vászquez *et al.*, 2004), which reflect presence of physically and/or functionally linked molecules that work synergistically to achieve a relatively autonomous functionality. These studies have led to numerous advances towards uncovering the organizational principles and functional properties of biological networks, and even identification of new regulatory events (Basso *et al.*, 2005).

However, most such results are based on analyses of *static networks*, i.e. networks with invariant topology over a given set of molecules. One example is a protein–protein interaction (PPI) network over all proteins of an organism, regardless of the conditions under which individual interactions may take place. Another example is a single-gene network inferred from microarray data even though the samples may be collected over a time course or multiple conditions. A major challenge in systems biology is to understand and model, quantitatively, the dynamic topological and functional properties of cellular networks, such as the rewiring of transcriptional regulatory circuitry and signal transduction pathways that control behaviors of a cell.

Over the course of a cellular process, such as a cell cycle or an immune response, there may exist multiple underlying 'themes' that determine the functionalities of each molecule and their relationships to each other, and such themes are dynamical and stochastic. As a result, the molecular networks at each time point are context-dependent and can undergo systematic rewiring rather than being invariant over time, as assumed in most current biological network studies. Indeed, in a seminal study by Luscombe *et al.* (2004), it was shown that the 'active regulatory paths' in a gene-regulatory network of *Saccharomyces cerevisiae* exhibit dramatic topological changes and hub transience during a temporal cellular process, or in response to diverse stimuli. However, the exact mechanisms underlying this phenomena remain poorly understood. We refer to this time- or condition-specific 'active parts' of the biological circuitry as the *active time-evolving network*, or simply, *time-varying network*. Our goal is to recover the latent time-evolving network of gene interactions from microarray time course.

What prevents us from an in-depth investigation of the mechanisms that drive the temporal rewiring of biological networks during various cellular and physiological processes? A key technical hurdle we face is the unavailability of *serial snapshots* of the time-evolving rewiring network during a biological process. Current technology does not allow for experimentally determining a series of time-specific networks, for a realistic dynamic biological system, based on techniques such as yeast two-hybrid or ChIP-chip systems; on the other hand, use of computational methods, such as structural learning algorithms for Bayesian networks, is also difficult because we can only obtain a few observations of gene expressions at each time point which leads to serious statistical issues in the recovered networks.

How can one derive a temporal sequence of time-varying networks for each time point based on only one or at most a few measurements of node-states at each time point? If we follow the naive assumption that each temporal snapshot of gene expressions is from a completely different network, this task would be statistically impossible because our estimator (from only the observations at the time point in question) would suffer from extremely high variance due to sample scarcity. Previous methods would instead pool observations from all time points together and infer a single 'average' network (Basso *et al.*, 2005; Friedman *et al.*, 2000; Ong, 2002), which means they choose to ignore network rewiring and simply assume that the observations are independently and

*To whom correspondence should be addressed.

identically distributed. To our knowledge, no method is currently available for genome-wide reverse engineering of time-varying networks underlying biological processes, with temporal resolution up to every single time point based on measurements of gene expressions.

In this article, we propose kernel-reweighted logistic regression (*KELLER*), a new machine learning algorithm for recovering time-varying networks on a *fixed* set of genes from time series of expression values. KELLER stems from the acronym KERWLLOR, which stands for *KErnel ReWeighed* l_1*-regularized LOgistic Regression*. Our key assumption is that the time-evolving networks underlying biology processes vary smoothly across time, therefore temporally adjacent networks are likely to share more common edges than temporally distal networks. This assumption allows us to aggregate observations from adjacent time points by reweighting them, and to decompose the problem of estimating time-evolving networks into one of estimating a sequence of separate and static networks. Extending the highly scalable optimization algorithms of ℓ_1-regularized logistic regression, we are able to apply our method to reverse engineer a genome-wide interactions with a temporal resolution up to every single time point.

It is worth emphasizing that earlier algorithms, such as the structure learning algorithms for dynamic Bayesian network (DBN) (Ong, 2002), learns a time-homogeneous dynamic system with fixed node dependencies, which is entirely different from our approach, which aims at snapshots of rewiring network. Our approach is also very different from earlier approaches which start from a priori static networks and then trace time-dependent activities. For example, the *trace-back* algorithm (Luscombe *et al.*, 2004) that enables the revelation of network changes over time in yeast is based on assigning time labels to the edges in a priori static summary network. The Achilles' heel of this approach is that edges that are transient over a short period of time may be missed by the summary static network in the first place. The DREM program (Ernst *et al.*, 2007) reconstructs dynamic regulatory maps by tracking bifurcation points of a regulatory cascade according to the ChIP-chip data over short time course. This is also different from our method, because KELLER aims at recovering the entire time-varying networks, not only the interactions due to protein–DNA binding, from long time series with arbitrary temporal resolution. One related approach is the Tesla algorithm by Ahmed *et al.* (2008). However, Tesla aims at recovering bursty rather than smoothly varying networks.

We apply our method to reverse engineer the time-evolving network between 588 genes involved in the developmental process during the life cycle of *Drosophila melanogaster*. These genes are a subset of the 4028 genes whose expression values are measured in a 66-step time series documented in Arbeitman *et al.* (2002). We validate the biological plausibility of the estimated time-evolving network from various aspects, ranging from the activity of functionally coherent gene sets, to previous experimentally verified interactions between genes, to regulatory cascade involved in nervous system development, and to gene functional enrichment. More importantly, the availability of time-evolving networks gives us the opportunity to further study the rich temporal phenomena underlying the biological processes that is not attainable using the traditional static network. For instance, such a downstream analysis can be a latent functional analysis of the genes in the time-evolving network appeared in Fu *et al.* (2008).

The remainder of the article is structured as follows. In Section 2, we will introduce our kernel reweighted method. In Section 3, we will use synthetic data and a time series of gene expression data collected during the life cycle of *D.melanogaster* to show the advantage as well as biological plausibility of estimating a dynamic network. We conclude the article with a discussion and outlook on future work in Section 4.

2 METHODS

First, we introduce our time-evolving network model for gene expression data, then explain our algorithm for estimating the time-evolving network and finally discuss the statistical property and parameter tuning for our algorithm.

2.1 Modeling time series of gene expression

Microarray profiling can simultaneously measure the abundance of transcripts from tens of thousands of genes. This technology provides a snapshot into the cell at a particular time point in a genome-wide fashion. However, microarray measurements are far from the exact values of the expression levels. First, the samples prepared for microarray experiments are usually a mixture of cells from different tissues and, possibly, at different points of a cell cycle or developmental stage. This means that microarray measurements are only rough estimates of the average expression levels of the mixture. Other sources of noise can also be introduced into the microarray measurements, e.g. during the stage of hybridization, digitization and normalization. Therefore, it is more robust if we only consider the qualitative level of gene expression rather than its actual value. That is we model gene expression as either being upregulated or downregulated. For this reason, we binarize the gene expression levels into $\mathcal{X} := \{-1, 1\}$ (-1 for downregulated and 1 for upregulated). For instance, for cDNA microarray, we can simply threshold at 0 the log ratio of the expression levels to those of the reference, above which a gene is declared to be upregulated and otherwise downregulated.

At a particular time point t, we denote the microarray measurements for p genes as a vector of random variables $X^{(t)} := (X_1^{(t)}, \ldots, X_p^{(t)})^\top \in \mathcal{X}^p$, where we have adopted the convention that the subscripts index the genes and the bracketed superscripts index the time point. We model the distribution of the expression values for these p genes at any given time point t as a binary pair-wise Markov Random Field (MRF):

$$\mathbb{P}_{\theta^{(t)}}(X^{(t)}) := \frac{1}{Z(\theta^{(t)})} \exp\left(\sum_{(u,v)\in\mathcal{E}^{(t)}} \theta_{uv}^{(t)} X_u^{(t)} X_v^{(t)} \right), \tag{1}$$

where $\theta_{uv}^{(t)} = \theta_{vu}^{(t)} \in \mathbb{R}$ is the parameter indicating the strength of undirected interaction between genes u and v; and a $\theta_{uv}^{(t)} = 0$ means that the expression values for genes u and v are conditionally independent given the values of all other genes. Therefore, a MRF is also associated to a network $\mathcal{G}^{(t)}$ with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}^{(t)}$: $\mathcal{V}$ corresponds to the invariant set of genes and hence without the superscript for time; each edge in $\mathcal{E}^{(t)}$ corresponds to an undirected interaction between two genes (and a non-zero $\theta_{uv}^{(t)}$). The difference between $\mathcal{E}^{(t)}$ and $\theta_{uv}^{(t)}$ can be viewed as follows: $\mathcal{E}^{(t)}$ only codes the structure of the model while $\theta_{uv}^{(t)}$ contains all information about the model. Finally, the partition function $Z(\theta^{(t)})$ in a MRF normalizes the model to a distribution.

The dynamic interactions between genes underlying temporal biological processes are reflected in the change of the magnitude of parameter $\theta_{uv}^{(t)}$ across time. In particular, increased values of $\theta_{uv}^{(t)}$ indicate strengthened or emerging interaction between gene u and v, and decreased values indicate weakened or disappearing interaction. Furthermore, we assume that the dynamic interactions between genes vary smoothly across time. Mathematically, this means that the change of $\theta_{uv}^{(t)}$ is small across time, i.e. the difference $|\theta_{uv}^{(t)} - \theta_{uv}^{(t+1)}|$ is upper bounded by a small constant C_θ. In other words, the networks at adjacent time points, $\mathcal{G}^{(t)}$ and $\mathcal{G}^{(t+1)}$ are very similar, i.e. $|\mathcal{E}^{(t)} \cap \mathcal{E}^{(t+1)}|/|\mathcal{E}^{(t)}|$ is lower bounded by a large constant $C_{\mathcal{E}}$ (here, we used $|\cdot|$ to denote the cardinality of a set).

Given time series of gene expression data measured at n time points, $D := \{x^{(t_1)}, \ldots, x^{(t_n)}\}$, our goal is to estimate a temporal sequence of networks $G := \{\mathcal{G}^{(t_1)}, \ldots, \mathcal{G}^{(t_n)}\}$ with each network for 1 time point. Note that, we will focus on estimating the structures of the interactions between genes ($\mathcal{G}^{(t)}$) rather than the detailed strength of these interactions ($\theta^{(t)}$). We hope by restricting our attention to estimating the structure, we can obtain better guarantees in terms of the ability of our algorithm to recover the true underlying interactions between genes. In Sections 2.2 and 2.4, we will provide further explanation on the advantage of focusing on $\mathcal{G}^{(t)}$.

Another important point of clarification is that the interactions between genes we are modeling are the statistical dependencies between their expression levels. This is a common choice for many existing methods, such as the methods by Friedman *et al.* (2000) and Ong (2002). Note that statistical dependency is different from causality, which focuses on directed statistical relations between random variables. In other words, it is more appropriate to view networks from our model as the co-regulation relations between genes. That is, if there is an edge between two genes in the dynamic network at time point t, then the changes of the expression levels of these two genes are likely to be regulated by the same biological process.

2.2 Estimating time-varying network

Two questions need to be addressed when we estimate a time-evolving network. First, what is the objective to optimize and second, what is the algorithmic procedure for the estimation? The first question is addressed in this section and it concerns both the consistency and efficiency of our method while the second question only concerns the efficiency of the algorithm, which we will discuss more in Section 2.3.

First, estimating the parameter vector $\theta^{(t)}$ by maximizing log-likelihood is not practically feasible since the evaluation of the partition function $Z(\theta^{(t)})$ involves a summation of exponential number of terms. Another approach to address this problem is to use a surrogate likelihood function, which can be tractably optimized. However, there is no statistical guarantee on how close an estimate obtained through maximization of a surrogate likelihood is to the true parameter (Banerjee *et al.*, 2008). Therefore, we adapt the neighborhood selection procedure of Wainwright *et al.* (2006) to estimate the time-evolving network $\mathcal{G}^{(t)}$ instead.

Overall, we have designed a method that decomposes the problem of estimating the time-evolving network along two orthogonal axes. The first axis is along the time, where we estimate the network for each time point separately by reweighting the observations accordingly; and the second axis is along the set of genes, where we estimate the neighborhood for each gene separately and then joining these neighborhoods to form the overall network. The additional benefit of such decomposition is that the estimation problem is reduced to a set of identical atomic optimization tasks in Equation (3). In the next section, we will discuss our procedure to solve this atomic optimization task efficiently.

In this new approach, estimating the network $\mathcal{G}^{(t)}$ is equivalent to recovering, for each gene $u \in \mathcal{V}$, its neighborhood of genes that u is interacting with, i.e. $\mathcal{N}^{(t)}(u) := \{v \in \mathcal{V} | (u,v) \in \mathcal{E}^{(t)}\}$. It is intuitive that if we can correctly estimate the neighborhood for all genes u in $\mathcal{V}$, we can recover the network $\mathcal{G}^{(t_i)}$ by joining these neighborhoods. In this alternative view, we can decompose the joint distribution in Equation (1) into a product of conditional distributions, $\mathbb{P}_{\theta^{(t)}}(X_u^{(t)} | X_{\setminus u}^{(t)})$, each of which is the distribution of the expression value of gene u conditioned on the expression values of all other genes (we use $\setminus u$ to denote the set of genes except gene u, i.e. $\setminus u := \mathcal{V} - \{u\}$). In particular, $\mathbb{P}_{\theta^{(t)}}(X_u^{(t)} | X_{\setminus u}^{(t)})$ takes the form of a logistic regression:

$$\mathbb{P}_{\theta_{\setminus u}^{(t)}}(X_u^{(t)} | X_{\setminus u}^{(t)}) = \frac{\exp\left(2X_u^{(t)} \left\langle \theta_{\setminus u}^{(t)}, X_{\setminus u}^{(t)} \right\rangle\right)}{\exp\left(2X_u^{(t)} \left\langle \theta_{\setminus u}^{(t)}, X_{\setminus u}^{(t)} \right\rangle\right) + 1}, \qquad (2)$$

where $\langle a,b \rangle = a^\top b$ denotes inner product and $\theta_{\setminus u}^{(t)} := \{\theta_{uv}^{(t)} \mid v \in \setminus u\}$ is the $(p-1)$-dimensional sub-vector of parameters associated with gene u. The neighborhood $\mathcal{N}^{(t)}(u)$ can be estimated from the sparsity pattern of the sub-vector $\theta_{\setminus u}^{(t)}$. Therefore, estimating the network $\mathcal{G}^{(t)}$ at time point t can be decomposed into p tasks, each for the sub-vector $\theta_{\setminus u}^{(t)}$ corresponding to a gene. For later exposition, we denote the log-likelihood of an observation x under Equation (2) as $\gamma(\theta_{\setminus u}^{(t)}; x) = \log \mathbb{P}_{\theta_{\setminus u}^{(t)}}(x_u | x_{\setminus u})$.

Recall that we assume that the time-evolving network varies smoothly across time. This assumption allows us to borrow information across time by reweighting the observations from different time points and then treating them as if they were *i.i.d.* observations. Intuitively, the weighting should place more emphasis on observations at or near time point t with weights becoming smaller as the observations move further away from time point t. Such reweighting technique has been employed in other tools for time series analyses, such as the short-time Fourier transformation where observations are reweighted before applying the Fourier transformation to capture transient frequency components (Nawab and Quatieri, 1987). In our case, at a given time point t, the weighing is defined as $w^{(t)}(t_i) := K_{h_n}(t - t_i) / \sum_{i=1}^{n} K_{h_n}(t - t_i)$, where $K_{h_n}(\cdot) := K(\cdot / h_n)$ is a symmetric non-negative kernel and h_n is the kernel bandwidth. We used the Gaussian RBF kernel, $K_{h_n}(t) = \exp(-t^2/h_n)$, in our later experiments. Note that multiple measurements at one time point can be trivially handled by assigning them the same weight. We consider multiple measurements to be *i.i.d.* observations.

Additionally, we will assume that the true network is sparse, or that the interactions between genes can be approximated with a sparse model. This sparsity assumption holds well in most cases. For example, a transcription factor only controls a small fraction of target genes under a specific condition (Davidson, 2001). Then, given a time series of gene expression data measured at n time points, $D = \{x^{(t_1)}, \ldots, x^{(t_n)}\}$, we can estimate $\theta_{\setminus u}^{(t)}$ or the neighborhood of $\mathcal{N}^{(t)}(u)$ of gene u at time point t using an ℓ_1 penalized log-likelihood maximization. Equivalently the estimator $\hat{\theta}_{\setminus u}^{(t)}$ is the solution of the following minimization problem:

$$\hat{\theta}_{\setminus u}^{(t)} = \underset{\theta_{\setminus u}^{(t)} \in \mathbb{R}^{p-1}}{\operatorname{argmin}} \left(-\sum_{i=1}^{n} w^{(t)}(t_i) \gamma(\theta_{\setminus u}^{(t)}; x^{(t_i)}) + \lambda \left\| \theta_{\setminus u}^{(t)} \right\|_1 \right), \qquad (3)$$

where $\lambda \geq 0$ is a regularization parameter specified by user that controls the size of the estimated neighborhood, and hence the sparsity of the network. Then, the neighborhood for gene u can be estimated as $\hat{\mathcal{N}}^{(t)}(u) = \{v \in \mathcal{V} \mid \hat{\theta}_{uv}^{(t)} \neq 0\}$, and the network can be estimated by joining these neighborhoods:

$$\hat{\mathcal{E}}^{(t)} = \left\{ (u,v) | v \in \hat{\mathcal{N}}^{(t)}(u) \text{ or } u \in \hat{\mathcal{N}}^{(t)}(v) \right\}. \qquad (4)$$

2.3 Efficient optimization

Estimating time-evolving networks using the decomposition scheme described in previous section requires solving a collection of optimization problems given in Equation (3). In a genome-wide reverse engineering task, there are tens of thousands of genes and hundreds of time points, so one can easily have a million optimization problems. Therefore, it is essential to develop an efficient algorithm for solving the atomic optimization problem in Equation (3), which can then be trivially parallelized across different genes and different time points.

The optimization problem in Equation (3) is an ℓ_1 penalized logistic regression with observation reweighting. This optimization problem has been an active research area in the machine learning community and various methods have been developed, including interior point methods (Koh *et al.*, 2007), trust region newton methods (Lin *et al.*, 2008) and projected gradient methods (Duchi *et al.*, 2008). In this article, we employed a projected gradient method due to its simplicity and efficiency.

The optimization problem in (3) can be equivalently written in a constrained form:

$$\hat{\theta}^{(t)}_{\backslash u} = \underset{\|\theta^{(t)}_{\backslash u}\|_1 \le C_\lambda}{\operatorname{argmin}} \left(-\sum_{i=1}^{n} w^{(t)}(t_i)\gamma(\theta^{(t)}_{\backslash u}; x^{(t_i)}) \right), \tag{5}$$

where C_λ is an upper bound for the ℓ_1 norm of $\theta^{(t)}_{\backslash u}$ and defines a region Ω in which the parameter lies. There is an one-to-one correspondence between C_λ in Equation (5) and λ in Equation (3). In this formulation, the objective $\mathcal{L}(\theta^{(t)}_{\backslash u})$ is a smooth and convex function, and its gradient with respect to $\theta^{(t)}_{\backslash u}$ can be computed simply as $\nabla^{(t)} := \partial\mathcal{L}(\theta^{(t)}_{\backslash u}) = -\sum_{i=1}^{n} w^{(t)}(t_i)\partial\gamma(\theta^{(t)}_{\backslash u})$.

The key idea of a projected gradient method is to update the parameter along the negative gradient direction. After the update, if the parameter lies outside the region Ω, it is projected back into the region Ω, otherwise, we move to the next iteration. The essential step in the algorithm is the efficiency with which we can project the parameter into the region Ω:

$$\theta^{(t)}_{\backslash u} \leftarrow \Pi_\Omega \left(\theta^{(t)}_{\backslash u} - \eta\nabla^{(t)} \right), \tag{6}$$

where $\Pi_\Omega(a) := \operatorname{argmin}_b\{\|a-b\| \,|\, b\in\Omega\}$ is the Euclidean projection of a vector a onto a region Ω. We employed an approach by Duchi *et al.* (2008) which involves only simple operations such as sorting and thresholding for this projection step.

Algorithm 1 gives a summary of the projected gradient method for the optimization problem in Equation (3). Note that the projected gradient algorithm has several internal parameters α, ϵ and σ, which, in our experiments, we set to typical values given in the literature (Bertsekas, 1999).

Algorithm 1 Projected Gradient Method for Equation (5)

```
Input: A time series $D = \{x^{(t_1)}, \ldots, x^{(t_n)}\}$, an upper bound $C_\lambda$
Output: $\hat{\theta}^{(t)}_{\backslash u}$
1: Initialize $\hat{\theta}^{(t)}_{\backslash u}$, $\tilde{\theta}^{(t)}_{\backslash u}$, set $\alpha = 0.1$, $\epsilon = 10^{-6}$, $\sigma = 10^{-2}$
2: repeat
3:     $\hat{\theta}^{(t)}_{\backslash u} \leftarrow \tilde{\theta}^{(t)}_{\backslash u}$, $\eta \leftarrow 1.0$
4:     repeat
5:         $\tilde{\theta}^{(t)}_{\backslash u} \leftarrow \Pi_\Omega\left(\theta^{(t)}_{\backslash u} - \eta\nabla^{(t)}\right)$, $\eta \leftarrow \alpha\eta$
6:     until $\mathcal{L}(\tilde{\theta}^{(t)}_{\backslash u}) - \mathcal{L}(\hat{\theta}^{(t)}_{\backslash u}) \le \sigma\nabla^{(t)}(\tilde{\theta}^{(t)}_{\backslash u} - \hat{\theta}^{(t)}_{\backslash u})$
7: until $\|\hat{\theta}^{(t)}_{\backslash u} - \tilde{\theta}^{(t)}_{\backslash u}\| \le \epsilon$
8: $\hat{\theta}^{(t)}_{\backslash u} \leftarrow \tilde{\theta}^{(t)}_{\backslash u}$
```

2.4 Statistical property

The main topic we discuss here is whether the algorithm described in Section 2.2 can estimate the true underlying time-evolving network correctly. In order to study the statistical guarantees of our algorithm, we need to take three aspects into account. First, a genome-wide reverse engineering task can involve tens of thousands of genes while the number of observations in time series can be quite limited (hundreds at most). Therefore, it is important to study the case in which the dimension p scales with respect to the sample size n, but still allows for recovery of networks. Second, the time-evolving nature of networks adds extra complication to the estimation problem, so, we have to take the amount of change between adjacent networks, $C_\theta := \max_{uv}\|\theta^{(t)}_{uv} - \theta^{(t+1)}_{uv}\|$, into account. Third, the intrinsic properties of the interactions between genes will also affect the correct recovery of the networks. Intuitively, the more complicated interactions the more difficult it is to recover networks, e.g. each gene interacts with a large fraction of other genes. In other words, the maximum size of the neighborhood for a gene $C_\mathcal{N} := \max_{u\in\mathcal{V}}\mathcal{N}(u)$ is also a deciding factor. To our knowledge, none of the earlier methods (Basso *et al.*, 2005; Friedman *et al.*, 2000; Ong, 2002) provide a statistical guarantee for recovered networks or are amenable to such analysis.

In contrast, the method we presented in Section 2.2 is highly amenable to a rigorous statistical analysis. Statistical guarantees have been provided for estimating static networks under the model in Equation (1) (Wainwright *et al.*, 2006) and we can extend them to the time-varying case. A detailed proof of a similar result for our approach is beyond the scope of this article and deserves a full treatment in a separate paper. At a high level, we can show that under a set of suitable conditions over the model, C_θ, $C_\mathcal{N}$, h_n and λ, with high probability, we can recover the true underlying time-evolving network even when the number of genes p is exponential in size of the number of observations n [for details of the proof, see M.Kolar and E.Xing (submitted for publication)]. A different analysis have been provided for time-varying Gaussian graphical models (Zhou *et al.*, 2008), in which the consistency of the interaction strengths is addressed, but not the consistency of the network topology.

2.5 Parameter tuning

The regularization parameter λ controls the sparsity of the estimated networks. Large values of λ result in sparse networks, while small values result in a dense networks that have higher log-likelihood, but more degrees of freedom. We employ the Bayesian Information Criterion (BIC) for choosing λ that trades off between the fit to the data and the model complexity. More specifically, we use an average of the BIC score defined below for each time point t and for each gene u:

$$\mathrm{BIC}(t,u) := \sum_{i=1}^{n} w^{(t)}(t_i)\gamma(\hat{\theta}^{(t)}_{\backslash u}; x^{(t_i)}) - \frac{\log(n)}{2}\mathrm{Nz}(\hat{\theta}^{(t)}_{\backslash u}) \tag{7}$$

where $\mathrm{Nz}(\cdot)$ counts the number of non-zero entries in $\hat{\theta}^{(t)}_{\backslash u}$. Then the final score is $\mathrm{BIC} := 1/n|\mathcal{V}|\sum_{u\in\mathcal{V}}\sum_{j=1}^{n}\mathrm{BIC}(t_j,u)$. A larger BIC score implies a better model.

The bandwidth parameter h_n controls the smoothness of the change in the time-evolving networks. Using wide bandwidths effectively incorporate more observations for estimating each network snapshot, but it also risks missing sharp changes in the network; using narrow bandwidths makes the estimate more sensitive to sharp changes, but this also makes the estimation subject to larger variance due to the reduced effective sample size. In this article, we use a heuristic for tuning the initial scale of the bandwidth parameter h_n: we first form a matrix (d_{ij}) with its entries $d_{ij} := (t_i - t_j)^2$ $(i = \{1,\ldots,n\})$. Then the scale of the bandwidth parameter is set to the median of the entries in (d_{ij}). Intuitively, the bandwidth parameter reflects the characteristic interval between time points. In our simulation experiments, we find that this heuristic provides a good initial guess for h_n, and it is quite close to the value obtained via more exhaustive search.

3 EXPERIMENTS

In this section, we use synthetic data to demonstrate the advantage of estimating a time-evolving network, and we used data collected from *Drosophila* to show that our method, KELLER, can estimate a biologically plausible time-evolving network and reveal some interesting properties of the dynamic interactions between genes.

3.1 Recovering synthetic networks

In this section, we compare KELLER with structural learning of DBN (Friedman *et al.*, 2000) and ℓ_1-regularization logistic regression for static network estimation using synthetic networks. Note that ℓ_1-regularization logistic regression can be obtained from KELLER: we only need to apply a uniform weight $w(t_i) = 1/n$ to all observations and estimate a single network using the same objective as Equation (5).

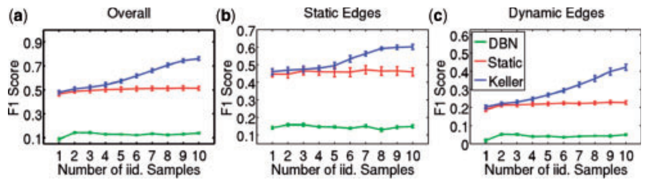


Fig. 1. As we increase the number of *i.i.d.* samples at each time point, KELLER estimating a time-evolving network clearly outperforms DBN and ℓ_1-regularized logistic regression for estimating a static network. Subplot (**a**) displays the performances for the overall networks, while (**b**) and (**c**) display the performance for the static edges and dynamic edges, respectively.

Starting at time t_1, we generate an Erdös–Rényi random graph $\mathcal{G}^{(t_1)}$ of $p=50$ nodes with an average degree of 2. The parameter $\theta_{uv}^{(t_1)}$ for the non-zero edges is chosen uniformly random from the range $[1,2]$. Then, we randomly select 15 edges and gradually decrease their weights to zero in 200 time points. Starting from t_1, we also chose 15 new edges and gradually increase their weights to a random target value between $[1,2]$ in 200 time points. Therefore, in the first 200 discrete time steps, 15 existing edges are deleted, 15 new edges are added and the final network maintains an overall size of 50 edges. We call these 30 time-evolving edges as dynamic edges and the remaining 35 edges as static edges. Then from time point 200, we start the second cycle of deleting 15 edges and adding 15 edges. This cycle is repeated five times, which results in a smooth time-evolving network with $n=1000$ time point. Furthermore, we draw 10 *i.i.d.* observations from the network at each time point and study how the performance of different methods scales with the number of *i.i.d.* observations at each time point.

We evaluate the estimation procedures using an $\mathsf{F1}$ score, which is the harmonic mean of precision (Pre) and recall (Rec), i.e. $\mathsf{F1} := 2*\mathsf{Pre}*\mathsf{Rec}/\mathsf{Pre}+\mathsf{Rec}$. Precision is calculated as $1/n\sum_{i=1}^{n}|\hat{\mathcal{E}}^{(t_i)}\cap\mathcal{E}^{(t_i)}|/|\hat{\mathcal{E}}^{(t_i)}|$, and recall as $1/n\sum_{i=1}^{n}|\hat{\mathcal{E}}^{(t_i)}\cap\mathcal{E}^{(t_i)}|/|\mathcal{E}^{(t_i)}|$. The $\mathsf{F1}$ score is a natural choice of the performance measure as it tries to balance between precision and recall; only when both precision and recall are high can $\mathsf{F1}$ be high. Furthermore, we use an initial bandwidth parameter h_n as explained in Section 2.5, then searched over a grid of parameters ($10^{[-0.5:0.1:0.5]}$ for λ and $h_n \times [0.5,1,2,5,10,50]$ for the bandwidth), and finally chose one that optimizes the BIC criterion defined in Section 2.5. When estimating the static network, we use the same range to search for λ.

The recovery results for the overall time-evolving network, the dynamic and static edges, are presented, respectively, in Figure 1. From the plots, we can see that estimating a static network does not benefit from increasing number of *i.i.d.* observations at all. In contrast, estimating a time-varying network always obtains a better performance and the performance also increases as more observations are available. Note that these results are not surprising since our time-varying network model fits better the data generating process. As time-evolving networks occur very often in biological systems, we expect our method will also have significant advantages in practice.

3.2 Recovering time-evolving interactions between genes in *D.melanogaster*

Over the developmental course of *D.melanogaster*, there exist multiple underlying 'themes' that determine the functionalities of each gene and their relationships to each other, and such themes are dynamical and stochastic. As a result, the gene-regulatory networks at each time point are context-dependent and can undergo systematic rewiring, rather than being invariant over time. In this section, we use KELLER to reverse engineer the dynamic interactions between genes from *D.melanogaster* based on a time series of expression data measured during its full life cycle.

We use microarray gene expression measurements collected by Arbeitman *et al.* (2002) as our input data. In such an experiment, the expression levels of 4028 genes are simultaneously measured at various developmental stages. Particularly, 66 time points are chosen during the full developmental cycle of *D.melanogaster*, spanning across four different stages, i.e. embryonic (1–30 time point), larval (31–40 time point), pupal (41–58 time points) and adult stages (59–66 time points). In this study, we focused on 588 genes that are known to be related to developmental process based on their gene ontologies. We use a regularization parameter of 10^{-2}, and a bandwidth parameter of $0.5\times h_n$ in this experiment (h_n is the median distance as explained in Section 2.5).

In Figure 3a, we plot two different statistics of the reversed engineered gene-regulatory networks as a function of the developmental time point (1–66). The first statistic is the network size as measured by the number of edges; the second is the average local clustering coefficient as defined by Watts and Strogatz (1998). The first statistic measures the overall connectedness of the networks, while the latter measures the average connectedness of the neighborhood local to each gene. For comparison, we normalize both statistics to the range between $[0,1]$. It can be seen that the network size and its local clustering coefficient follow very different trajectories during the developmental cycle. The network size exhibits a wave structure featuring two peaks at mid-embryonic stage and the beginning of pupal stage. Similar pattern of gene activity has also been observed by Arbeitman *et al.* (2002). In contrast, the clustering coefficients of the time-evolving networks drop sharply after the mid-embryonic stage, and they stay low until the start of the adult stage. One explanation is that at the beginning of the developmental process, genes have a more fixed and localized function, and they mainly interact with other genes with similar functions; however, after mid-embryonic stage, genes become more versatile and involved in more diverse roles to serve the need of rapid development; as the organism turns into an adult, its growth slows down and each gene may be restored to its more specialized role.

To illustrate how the network properties change over time, we visualize two networks from mid-embryonic stage (time point 15) and mid-pupal stage (time point 45) in Figure 3b and 3c respectively. Although the size of the two networks are comparable, we can see that there are much clearer local clusters of interacting genes during mid-embryonic stage. To provide a better view of the evolving nature of these clusters, we cluster genes based on the network at time point 1 using spectral clustering, and visualize the gradual disappearance of these clusters in Figure 2. Note that our visualization does not indicate that genes do not form clusters in later developmental stage. Genes may cluster under different groupings, but these clusters cannot be revealed by the visualization since the positions of the genes have been fixed in the visualization.

To judge whether the learned networks make sense biologically, we zoom into three groups of genes functionally related to different stages of development process. In particular, the first group (30 genes) is related to embryonic development; the second group

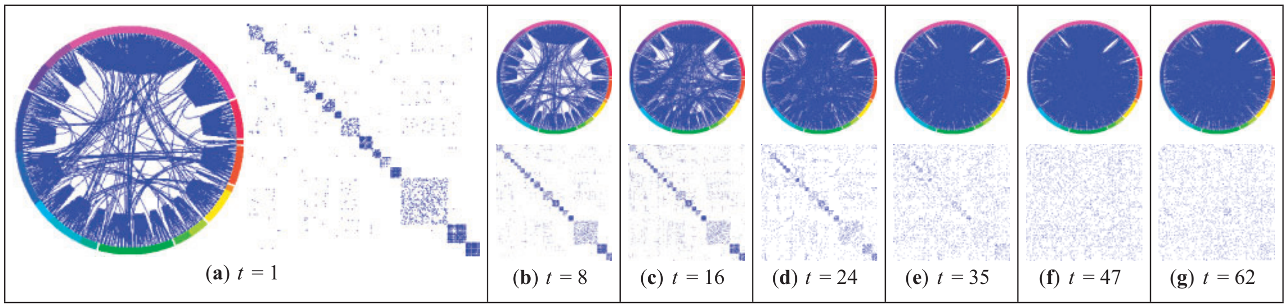


Fig. 2. (**a–g**) Circular plots of the networks (left or top part of each table cell) and dot plots of the adjacency matrices of the networks (right or bottom part of each table cell) at 7 time points. Note that we have clustered the genes according to the network connections at time point 1 and used this clustering result to fix the order of the genes in all plots. In the circular plots, genes are arranged along the outer rim and the colors indicates the boundaries between different clusters (20 clusters in total). Furthermore, we have added curvature to the edges such that connections within and between clusters can be seen more clearly.

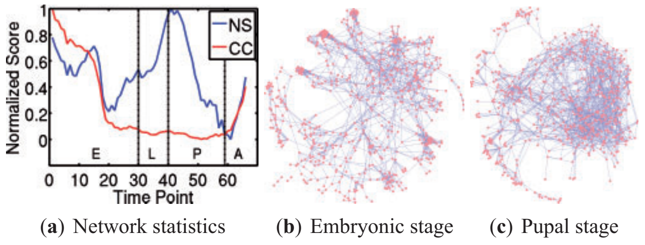


Fig. 3. Characteristic of the time-evolving networks estimated for the genes related to developmental process. (**a**) Plot of two network statistics as functions of development time line (NS, network size; CC, clustering coefficient). (**b** and **c**) visualization of two example of networks from different time point. We can see that network size can evolve in a very different way from the local clustering coefficient.

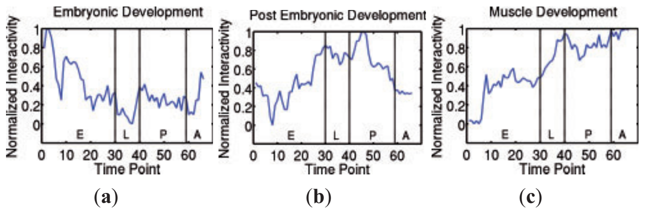
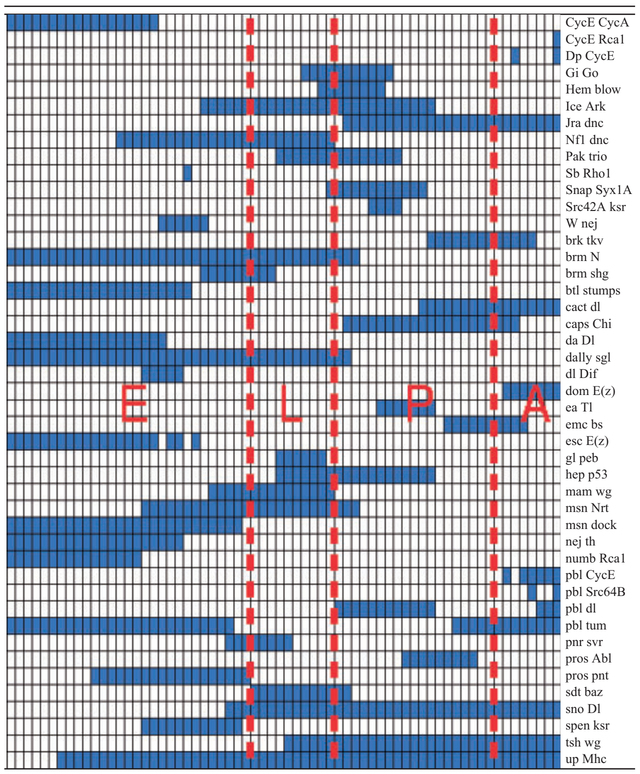


Fig. 4. Interactivity of three groups of genes related to (**a**) embryonic development; (**b**) post-embryonic development; and (**c**) muscle development. The higher the interactivity, the more active the group of genes. We see that the interactivity of the three groups is very consistent with their functional annotation.

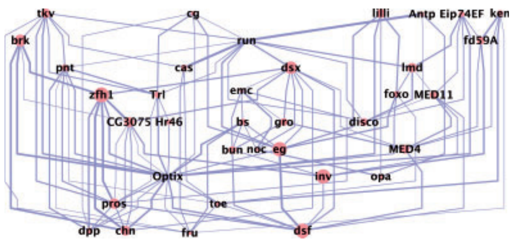
(27 genes) is related to post-embryonic development; and the third group (25 genes) is related to muscle development. (The genes are assigned to their respective groups according to their ontology labels.) We used interactivity, which is the total number of edges a group of genes is connected to, to describe the activity of each group genes. In Figure 4, we plotted the time courses of interactivity for the three groups, respectively. For comparison, we normalize all scores to the range of [0, 1]. We see that the time courses have a nice correspondence with their supposed roles. For instance, embryonic development genes have the highest interactivity during embryonic stage, and post-embryonic genes increase their interactivity during larval and pupal stage. The muscle development genes are less specific to certain developmental stages, since they are needed across the developmental cycle. However, we see its increased activity when the organism approaches its adult stage where muscle development becomes increasingly important.

Table 1. Timeline of 45 known gene interactions

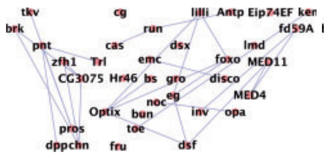


Each cell in the plot corresponds to one gene pair of gene interaction at one specific time point. The cells in each row are ordered according to their time point, ranging from embryonic stage (E) to larval stage (L), to pupal stage (P) and to adult stage (A). Cells colored blue indicate the corresponding interaction listed in the right column is present in the estimated network; blank color indicates the interaction is absent.

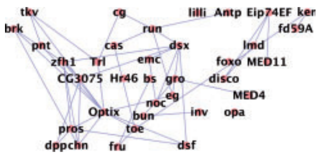
The estimated networks also recover many known interactions between genes. In recovering these known interactions, the time-evolving networks also provide additional information as to when interactions occur during development. In Table 1, we listed these



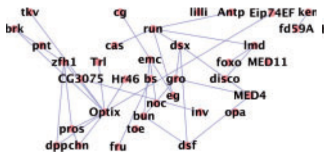
(**a**) Summary network



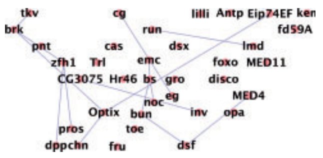
(**b**) Embryonic stage



(**c**) Larval stage



(**d**) Pupal stage



(**e**) Adult stage

Fig. 5. The largest TFs cascade involving 36 TFs. (**a**) The summary network is obtained by summing the networks from all time points. Each node in the network represents a TF, and each edge represents an interaction between them. The width of an edge is proportional to the number of the times the edge is present during the development; the size of a node is proportional to the sum of its edge weights. During different stages of the development, the networks are different, (**b**–**e**) shows representative networks for the embryonic ($t=15$), larval ($t=35$), pupal ($t=49$) and adult stage of the development, respectively ($t=62$).

recovered known interactions and the precise time when they occur. This also provides a way to check whether the learned networks are biologically plausible given the prior knowledge of the actual occurrence of gene interactions. For instance, the interaction between genes msn and dock is related to the regulation of embryonic cell shape and correct targeting of photoreceptor axons. This is consistent with the timeline provided by the time-evolving networks. A second example is the interaction between genes sno and Dl which is related to the development of compound eyes of *Drosophila*. A third example is between genes caps and Chi which are related to wing development during pupal stage. What is most interesting is that the time-evolving networks provide timelines for many other gene interactions that have not yet been verified experimentally. This information will be a useful guide for future experiments.

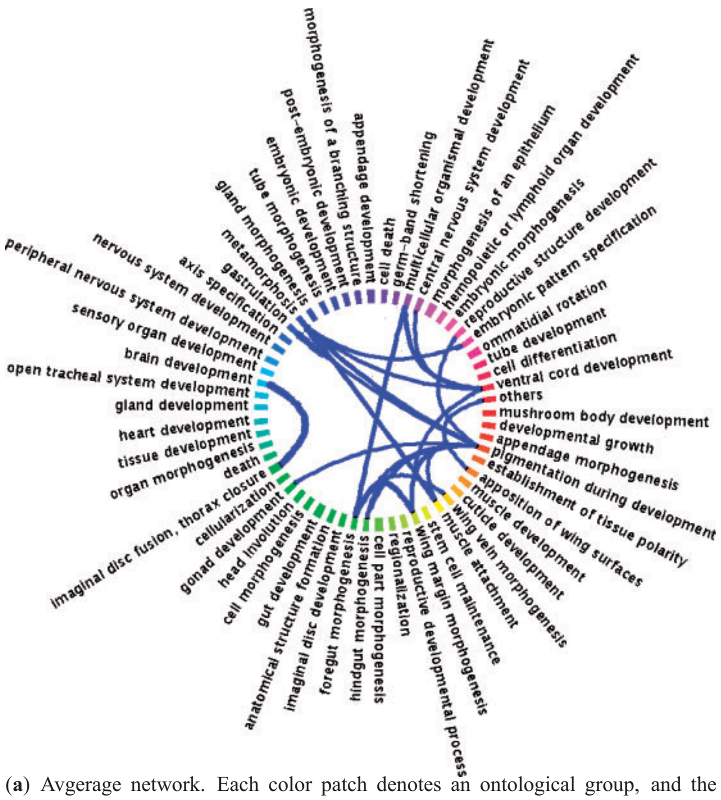
We further study the relations between 130 transcriptional factors (TFs). The network contains several clusters of transcriptional cascades, and we will present in detail the largest TF cascade involving 36 TFs (Fig. 5). This cascade of TFs is functionally very coherent, and many TFs in this network play important roles in the nervous system and eye development. For example, Zn finger homeodomain 1 (zhf1), brinker (brk), charlatan (chn), decapentaplegic (dpp), invected (inv), forkhead box, subgroup 0 (foxo), Optix, eagle (eg), prospero (pros), pointed (pnt), thickveins (tkv), extra macrochaetae (emc), lilliputian (lilli), doublesex (dsx) are all involved in nervous and eye development. Besides functional coherence, the networks also reveal the dynamic nature of gene regulation: some relations are persistent across the full developmental cycle while many others are transient and specific to certain stages of development. For instance, five TFs, brk–pnt–zfh1–pros–dpp, form a long cascade of regulatory relations which are active across the full developmental cycle. Another example is gene Optix which is active across the full developmental cycle and serves as a hub for many other regulatory relations. As for transience of the regulatory relations, TFs to the right of Optix hub reduce their activity as development proceeds to later stages. Furthermore, Optix connects two disjoint cascades of gene regulations to its left and right side after embryonic stage.

The time-evolving networks also provide an overview of the interactions between genes from different functional groups. In Figure 6, we grouped genes according to 58 ontologies and visualized the connectivity between groups. We can see that large topological changes and network rewiring occur between functional groups. Besides expected interactions, the figure also reveals many seemingly unexpected interactions. For instance, during the transition from pupa stage to adult stage, *Drosophila* is undergoing a huge metamorphosis. One major feature of this metamorphosis is the development of the wing. As can be seen from Figure 6r and s, genes related to metamorphosis, wing margin morphogenesis, wing vein morphogenesis and apposition of wing surfaces are among the most active groups of genes, and they carry their activity into adult stage. Actually, many of these genes are also very active during early embryonic stage (for example, Fig. 6b and c); though the difference is that they interact with different groups of genes. On one hand, the abundance of transcripts from these genes at embryonic stage is likely due to maternal deposit (Arbeitman *et al.*, 2002); on the other hand, this can also be due to the diverse functionalities of these genes. For instance, two genes related to wing development, held out wings (how) and tolloid (td), also play roles in embryonic development.

4 CONCLUSION

Numerous algorithms have been developed for inferring biological networks from high-throughput experimental data, such as microarray profiles (Ong, 2002; Segal *et al.*, 2003), ChIP-chip genome localization data (Bar-Joseph *et al.*, 2003; Harbison *et al.*, 2004; Lee *et al.*, 2002) and PPI data (Causier, 2004; Giot *et al.*, 2003; Kelley *et al.*, 2004; Uetz *et al.*, 2000), based on formalisms such as graph mining (Tanay *et al.*, 2004), Bayesian networks (Cowell *et al.*, 1999) and DBN (Friedman *et al.*, 2000; Kanazawa *et al.*, 1995). However, most of this vast literature focused on modeling static network or time-invariant networks, and much less has been done towards modeling the dynamic processes underlying networks that are topologically rewiring and semantically evolving over time. The method presented in this article represents a successful and practical tool for genome-wide reverse engineering dynamic interactions between genes based on their expression data.

Given the rapid expansion of categorization and characterization of biological samples and improved data collection technologies, we expect collections of complex, high-dimensional and feature-rich data from complex dynamic biological processes, such as cancer progression, immune response and developmental processes, to continue to grow. Thus, we believe our new method, KELLER, is a timely contribution that can narrow the gap between



morphogenesis of a branching structure
appendage development
cell death
germ–band shortening
multicellular organismal development
central nervous system development
morphogenesis of an epithelium
hemopoietic or lymphoid organ development
embryonic morphogenesis
reproductive structure development
embryonic pattern specification
ommatidial rotation
tube development
cell differentiation
ventral cord development
others
mushroom body development
developmental growth
appendage morphogenesis
pigmentation during development
establishment of tissue polarity
apposition of wing surfaces
cuticle development
muscle development
wing vein morphogenesis
muscle attachment
stem cell maintenance
wing margin morphogenesis
reproductive developmental process
regionalization
cell part morphogenesis
hindgut morphogenesis
foregut morphogenesis
imaginal disc development
anatomical structure formation
gut development
cell morphogenesis
head involution
gonad development
cellularization
imaginal disc fusion, thorax closure
death
organ morphogenesis
tissue development
heart development
gland development
open tracheal system development
brain development
sensory organ development
peripheral nervous system development
nervous system development
axis specification
gastrulation
metamorphosis
gland morphogenesis
tube morphogenesis
embryonic development
post–embryonic development

(**a**) Avgerage network. Each color patch denotes an ontological group, and the position of these ontological groups remain the same from (a) to (u). The annotation in the outer rim indicates the function of each group.

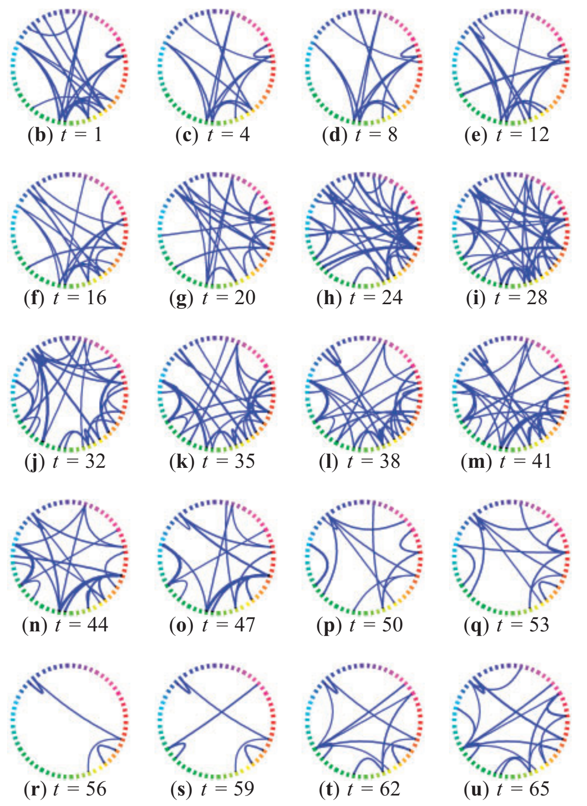


Fig. 6. Interactions between gene ontological groups related to developmental process undergo dynamic rewiring. The weight of an edge between two ontological groups is the total number of connection between genes in the two groups. In the visualization, the width of an edge is proportional to its edge weight. We thresholded the edge weight at 30 in (**b**)–(**u**) so that only those interactions exceeding this number are displayed. The average network in (**a**) is produced by averaging the networks underlying (**b**)–(**u**). In this case, the threshold is set to 20 instead.

imminent methodological needs and the available data and offer deeper understanding of the mechanisms and processes underlying biological networks.

Acknowledgements

We thank Thomas Gulish for proofreading the manuscript.

Funding: National Science Foundation CAREER Award (grant DBI-0546594 and NSF IIS-0713379 to E.P.X.); Alfred P. Sloan Research Fellowship (to E.P.X.); Ray and Stephenie Lane Research Fellowship (to L.S.).

Conflict of Interest: none declared.

REFERENCES

Ahmed,A. *et al*. (2008) Time-varying networks: Recovering temporally rewiring genetic networks during the life cycle of drosophila melanogaster. arXiv:0901.0138.

Arbeitman,M. *et al.* (2002) Gene expression during the life cycle of *Drosophila melanogaster*. *Science*, **297**, 2270–2275.

Banerjee,O. *et al.* (2008) Model selection through sparse maximum likelihood estimation. *J. Mach. Learn. Res.*, **9**, 485–516.

Bar-Joseph,Z. *et al*. (2003) Computational discovery of gene module and regulatory networks. *Nat. Biotechnol.*, **21**, 1337–1342.

Barabasi,A.L. and Albert,R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.

Barabasi,A.L. and Oltvai,Z.N. (2004) Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101–113.

Basso,K. *et al.* (2005) Reverse engineering of regulatory networks in human b cells. *Nat. Genet.*, **4**, 382–390.

Bertsekas,D. (1999) *Nonlinear Programming*. Athena Scientific, Nashua, New Hampshire, USA.

Causier,B. (2004) Studying the interactome with the yeast two-hybrid system and mass spectrometry. *Mass. Spectrom. Rev.*, **23**, 350–367.

Cowell,R.G. *et al.* (1999) *Probabilistic Networks and Expert Systems*. Springer, New York, USA.

Davidson,E.H. (2001) *Genomic Regulatory Systems*. Academic Press, New York, USA.

Duchi,J. *et al.* (2008) Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning*. Helsinki, Finland, pp. 272–279

Ernst,J. *et al.* (2007) Reconstructing dynamic regulatory maps. *Mol. Syst. Biol.*, **3**.

Friedman,N. *et al.* (2000) Using bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Fu,W. *et al.* (2008) A State-Space Mixed Membership Blockmodel for Dynamic Network Tomography. arXiv:0901.0135.

Giot,L. *et al.* (2003) A protein interaction map of drosophila melanogaster. *Science*, **302**, 1727–1736.

Harbison,C.T. *et al.* (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99–104.

Kanazawa,K. *et al.* (1995) Stochastic simulation algorithms for dynamic probabilistic networks. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence*. San Francisco, USA, pp. 346–350.

Kelley,B.P. *et al*. (2004) Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Res.*, **23**, 83–88.

Koh,K. *et al*. (2007) An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, **8**, 1519–1555.

Lee,T.I. *et al*. (2002) Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, **298**, 799–804.

Lin,C. *et al*. (2008) Trust region newton method for large-scale logistic regression. *J. Mach. Learn. Res.*, **9**, 627–650.

Luscombe,N. *et al*. (2004) Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, **431**, 308–312.

Nawab,S. and Quatieri,T. (1987) *Short-time Fourier Transform*. Prentice-Hall, Upper Saddle River, New Jersey, USA.

Ong,I.M. (2002) Modelling regulatory pathways in E.coli from time series expression profiles. *Bioinformatics*, **18**, 241–248.

Segal,E. *et al*. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, **34**, 166–176.

Tanay,A. *et al*. (2004) Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl Acad. Sci.*, **101**, 2981–2986.

Uetz,P. *et al*. (2000) A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, **403**, 601–603.

Vászquez,A. *et al*. (2004) The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *Proc. Natl Acad. Sci.*, **101**, 17940–17945.

Wainwright,M. *et al*. (2006) High dimensional graphical model selection using l1-regularized logistic regression. In *Neural Information Processing Systems 19*. MIT Press, Cambridge, Massachusetts, USA, pp. 1465–1472.

Watts,D. and Strogatz,S. (1998) Collective dynamics of ‘small-world’ networks. *Nature*, **393**, 440–442.

Zhou,S. *et al*. (2008) Time varying undirected graphs. In *Proceeding of Annual Conference on Learning Theory*. Helsinki, Finland, pp. 455–466