Rowan University

Rowan Digital Works

Theses and Dissertations

9-30-2014

Towards a dynamic view of genetic networks: A Kalman filtering framework for recovering temporally-rewiring stable networks from undersampled data

Jehandad Khan

Follow this and additional works at: https://rdw.rowan.edu/etd

Part of the Electrical and Computer Engineering Commons

Recommended Citation

Khan, Jehandad, "Towards a dynamic view of genetic networks: A Kalman filtering framework for recovering temporally-rewiring stable networks from undersampled data" (2014). *Theses and Dissertations*. 335.
https://rdw.rowan.edu/etd/335

This Thesis is brought to you for free and open access by Rowan Digital Works. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Rowan Digital Works. For more information, please contact graduateresearch@rowan.edu.

Towards a Dynamic View of Genetic Networks: A Kalman Filtering Framework for Recovering Temporally-Rewiring Stable Networks from Undersampled Data

by
Jehandad Khan

A Thesis

Submitted to the
Department of Electrical & Computer Engineering
College of Engineering
In partial fulfillment of the requirement
For the degree of
Master of Science in Engineering
at
Rowan University
Jun 24, 2014

Thesis Chair: Dr. Nidhal Bouaynaya

© 2014 Jehandad Khan

Dedication

To my mother and father

Acknowledgements

I would like to acknowledge the kind help of my advisor Dr. Nidhal Bouaynaya, Dr. Robi Polikar and Dr. Ghulaam Rasool.

Abstract

Jehandad Khan

INFERENCE AND ANALYSIS OF TIME VARYING NETWORKS
2014/06
Nidhal Bouaynaya, Ph.D.
Master of Science in Electrical & Computer Engineering

It is widely accepted that cellular requirements and environmental conditions dictate the architecture of genetic regulatory networks. Nonetheless, the status quo in regulatory network modeling and analysis assumes an invariant network topology over time. We refocus on a dynamic perspective of genetic networks, one that can uncover substantial topological changes in network structure during biological processes such as developmental growth and cancer progression. We propose a novel outlook on the inference of time-varying genetic networks, from a limited number of noisy observations, by formulating the networks estimation as a target tracking problem. Assuming linear dynamics, we formulate a constrained Kalman filtering framework, which recursively computes the minimum mean-square, sparse and stable estimate of the network connectivity at each time point. The sparsity constraint is enforced using the weighted l1-norm; and the stability constraint is incorporated using the Lyapounov stability condition. The proposed constrained Kalman filter is formulated to preserve the convex nature of the problem. The algorithm is applied to estimate the time-varying networks during the life cycle of the Drosophila Melanogaster (fruit fly).

Table of Contents

List of Figures

Figure Page

List of Tables

Table Page

Chapter 1

Research Objectives and Contributions

1.1 Research Objectives

The objectives of this research are:

- To estimate time-varying interactions from gene expression data sampled at different time intervals.
- To derive an inference method that has significant statistical power and high temporal resolution.
- To derive a method that is computationally scalable to large genetic networks.
- To exploit parallelism and leverage the availability of large-scale computational resources, such as a High-Performance Computer (HPC), in the implementation of the proposed methodology.

1.2 Research Contributions

The research contributions of this thesis are:

- Formulate the dynamic network inference problem as a time-varying tracking problem, where the moving target is the set of evolving gene interactions.

- Increase the statistical power and temporal resolution by deriving a constrained sequential tracker with forward and backward steps.

- Address the unavailability of multiple snapshots at a given time by incorporating a sparsity constraint on the network connectivity.

- Include a stability constraint to ensure the estimated network does not lead to unstable dynamics.

- Implement the proposed approach using High Performance Computing techniques, and exploiting parallelism both in the algorithm and in the available hardware.

- Scale the High-Performance implementation to efficiently process genetic networks of the order of thousands of genes.

1.3 Research Methods and Techniques

- **Optimal Bayesian Estimation** in Linear State-Space Models: the Kalman Filter.

- **Compressive Sensing** and the l_1-norm convex approximation.

- **Lyapounov Stability** as a Semi-Definite Programming (SDP) Problem.

- **Open MPI (Message Passing Interface)** to implement the algorithm on an HPC platform that scales to large networks.

Chapter 2

Introduction

2.1 Motivation

A major challenge in systems biology today is to understand the behaviors of living cells from the dynamics of complex genomic regulatory networks. It is no more possible to understand the cellular function from an informational point of view without unraveling the underlying regulatory networks than to understand protein binding without knowing the protein synthesis process. The advances in experimental technology have sparked the development of genomic network inference methods, also called *reverse-engineering* of genomic networks. Most popular methods include (probabilistic) Boolean networks [31,49], (dynamic) Bayesian networks [20,21,38] information-theoretic approaches [9,36,60,61] and differential equation models [8,16,44]. A comparative study is compiled in [27]. The DREAM (Dialogue on Reverse Engineering Assessment and Methods) project, which built a blind framework for performance assessment of methods for gene network inference, showed that there is no single inference method that performs optimally across all data sets. In contrast, integration of predictions from multiple inference methods shows robust and high performance across diverse data sets [35].

These methods, however, estimate one single network from the available data, independently of the cellular "themes" or environmental conditions under which the measurements were collected. In signal processing, it is senseless to find the Fourier spectrum of a non-stationary time series [29]. Similarly, time-dependent genetic data from dynamic biological processes such as cancer progression, therapeutic responses and developmental processes, cannot be used to describe a unique time-invariant or static network [11], [33]. Inter and intracellular spatial cues affect the course of events in these processes by rewiring the connectivity between the molecules to respond to specific cellular requirements, e.g., going through the successive morphological stages during development. Inferring a unique static network from a time-dependent dynamic biological process results in an "average" network that cannot reveal the regime-specific and key transient interactions that cause cell biological changes to occur. For a long time, it has been clear that the evolution of the cell function occurs by change in the genomic program of the cell, and it is now clear that we need to consider this in terms of change in regulatory networks [11], [33].

2.2 Related Work

While there is a rich literature on modeling static or time-invariant networks, much less has been done towards inference and learning techniques for recovering topologically rewiring networks. In 2004, Luscombe *et al.* made the earliest attempt to unravel topological changes in genetic networks during a temporal cellular process, or in response to diverse stimuli [33]. They showed that, under different cellular conditions, transcription factors, in a genomic regulatory network of *Saccharomyces* *cerevisiae*, alter their interactions to varying degrees, thereby rewiring the network. Their method, however, is still based on a static representation of known regulatory interactions. To get a dynamic perspective, they integrated gene expression data for five conditions: cell cycle, sporulation, diauxic shift, DAN damage and stress response. From these data, they traced paths in the regulatory network that are active in each condition using a trace-back algorithm [33].

The main challenge facing the community in the inference of time-varying genomic networks is the unavailability of multiple measurements of the networks or multiple observations at every instant t. Usually, one or at most a few observations are available at each instant. This leads to the "large p small n" problem, where the number of unknowns is larger than the number of available observations. The problem may seem ill-defined because no unique solution exists. However, we will show that this hurdle can be circumvented by using prior information.

One way to ameliorate this data scarcity problem is to presegment the time-series into stationary epochs, and infer a static network for each epoch separately [15, 17, 25, 43, 43, 47, 57]. The segmentation of the time-series into stationary pieces can be achieved using several methods including estimation of the posterior distribution of the change points [17], HMMs [15], clustering [43], detecting geometric structures transformed from time series [57], MCMC sampling algorithm to learn the times of non-stationarities (transition times) [25], [47]. The main problem with the segmentation approach for estimating time-varying gene networks is the limited number of time points available in each stationary segment, which is a subset of the already limited data. Since the time-invariant networks are inferred in each segment using only the data points within that segment and disregarding the rest of the data, the resulting networks are limited in terms of their temporal resolution and statistical power.

A semi-flexible model based on a piecewise homogeneous dynamic Bayesian network, where the network structure in each segment shares information with adjacent segments, was proposed in [12]. This setting allows the network to vary gradually through segments. However, some information is lost by not considering the entire data samples for the piecewise inference. A more flexible model of time-varying Bayesian networks based on a non parametric Bayesian method for regression was recently proposed in [37]. The nonparametric regression is expected to enable capturing nonlinear dynamics among genes [12]. However, a full-scale study of a time-varying system was lacking; the approach was only tested on an 11-gene Drosophila melanogaster network.

Full resolution techniques, which allow a time-specific network topology to be inferred from samples measured over the entire time series, rely on model-based approaches [2], [26]. However, these methods learn the structure (or skeleton) of the network but not the detailed strength of the interactions between the nodes. Dynamic Bayesian networks (DBNs) have been extended to the time varying case [32], [42], [46], [51]. Among the earliest models is the time varying autoregressive (TVAR) model [42], which describes nonstationary linear dynamic systems with continuously changing linear coefficients. The regression parameters are estimated recursively using a normalized least-squares algorithm. In time-varying DBNs (TVDBN), the time-varying structure and parameters of the networks are treated as additional hidden nodes in the graph model [32].

In summary, the current state-of-the-art in time-varying network inference relies on either chopping the time-series sequence into homogeneous subsequences [15, 17, 19, 25, 39, 41, 43, 47, 54, 57] (concatenation of static networks) or extending graphical models to the time-varying case [32, 42, 46, 51] (time modulation of static networks).

Chapter 3

The State-Space Model

3.1 The State-Space Model

Static gene networks have been modeled using a standard state-space representation, where the state $\boldsymbol{x}_k$ represents the gene expression values at a particular time k and the microarray data $\boldsymbol{y}_k$ constitutes the set of noisy observations [55], [40]. A naive approach to tackle the time-varying inference problem is to generalize this representation of time-invariant networks, and augment the gene profiles state vector by the network parameters at all time instants. This approach, however, will result in a very poor estimate due to the large number of unknown parameters. Instead, we propose to re-formulate the state-space model as a function of the time-varying connections or parameters rather than the gene expression values. In order to do so, we need to model the time evolution of the parameters using, for instance, prior knowledge about the biological process. Denoting by $\boldsymbol{a}_k$ the network parameters to be estimated, the state-space model of the time-varying network parameters can be written as

$$\boldsymbol{a}(k+1) = f_k(\boldsymbol{a}(k)) + \boldsymbol{w}(k), \tag{3.1}$$

$$\boldsymbol{y}(k) = g_k(\boldsymbol{a}(k)) + \boldsymbol{v}(k). \tag{3.2}$$

Where, the function f_k models the dynamical evolution of the network parameters, e.g., smooth evolution or abrupt changes across time. The observation function g_k characterizes the regulatory relationships among the genes, and can be, for instance, derived from a differential equation model of gene expression (see Eq. (3.7)). In particular, observe that the state-space model in (3.1)-(3.2) does not incorporate the "true" gene expression values, which have to be estimated and subsequently discarded. It only includes the measured gene expression values with an appropriate measurement noise term.

3.2 The Observation Model

We model the concentrations of mRNAs, proteins, and other molecules using a time-varying ordinary differential equation (ODE). More specifically, the concentration of each molecule is modeled as a linear function of the concentrations of the other components in the system. The time-dependent coefficients of the linear ODE capture the rewiring structure of the network. We have

$$\dot{x}_i(t) = -\lambda_i(t)x_i(t) + \sum_{j=1}^{p} w_{ij}(t)x_j(t) + b_i + v_i(t), \tag{3.3}$$

where $i = 1, \cdots, p$, p being the number of genes, $x_i(t)$ is the expression level of gene i at time t, $\dot{x}_i(t)$ is the rate of change of expression of gene i at time t, λ_i is the self degradation rate, $w_{ij}(t)$ represents the time-varying influence of gene j on gene i, b_i is the base production rate and $v_i(t)$ models the measurement and biological noise. The goal is to infer the time-varying gene interactions $\lambda_i(t), \{w_{ij}(t)\}_{i,j=1}^{p}$, given a limited number of measurements $n < p$.

To simplify the notation, we absorb the self degradation rate $\lambda_i(t)$ into the interaction parameters by letting $a_{ij}(t) = w_{ij}(t) - \lambda_i(t)\delta_{ij}$, where δ_{ij} is the Kronecker delta function. The external perturbation is assumed to be known. The discrete-time equivalent of (3.3) can, therefore, be expressed as

$$\dot{x}_i(k) = \sum_{j=1}^{p} a_{ij}(k)x_j(k) + b_i + v_i(k),\ i = 1, \cdots, p,\ k = 1, \ldots, n. \tag{3.4}$$

Writing (3.4) in matrix form, we obtain

$$\mathbf{y}(k) = A(k)\ \mathbf{x}(k) + \mathbf{b} + \mathbf{v}(k), \tag{3.5}$$

where $\mathbf{y}(k) = [y_1(k), \ldots, y_p(k)]^T$, $A(k) = \{a_{ij}(k)\}$ is the matrix of time-dependent interactions, $\mathbf{x}(k) = [x_1(k), \ldots x_p(k)]^T$,$\mathbf{b} = [b_1, \ldots, b_p]^T$ is the base production rate and $\mathbf{v}(k) = [v_1(k), \ldots, v_p(k)]^T$.

Let $1 \leq m_k < p$ be the number of available observations at time k. Taking into account all m_k observations, Eq. (3.5) becomes

$$\mathbf{Y}(k) = \boldsymbol{A}(k)\ \mathbf{X}(k) + \mathbf{B} + \mathbf{V}(k), \tag{3.6}$$

where $\boldsymbol{Y}(k), \boldsymbol{X}(k)$ and $\boldsymbol{V}(k) \in \mathbb{R}^{p \times m_k}$ with the m_k observations ordered in the columns of the corresponding matrices, the matrix $\mathbf{B} = \mathbf{b}\ \mathbf{1}^T$ represents the same response of a particular gene in all the measurements.

The linear model in Eq. (3.6) can be decomposed into p independent linear models as follows:

$$\boldsymbol{y}_i^t(k) = \boldsymbol{a}_i^t(k)\boldsymbol{X}(k) + \boldsymbol{b}_i\mathbf{1}^T + \boldsymbol{v}_i^t(k), \tag{3.7}$$

where $\boldsymbol{y}_i^t(k), \boldsymbol{a}_i^t(k)$, $\boldsymbol{b}_i\mathbf{1}^T$ and $\boldsymbol{v}_i^t(k)$ are the i^{th} rows of $\boldsymbol{Y}(k), \boldsymbol{A}(k)$, $\boldsymbol{B}$ and $\boldsymbol{V}(k)$, respectively. In particular, the vector $\boldsymbol{a}_i(k)$ represents the set of incoming edges to gene i at time k. Equation (3.7) represents the observation equation for gene i.

3.3 The Linear State-Space Model

The state equation models the dynamics of the state vector $\boldsymbol{a}_i(k)$ given a priori knowledge of the system. In this work, we assume a random walk model of the network parameters. The random walk model is chosen for two reasons. First, it reflects a flat prior or a lack of a priori knowledge. Second, it leads to a smooth evolution of the state vector over time (if the variance of the random walk is not very high). The state space model of the incoming edges for gene i is, therefore, given by

$$\begin{cases} \boldsymbol{a}_i(k+1) = \boldsymbol{a}_i(k) + \boldsymbol{w}_i(k) \\ \\ \boldsymbol{y}_i(k) = \boldsymbol{X}^t(k)\boldsymbol{a}_i(k) + \boldsymbol{b}_i\mathbf{1} + \boldsymbol{v}_i(k), \end{cases} \tag{3.8}$$

where $i = 1, \cdots, p$, $\boldsymbol{w}_i(k)$ and $\boldsymbol{v}_i(k)$ are, respectively, the process noise and the observation noise, assumed to be zero mean Gaussian noise processes with known covariance matrices, $Q(k)$ and $R(k)$, respectively. In addition, the process and observation noise are assumed to be uncorrelated with each other and with the state vector

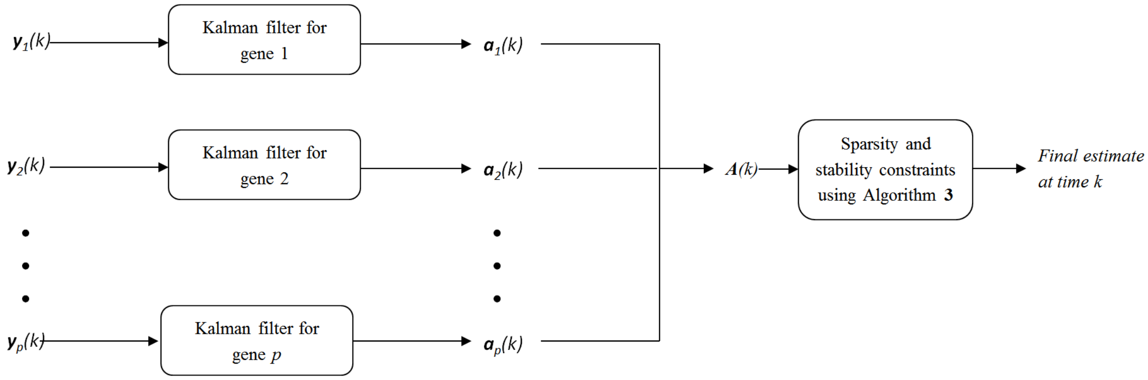


Figure 3.1: Parallel architecture of the tracker. The tracking is performed for each gene separately to find its incoming edges. The connectivity matrix $\boldsymbol{A}(k) = [\boldsymbol{a}_1^t; \cdots ; \boldsymbol{a}_p^t]$.

$\boldsymbol{a}_i(k)$. In particular, we have p independent state-space models of the form (3.8) for $i = 1, \cdots, p$. Thus, the connectivity matrix A can be recovered by simultaneous recovery of its rows. Another important advantage of the representation in (3.8) is that the state vector $\boldsymbol{a}_i(k)$ has dimension p (the number of genes in the network) rather than p^2 (the number of possible connections in the network); thus avoiding the curse of dimensionality problem. For instance, in a network of 100 genes, the state vector will have dimension 100 instead of 10,000! Though the number of genes p can be large, we show in simulations that the performance of the Kalman tracker is unchanged for p as large as 5000 genes by using efficient matrix decompositions to find the numerical inverse of matrices of size p. A graphical representation of the parallel architecture of the tracker is shown in Fig. 3.1.

It is well known that the minimum mean square estimator, which minimizes $E[\|\boldsymbol{a}(k) - \hat{\boldsymbol{a}}(k)\|_2^2]$, can be obtained using the Kalman filter if the system is observable. If the system is unobservable, then the classical Kalman filter cannot recover the optimal estimate. In particular, it seems hopeless to recover $\boldsymbol{a}_i(k) \in \mathbb{R}^p$ in (3.8)

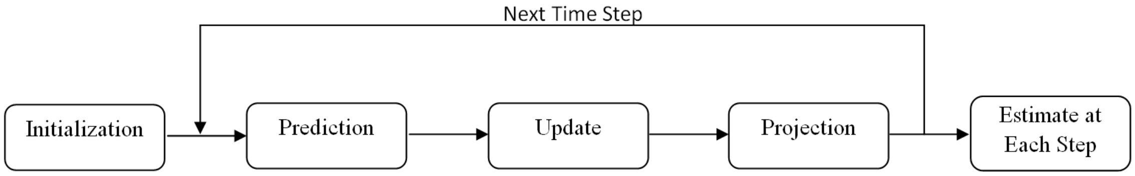


Figure 3.2: The Constrained Kalman filter: the prior estimate is predicted to give $\boldsymbol{a}_{k|k-1}$. The filter is updated with the observations to give the unconstrained estimate $\boldsymbol{a}_{k|k}$. The projection operator projects this estimate to enforce the constraint. This procedure is repeated for all time steps $k = 1, \cdots, n$.

from an under-determined system where $m_k < p$. Fortunately, this problem can be circumvented by taking into account the fact that $\boldsymbol{a}_i(k)$ is sparse. Genomic regulatory networks are known to be sparse, that is each gene is governed by only a small number of the genes in the network [44]. We give further details in the following chapter.

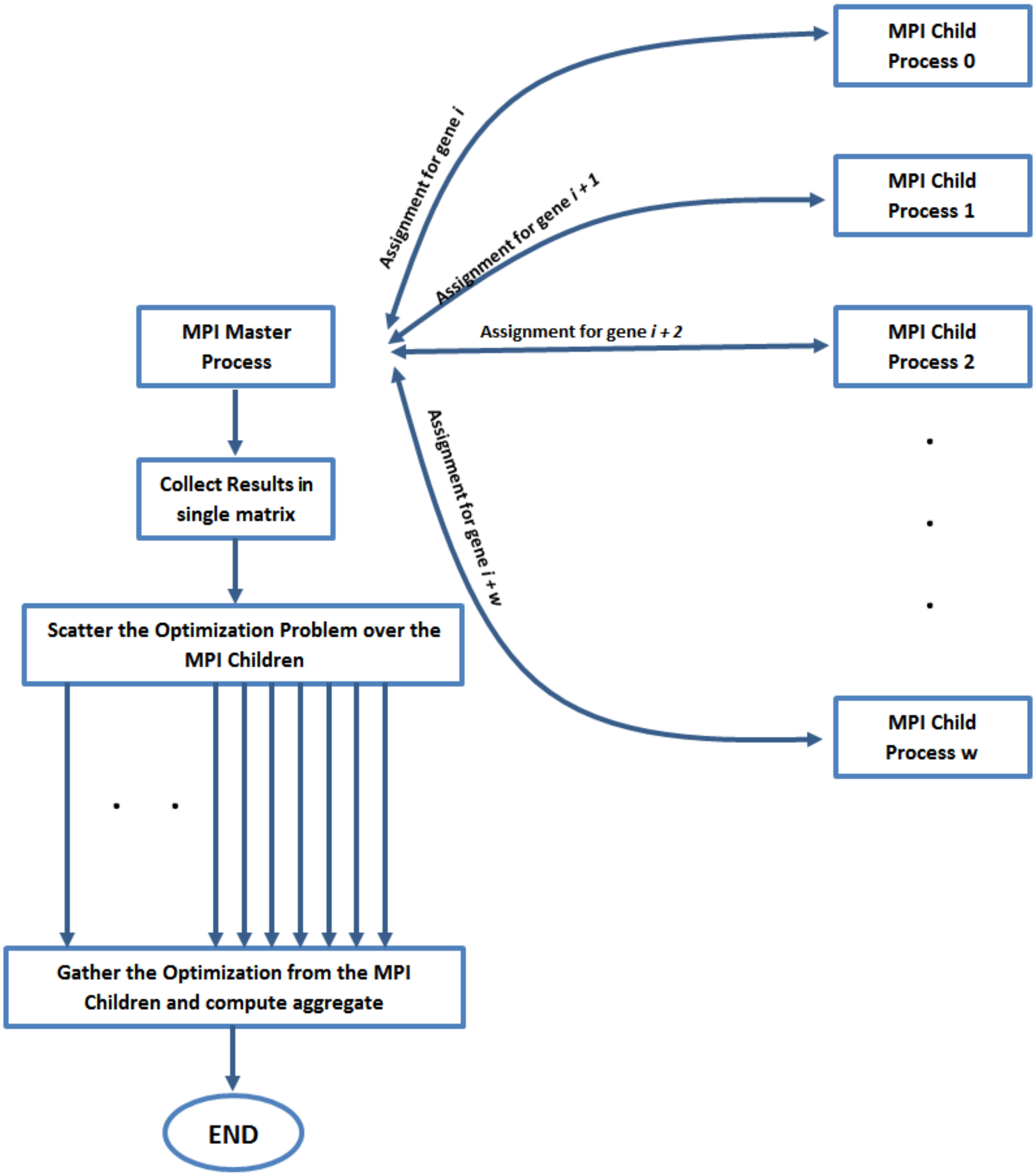

MPI Child Process 0
MPI Child Process 1
MPI Child Process 2
MPI Child Process w
Assignment for gene i
Assignment for gene i + 1
Assignment for gene i + 2
Assignment for gene i + w
MPI Master Process
Collect Results in single matrix
Scatter the Optimization Problem over the MPI Children
Gather the Optimization from the MPI Children and compute aggregate
END

Figure 3.3: Parallel architecture of the tracker mapped to the MPI implementation. Image depicting the role of MPI children, both in tracking and optimization.

Chapter 4

The Compressive-Kalman Filter

4.1 Sparse Signal Recovery

Recent studies [7], [18] have shown that sparse signals can be exactly recovered from an under-determined system of linear equations by solving the optimization problem

$$\min \|\hat{\boldsymbol{z}}\|_0 \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{z}}\|_2^2 \leq \epsilon, \tag{4.1}$$

for a sufficiently small ϵ and where the l_0-norm, $\|\boldsymbol{z}\|_0$, denotes the support of $\boldsymbol{z}$ or the number of non-zero elements in $\boldsymbol{z}$. The optimization problem in (4.1) can be extended to the stochastic case as follows

$$\min \|\hat{\boldsymbol{z}}\|_0 \text{ s.t. } E_{\boldsymbol{z}|\boldsymbol{y}}[\|\boldsymbol{z} - \hat{\boldsymbol{z}}\|_2^2] \leq \epsilon. \tag{4.2}$$

Unfortunately, the above optimization problem is, in general, NP hard. However, it has been shown that if the observation matrix $\boldsymbol{H}$ obeys the restricted isometry property (RIP), then the solution of the combinatorial problem (4.1) can be recovered by solving instead the convex optimization problem

$$\min \|\hat{\boldsymbol{z}}\|_1 \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{z}}\|_2^2 \leq \epsilon. \tag{4.3}$$

This is a fundamental result in the emerging theory of *compressed sensing*(CS) [7], [18]. CS reconstructs large dimensional signals from a small number of measurements, as long as the original signal is sparse or admits a sparse representation in a certain basis. Compressed sensing has been implemented in many applications including digital tomography [7], wireless communication [52], image processing [4] and camera design [14]. For a further review of CS, the reader can refer to [7], [18].

Inspired by the compressed sensing approach given that genomic regulatory networks are sparse, we formulate a constrained Kalman objective

$$\min_{\hat{\boldsymbol{z}}} E_{\boldsymbol{z}|\boldsymbol{y}} \left[\|\boldsymbol{z} - \hat{\boldsymbol{z}}\|_2^2\right] \text{ s.t. } \|\hat{\boldsymbol{z}}\|_1 \leq \epsilon. \tag{4.4}$$

The constrained Kalman objective in (4.4) can be seen as the regularized version of least squares known as least absolute shrinkage and selection operator (LASSO) [53], which uses the l_1 constraint to prefer solutions with fewer non-zero parameter values, effectively reducing the number of variables upon which the given solution is dependent. For this reason, the LASSO and its variants are fundamental to the theory of compressed sensing. In this work, we have used the *weighted* l_1 norm as the sparsity inducing operator. Compared to the l_1 norm, the weighted l_1 norm eliminates any weak genetic interactions in the estimated matrix. Furthermore, recent theoretical results [59] show that, in some cases, minimizing the weighted l_1 norm of a matrix A minimizes the cardinality or l_0 norm of A with high probability.

4.2 Stability Criterion

Incorporating stability in the estimation of biological networks is crucial in order to obtain meaningful estimates; otherwise, the estimated networks may be unstable leading to diverging observations. Let us consider the nonlinear time-invariant system given by :

$$\dot{x} = f(x) \tag{4.5}$$

where $f : \mathbf{R}^n \to \mathbf{R}^n$. Let $x_e \in \mathbf{R}^n$ be the *equilibrium point* of the above system i.e., $f(x_e) = 0$. We define two types of stability.

Global Asymptotic Stability The system (4.5) is *Globally Asymptotically Stable* (G.A.S.) if for every trajectory $x(t)$, we have $x(t) \to x_e$ as $t \to \infty$. In other words, regardless of the starting point of the system, given enough time it would always reach its equilibrium point.

Local Asymptotic Stability The system (4.5) is *Locally Asymptotically Stable* near or at x_e if there exists an $R > 0$ s.t. $\|x(0) - x_e\| \leq R \implies x(t) \to x_e$ as $t \to \infty$. More simply, if the state trajectory of a system is inside a ball of radius R centered at x_e, then as time approaches infinity, the system will attain the equilibrium state x_e.

A linear system with dynamics given by $\dot{x} = Ax$ is both G.A.S. and L.A.S. if the real part of all the eigenvalues of A is negative i.e., $\Re\{\lambda_i(A)\} < 0$.

4.2.1 Lyapunov Stability Theorem.

The Lyapunov Stability Theorem [5] establishes a sufficient condition for global asymptotic stability, as follows

If there exists a function $V : \mathbf{R}^n \rightarrow \mathbf{R}$ such that

$$\begin{aligned} V(x(t)) &\geq 0, \forall x(t) && (4.6) \\ V(x(t)) &= 0, \text{ if and only if } x(t) = 0 && (4.7) \\ V(x(t)) &\rightarrow \infty \text{ as } x(t) \rightarrow \infty && (4.8) \\ \dot{V}(x(t)) &< 0 \ \forall x(t) \neq 0, \dot{V}(0) = 0 && (4.9) \end{aligned}$$

then, every trajectory of $\dot{x} = f(x)$ converges to zero as $t \rightarrow \infty$. Thus the system is *globally asymptotically stable*, the function V is known as the Lyapunov function and the first three conditions state that the function is positive definite.

In light of the above theorem, if we are given the following linear time-varying system

$$\dot{x}(t) = A(t)x(t) \tag{4.10}$$

with $A(t) \in \{A_1, \cdots, A_K\}$. Then it is globally asymptotically stable for the Lyapunov function $V(z) = z^T P z$ if

$$A_i^T P + P A_i \leq 0, \quad i = 1, \cdots, K \tag{4.11}$$

Conversely if the system (4.10) is stable, then there exists a symmetric positive definite matrix P that satisfies (4.11). In order to constrain a system to be stable we need to find the corresponding Lyapounov matrix. We will show that finding the Lyapounov matrix P can be casted as a semidefinite program (SDP) [6].

Let $\hat{A}_{KF}$ be the matrix estimated by the Kalman filter, which, in general, is not stable. The aim is to perturb A by a "small" perturbation D so that $\hat{A}_{KF} + D$ is stable and sparse. Let $\tilde{A} = \hat{A}_{KF} + D$. As we discussed earlier, the necessary and sufficient condition for the stability of $\tilde{A}$ is the existence of a symmetric, positive definite Lyapounov matrix P such that

$$\tilde{A}^T P + P\tilde{A} < 0. \tag{4.12}$$

Following the work in [59], we let $L = PD$. Equation (4.12) then becomes

$$\hat{A}_{KF}^T P + L^T + P\hat{A}_{KF} + L < 0, \tag{4.13}$$

which is a linear matrix inequality in both P and L. In order to solve for D, or equivalently P and L, we minimize the error between the data resulting from the unstable estimate $\hat{A}_{KF}$ and the stable estimate $\tilde{A}$, i.e., we consider the objective

$$||(\tilde{A}X + BU) - (\hat{A}_{KF}X + BU)||_2 = ||P^{-1}LX||_2 \leq \frac{||LX||_2}{||P||_2} \leq ||LX||_2, \tag{4.14}$$

where $||P|| \geq 1$. The SDP optimization problem to solve for P and L is then given by

$$\begin{aligned} \min_{L} \quad & \|LX\|_2 \\ \text{subject to} \quad & \hat{A}_{KF}^T P + P\hat{A}_{KF} + L^T + L \leq 0 \\ & P \geq I \end{aligned} \tag{4.15}$$

Let P^* and L^* be the unique solution of Eq. (4.15). The stable matrix is then given by $\tilde{A} = \hat{A}_{KF} + P^{*-1}L^*$. However, $\tilde{A}$ may no longer be sparse. Therefore, we need to further perturb $\tilde{A}$ in order to impose the sparsity constraint as well as any other desired constraints. We formulate the optimization problem to find the stable and sparse matrix A as follows

$$\begin{aligned} \min_{A,B,\epsilon,\eta} \quad & \alpha \sum_{i,j=1}^{N} w_{ij}|a_{ij}| + \beta\ \epsilon + \gamma\ \eta \\ \text{subject to} \quad & A^T P^* + P^* A \leq 0 \\ & \|Y - (AX + B)\|_2^2 \leq \epsilon \\ & \|A - \hat{A}_{KF}\|_2^2 \leq \eta \end{aligned} \tag{4.16}$$

Where α, β and γ are fixed weighting parameters satisfying $\alpha + \beta + \gamma = 1$. The first term in the objective function, weighted by α, ensures that the connectivity matrix A is sparse. The second term, weighted by β, ensures that the stable matrix also minimizes the error between the model and the observations. The third term, weighted by γ, ensures that the stable matrix is within the vicinity of the (unstable) Kalman estimate. The matrix $W = \{w_{ij}\}$ is the weighting matrix. In each pass of the above optimization algorithm the weights w_{ij} are updated as [59]

$$w_{ij} = \frac{\delta}{\delta + |a_{ij}|} \tag{4.17}$$

The intuition behind this heuristic weight update is that large weights are assigned to small matrix entries $|a_{ij}|$ and small weights to large entries, which eliminates any weak interactions in the final matrix. This process is repeated until the values of w_{ij} converge. In practice it takes no more than 10 iterations to converge, however this number might change with the number of genes in the system. This algorithm is shown in algorithm 1.

Algorithm 1 State Constraint

```
Require: $\{w_{ij}\} = 1$, $\alpha$, $\beta$ and $\gamma$.
Lyapunov Matrix: Solve the SDP (4.15) to determine $P^*$
for idx = 1 to N do
    Solve the SDP (4.16) to determine $A$ and $B$
    Update the weights using equation (4.17)
    if $\|W(idx) - W(idx-1)\|_2 < \theta$ then
        end for loop
    end if
end for
```

4.3 Constrained Kalman Filtering

Constrained Kalman filtering has been mainly investigated in the case of linear equality constraints of the form $\boldsymbol{Dx} = \boldsymbol{d}$, where $\boldsymbol{D}$ is a known matrix and $\boldsymbol{d}$ is a known vector [50]. The most straightforward method to handle linear equality constraints is to reduce the system model parametrization [56]. This approach, however, can only be used for linear equality constraints and cannot be used for inequality constraints (i.e., constraints of the form $\boldsymbol{Dx} \leq \boldsymbol{d}$). Another approach is to treat the state constraints as perfect measurements or pseudo-observations (i.e., no measurement noise) [28]. The perfect measurements technique applies only to equality constraints as it augments the measurement equation with the constraints. The third approach is to project the standard (unconstrained) Kalman filter estimate onto the constraint surface [50]. Though non-linear constraints can be linearized and then treated as perfect observations, linearization errors can prevent the estimate from converging to the true value. Non-linear constraints are, thus, much harder to handle than linear constraints because they embody two sources of errors: truncation errors and base point errors [23], [30]. Truncation errors arise from the lower order Taylor series approximation of the constraint, whereas base point errors are due to the fact that the filter linearizes around the estimated value of the state rather than the true value.

In this work, we adopt the projection approach, which projects the unconstrained Kalman estimate at each step onto the set of stable and sparse vectors, as defined by the optimization problem in (4.16). Hence the Kalman Filter computes a non-sparse and possibly unstable solution to the state space, which is projected to a sparse and stable space.

The reader might recall that this estimate may be computed in parallel since each gene is independently characterized by the observation equation (3.7). However the stability and sparsity criterion described in (4.16) are global properties of the time varying connectivity matrices. Thus breaking the parallel nature of the algorithm. This problem is addressed by employing a parallel sdp solver such as [58], which is capable to solve large scale Semidefinite programming problems in parallel. It might also be noted that the problem at hand is sparse in nature and thus may exploit efficient linear algebra routines for fast parallel solutions.

The Kalman filter equations for a Linear Time Varying system are divided in two steps, also known as the prediction and the update steps. These equations with a known input are given as [24]

Prediction

$$\begin{aligned} \boldsymbol{a}_{k|k-1} &= \boldsymbol{a}_{k-1|k-1} \qquad (4.18) \\ \boldsymbol{V}_{k|k-1} &= \boldsymbol{V}_{k-1|k-1} + \boldsymbol{Q}_k \qquad (4.19) \end{aligned}$$

Update

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{V}_{k|k-1}\boldsymbol{X}_k(\boldsymbol{X}_k^t\boldsymbol{V}_{k|k-1}\boldsymbol{H}_k^t + \boldsymbol{R}_k)^{-1}, \qquad (4.20) \\ \boldsymbol{a}_{k|k} &= \boldsymbol{a}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{X}_k^t\boldsymbol{a}_{k|k-1} - \boldsymbol{B}_k), \qquad (4.21) \\ \boldsymbol{V}_{k|k} &= (I - \boldsymbol{K}_k\boldsymbol{X}_k^t)\boldsymbol{V}_{k|k-1}. \qquad (4.22) \end{aligned}$$

Where $\boldsymbol{a}_{k|k-1}$ is the state estimate at time k, given observations till time $k-1$, $\boldsymbol{V}_{k|k-1}$ is the error covariance of the estimate at time k given the observations till time $k-1$, $\boldsymbol{K}_k$ is the Kalman gain at time k and rest of the variables correspond to the state space model. The complete procedure is listed in algorithm 3, which brings all the pieces together and forms the complete picture.

4.3.1 The Initial Condition.

The Kalman filter is known to converge asymptotically to the true solution regardless of the initial condition that is supplied to initiate the filter [50]. However for a constrained Kalman filter, particularly in this scenario when the total number of time points is limited, it is paramount to have an initial condition that is as close to the true state as possible. Intuitively a more informed initial condition will result in a much more accurate filtering trajectory as opposed to a random or uninformed one. Moreover the Kalman filter given in equations (4.18) and (4.20) assume known $B(k)$ matrices which must be supplied. To address this issue we employ the sparse Maximum Likelihood (sML) method adopted in [45] to compute the initial condition for the Kalman filter.

Initial Condition Step 1 - Determine an unstable sparse solution

To determine the initial condition, first we compute a sparse and possibly unstable solution given the data by recursively solving the following optimization problem until convergence of the weights:

$$\begin{aligned} \min_{A,B} \quad & t\sum_{i,j=1}^{N} w_{ij}|a_{ij}| + (1-t)\|Y-(AX+B)\|_2^2 \\ \text{subject to} \quad & A \in \quad S \end{aligned} \tag{4.23}$$

where we incorporate any known interactions in the set S in the initial condition. More specifically the set of matrices S is defined as follows:

$$A \in S \implies \begin{cases} a_{ij} \geq \delta & , if s_{ij} = +1 \\ a_{ij} \leq \delta & , if s_{ij} = -1 \\ a_{ij} \in \mathbb{R} & , \text{otherwise} \end{cases} \tag{4.24}$$

This information further improves the estimate by incorporating existing biological knowledge.

Step 2 - Determine a Lyapunov matrix for this solution

The resultant A matrix from the above problem might be unstable. To determine a corresponding stable matrix, we compute a symmetric positive semidefinite Lyapunov matrix. To compute the Lyapunov matrix Q we solve the following SDP [59]

$$\begin{aligned} \min_{Q,L} \quad & \|LX\|_2 \\ \text{subject to} \quad & Q \geq I \\ & A^T Q + QA + L^T + L \leq 0 \end{aligned} \tag{4.25}$$

Where X is the matrix of observations for the first time epoch. Let Q^* be the solution of (4.25).

Step 3 - Stabilize the solution

We use the Lyapunov solution matrix Q^* of (4.25) in order to stabilize the matrix A in (4.23). The final problem that needs to be solved becomes

$$
\begin{aligned}
\min_{A,B} \quad & t\sum_{i,j=1}^{N} w_{ij}|a_{ij}| + (1-t)\|Y-(AX+B)\|_2^2 \\
\text{subject to} \quad & A^TQ^* + Q^*A \leq 0 \\
& A \in S
\end{aligned}
\tag{4.26}
$$

Algorithm 2 Initial Condition

```
Require: $\{w_{ij}\} = 1$, known interactions $S$, gene expression data $X$ and $Y$.
for idx = 1 to N do
    Solve the SDP (4.23) to determine $A$ and $B$
    Update the weights using equation (4.17)
    if $\|W(idx) - W(idx-1)\|_2 < \theta$ then
        end for loop
    end if
end for
Lyapunov Matrix: Solve the SDP (4.25) to determine $P$
for idx = 1 to N do
    Solve the SDP (4.26) to determine $A$ and $B$
    Update the weights using equation (4.17)
    if $\|W(idx) - W(idx-1)\|_2 < \theta$ then
        end for loop
    end if
end for
```

Algorithm 3 Constrained Kalman Filter

```
Initialization Apply algorithm 2 to find the initial condition
for each time $t$ do
    Compute the Kalman Estimate using equations (4.18) and (4.20)
    Constraint: Apply algorithm 1 to constrain the estimate
end for
```

Chapter 5

Results and Discussion

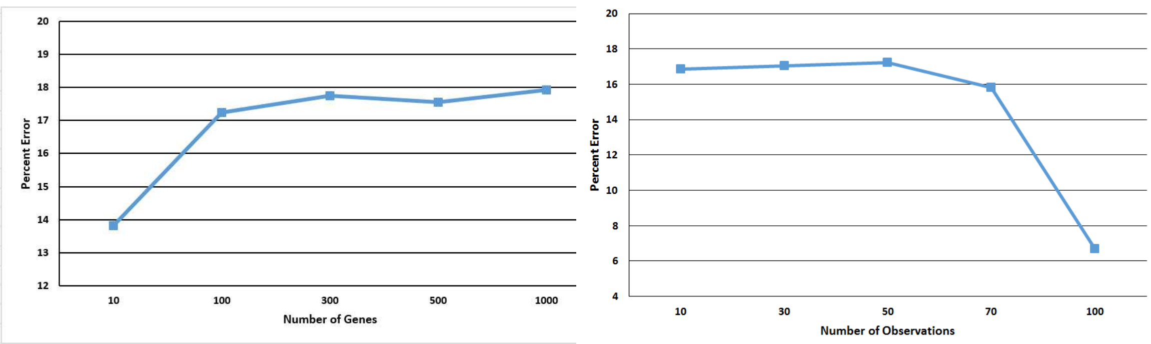
5.1 Synthetic Data

In order to assess the efficacy of the proposed compressive Kalman filter in estimating the connectivity of time-varying networks, we first perform Monte Carlo simulations on generated data to assess the prediction error using the following criterion

$$|a_{ij} - \hat{a_{ij}}| \leq \alpha |a_{ij}| \tag{5.1}$$

Where a_{ij} is the $(i, j)^{th}$ true edge value and $\hat{a_{ij}}$ is the corresponding predicted edge value. The criterion in (5.1) counts an error if the estimated edge value is outside an α-vicinity of the true edge value. In our simulations, we adopted a value of α equal to 0.2. That is, the error tolerance interval is ±20% of the true value. The percentage of total correct or incorrect edges in a connectivity matrix is used to determine the accuracy of the algorithm.

We first investigate the effect of the network size on the estimation error. We generate networks of different sizes according to the model in (3.6), and calculate the prediction error. Figure 5.1a shows the prediction error as a function of the network size with a number of measurements equal to 70% the network size p. We observe that the network estimation error is about constant between $p = 100$ to $p = 1000$, and is thus unaffected by how large the network is, at least for networks of size few thousand genes. The reason for this outcome may be the linear increase of the size vector with the number of genes, which is due to the splitting of the original connectivity estimation problem (p^2 parameters) into p smaller problems, that can be solved simultaneously.

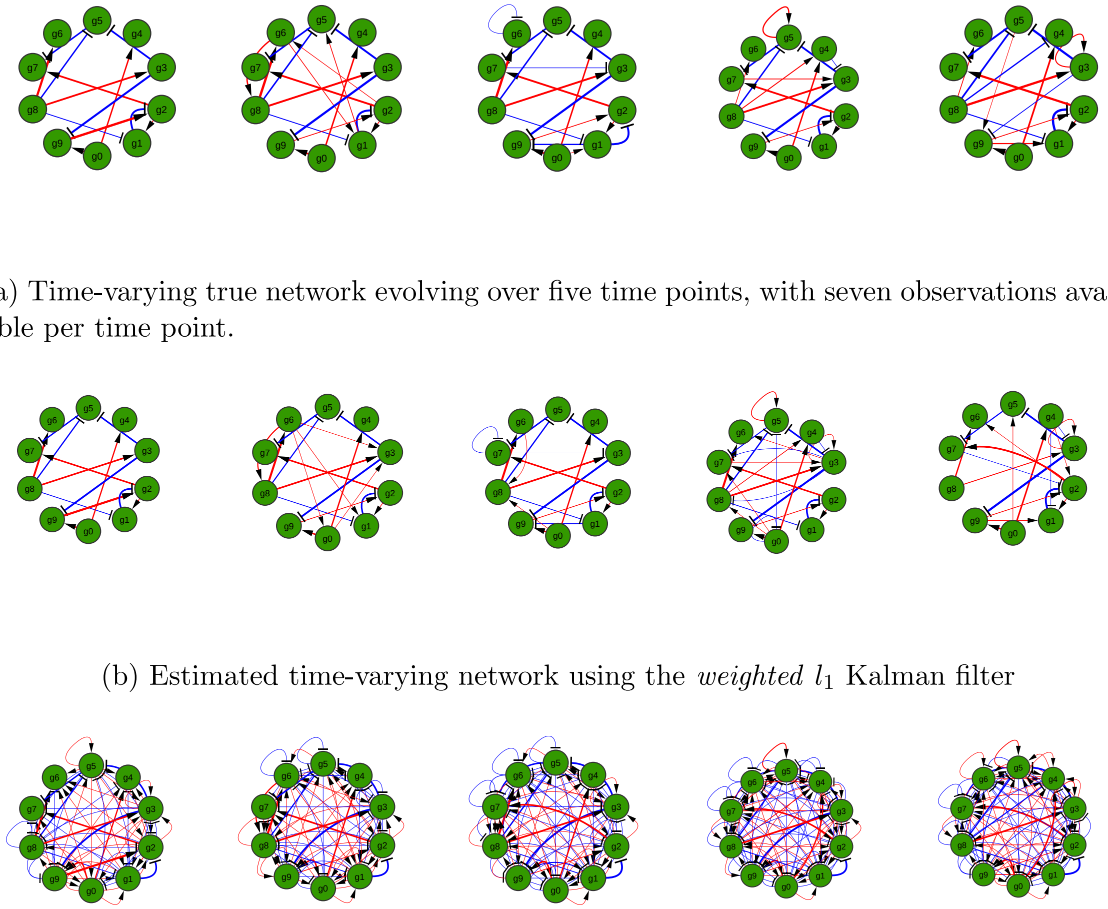


(a) Error v.s. number of genes

(b) Error v.s. number of observations

Figure 5.1: (a) Effect of the network size on the prediction error; (b) Effect of the number of observations on the prediction error

We subsequently investigated the effect of the number of measurements m on the prediction accuracy. Fig 5.1b shows the prediction error as a function of the number of observations for a network of size $p = 100$. The estimation error seems to be constant up to 50 measurements then decreases rapidly as the number of observations increase to 100. But even for a small number of observations (10% of the network size), the estimation error is fairly small (less than 18 %). This is an important result because in real-world applications the number of available observations is very limited. We believe that the reason the error stays about constant for a small number



(a) Time-varying true network evolving over five time points, with seven observations available per time point.

(b) Estimated time-varying network using the *weighted* l_1 Kalman filter

(c) Estimated time-varying network using the classical Kalman filter.

Figure 5.2: Tracking of a 10-gene network evolving over five time points, with seven observations or measurements available at each time point.

of measurements (up to 50) is due to the good initial condition that is adopted in these simulations. For randomly chosen initial conditions, the *weighted* l_1 Kalman filter takes a longer time, and thus requires more observations, to converge.

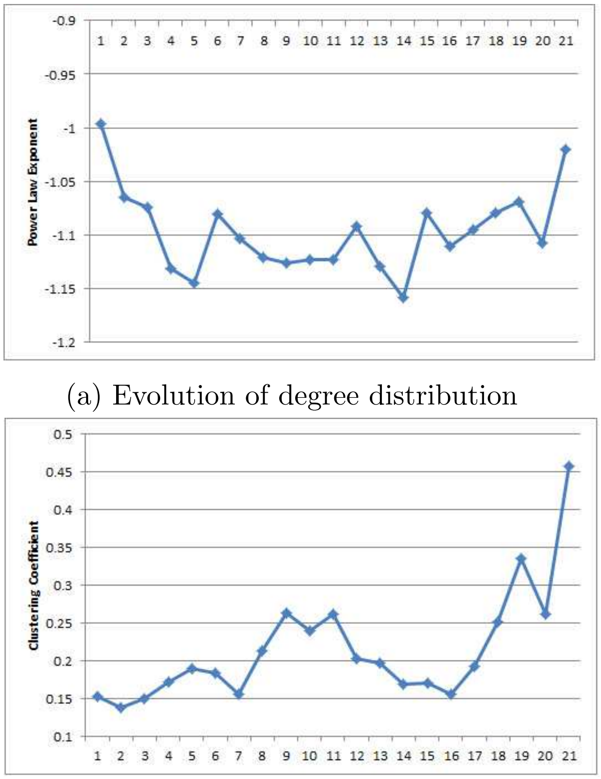
Figure 5.2 shows a ten-gene directed time-varying network over five time points (5.2(a)). For each time point, we assume that seven observations are available. The thickness of the edge indicates the strength of the interaction. Blue edges indicate stimulative interactions, whereas red edges indicate repressive or inhibitive interactions. In order to show the importance of the *weighted* l_1 formulation , we track the network using the classical Kalman filter (5.2(c)) and the *weighted* l_1 Kalman filter (5.2(b)). It can be seen that the *weighted* l_1 constraint is essential in imposing the sparsity of the network, hence significantly reducing the false positive rate.

In order to obtain a more meaningful statistical evaluation of the proposed *weighted* l_1 Kalman, we randomly generated 1000 sparse ten-gene networks evolving over five time points. The true positive (TP), true negative (TN), false positive (FP), false negative (FN) rates as well as the sensitivity, specificity, accuracy and precision are shown in Table 5.1. The results reported in Table 5.1 do not take into account the sign or strength of the interactions, but consider only the presence or absence of an interaction between two genes. Observe that the TP rate of the classical Kalman filter is high because the Kalman filter is very dense and contains many spurious connections. This leads to an “artificially” high sensitivity (97% ability to detect edges) but a very low specificity (50% ability to detect the absence of an interaction or sparsity) for the Kalman filter. The *weighted* l_1 Kalman filter achieves a good balance between sensitivity (95%) and specificity (72%).

	TP	TN	FP	FN	sensitivity	specificity
Classical Kalman	71.06%	13.60%	13.11 %	2.22%	0.97	0.50
Compressive-Kalman	80.21%	11.527%	4.32%	3.93%	0.95	0.72

Table 5.1: Performance analysis of the Compressive-Kalman filter and the classical Kalman filter.

5.2 Time-Varying Gene Networks in *Drosophila melanogaster*



(a) Evolution of degree distribution

(b) Evolution of clustering coefficient

Figure 5.3: Temporal network characteristics: (a) Evolution of the degree distribution using its power law exponent; (b) Evolution of the clustering coefficientsfor each snapshort of the temporal network.

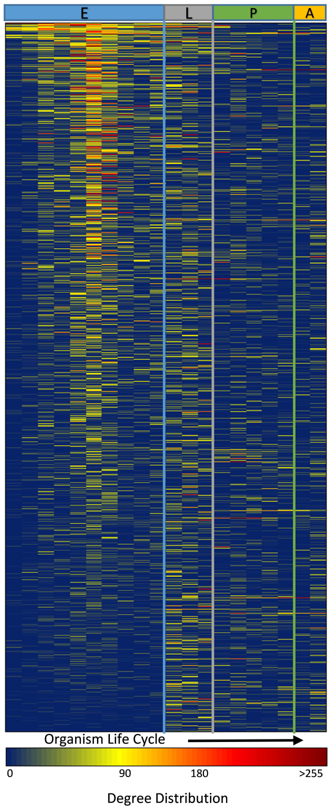


Figure 5.4: Gene degree connectivity ordered by onset of their first increase. Each row represents data for one gene, and each column is a developmental time point; blue indicates low degrees and red indicates high degrees.

A genome-wide microarray profiling of the life cycle of the Drosphila melanogaster revealed the evolving nature of the gene expression patterns during the time course of its development [3]. In this study, cDNA microarrays were used to analyze the RNA expression levels of 4028 genes in wild-type flies examined during 66 sequential time periods beginning at fertilization and spanning embryonic, larval, pupal and the first 30 days of adulthood. Since early embryos change rapidly, overlapping 1-hour periods were sampled; adults were sampled at multiday intervals [3]. The time points span the embryonic (samples 1-30; time E01h till E2324h), larval (samples 31-40; time L24h till L105h), pupal (samples 41-58; M0h till M96h) and adulthood (samples 59-66; A024h till A30d) periods of the organism.

Costello *et al.* [10] normalized the Arbeitman *et al.* raw data [3] using the optimized local intensity-dependent normalization (OLIN) algorithm [22]. Details of the normalization protocol can be found at `http://www.sciencemag.org/content/suppl/2002/09/26/297.5590.2270.DC1/ArbeitmanSOM.pdf`. In their procedure, a gene may be flagged for several reasons: the corresponding transcript not being expressed under the considered condition, the amplification of the printed cDNA was reported as “failed” in the original data, or the data is missing for technical reasons.

A statistical test was also conducted to determine if the expression of a labeled sample is significantly above the distribution of background values. Spots with a corrected p-value greater than 0.01 were considered absent (or within the distribution of background noise). In this study, we downloaded the Costello *et al.* dataset [10] and considered the unflagged genes only, which amount to a total of 1863 genes.

The *weighted* l_1 Kalman filter was used to estimate 21 dynamic gene networks, one per 3 time points, during the life cycle of Drosophila melanogaster. Figure 5.5 shows the estimated networks, where edges with absolute strength less than 10^{-3} were set to zero. The networks were visualized in Cytoscape using a force-directed layout [48]. Markov clustering [13] was used to identify clusters within each network. Clusters containing more than thirty gene were tested for functional enrichment using the BiNGO plugin for Cytoscape [34]. The Gene Ontology term with the highest enrichment in a particular cluster was used to label the cluster on the network. The changing connectivity patterns are an evident indication of the evolution of gene connectivity over time.

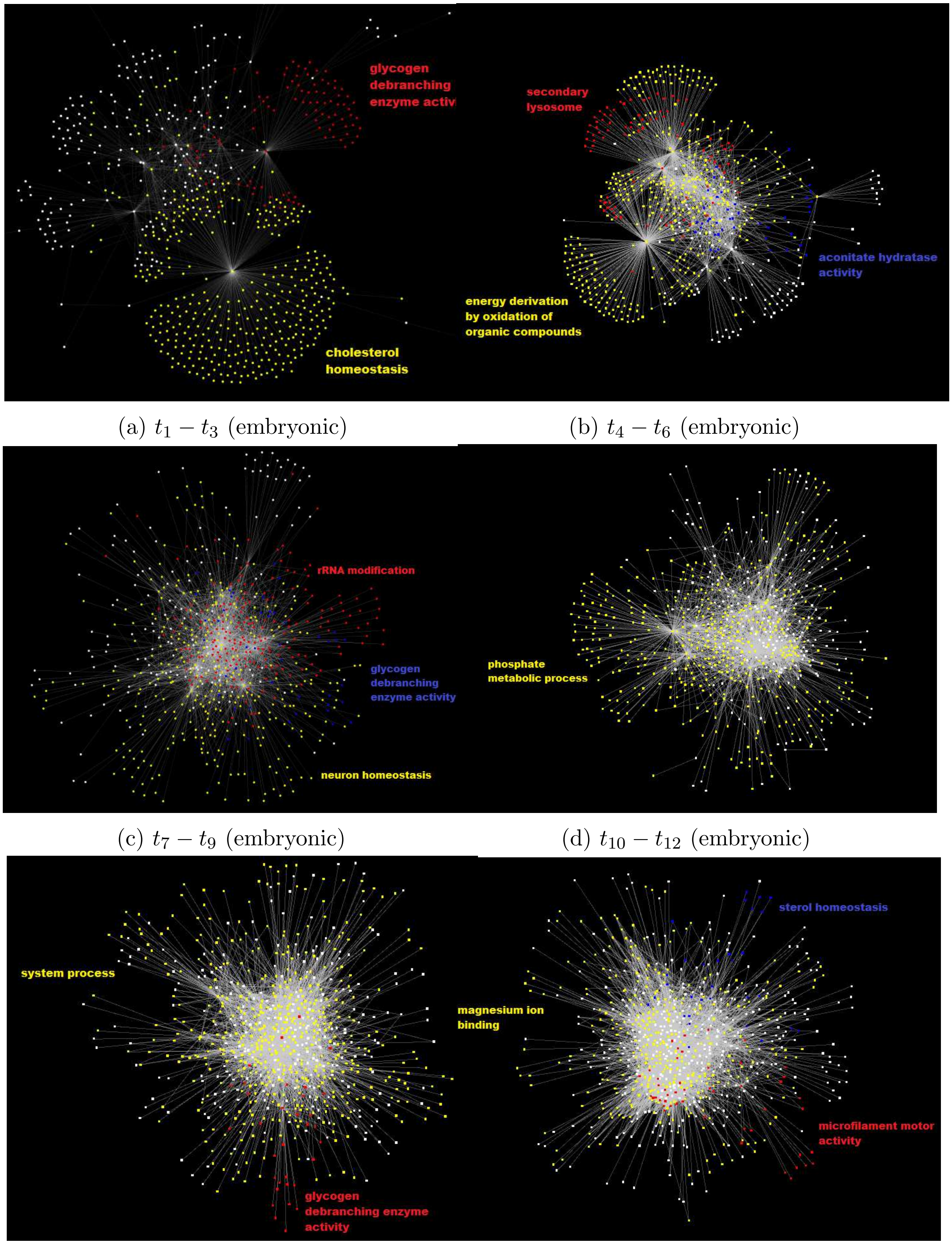
Figure 5.4 shows the evolution of the degree connectivity of each gene as a function of time. This plot helps visualize the hubs (high degree nodes) at each time point; and shows which genes are active during the phases of the organism's development. It is clear that certain genes are mainly active during specific developmental phases (transient genes), whereas others seem to play a role during the entire developmental process (permanent genes).

We quantified the structural properties of the temporal network by its degree distribution and clustering coefficient. We found that the degree distribution of each snapshot network follows a power law distribution, which indicates that the networks self-organize into a scale-free state (a global property). The power law exponents of the snapshot networks are plotted in Fig. 5.3(a). The clustering coefficient, shown in Fig. 5.3(b), measures the cliquishness of a typical neighborhood (a local property) or the degree of coherence inside potential functional modules. Interestingly, the trends (maximums and minimums) of the degree distribution and the clustering coefficients over time corroborate the results in [1], except for the clustering coefficient during early embryonic period. The Compressive-Kalman filter found a small clustering coefficient in early embryonic stage, whereas the model-based Tesla algorithm in [1] reported a high clustering coefficient for that phase.

To show the advantages of dynamic networks over a static network, we compared the recovered interactions against a list of known undirected gene interactions hosted in Flybase. The Compressive-Kalman algorithm was able to recover 1065 gene interactions (ignoring all interactions smaller or equal than 10^{-3}). The static network, computed as one network across all time periods using the algorithm in [44], recovers 248 interactions. Using the segmentation approach, we also computed four networks, where each network uses the number of samples in each developmental phase of the organism (embryonic, larval, pupal and adulthood). The embryonic-stage network uses the 30 time points sampled during the embryonic phase, and recovers 121 interactions. The larval-stage network uses the 9 time points available for the larval phase, and recovers 28 known interactions. The pupal-stage network uses 18 time points collected during the pupal period, and recovers 125 interactions. The adult-stage network utilizes 8 time points sampled during adulthood, and recovers 41 interactions. Hence, in total, the segmentation approach recovers 315 interactions. The dynamic networks of Tesla [1] were able to recover 96 known interactions. We mention that, in [1], the network size was 4028 genes, whereas we considered a subset of 1863 unflagged genes. Thus, Tesla's recovery rate is 2.4%, whereas Compressive-Kalman filter's recovery rate is 57.2%. The low recovery rate of Tesla in [1] may be due to the presence of spurious samples since flagged genes were included in the networks.

The proposed algorithm and the constraint was first tested in MATLAB ®for testing and validation of the algorithm. Once the accuracy and effectiveness of the algorithm was ascertained, an HPC based implementation of the algorithm was developed so that it could be scaled to the levels of modern HPC magnitude, this enabled us to actually examine a large number of genes. The parallel formulation of the problem helped us in realizing a highly scalable version of the process. Thus each HPC core runs code to calculate each gene at a time, while the communication between these individual processes is coordinated by the Open Message Passing Interface (Open MPI). Due to the mathematical complexity of the problem, the computation of very large matrices (e.g. for a p gene network, the Kalman filter requires a $p \times p$ covariance matrix), both the Intel(R) C Compiler and the Intel(R) Math Kernel Library (Intel(R) MKL) were used on a Linux based platform for maximum performance. This approach enabled an implementation that is highly efficient, inherently parallel (for matrix multiplications etc.) and has built in support for the HPC architecture. The implementation starts by the main MPI process spawning the child processes, each child process is assigned an individual gene which it computes based on the gene expression data that is made available to it using the file system.

The child process returns the computed result to the main process, which assigns it the next gene and in this way the process continues till the last gene. Finally the master process puts together the computed results in a contagious matrix. To minimize the memory requirements of the system, the sequential nature of the large covariance matrices is exploited to ease the burden of memory management on the Linux Kernel. To run this implementation, the Razor II HPC system at Univ of Arkansas at Fayetteville was used. It has sixteen cores per node, with 32 Giga Bytes of memory, each node interconnected using a 40Gbps QLogic quad-data rate QDR InfiniBand. 40 such nodes were employed at a given time, though the computing system at University of Arkansas at Fayetteville has 112 total nodes. This implementation also supports increasing the number of genes and is thus completely scalable for future investigations.


glycogen debranching enzyme activ
cholesterol homeostasis
secondary lysosome
energy derivation by oxidation of organic compounds
aconitate hydratase activity
rRNA modification
glycogen debranching enzyme activity
neuron homeostasis
phosphate metabolic process
system process
glycogen debranching enzyme activity
sterol homeostasis
magnesium ion binding
microfilament motor activity

(a) $t_1 - t_3$ (embryonic)

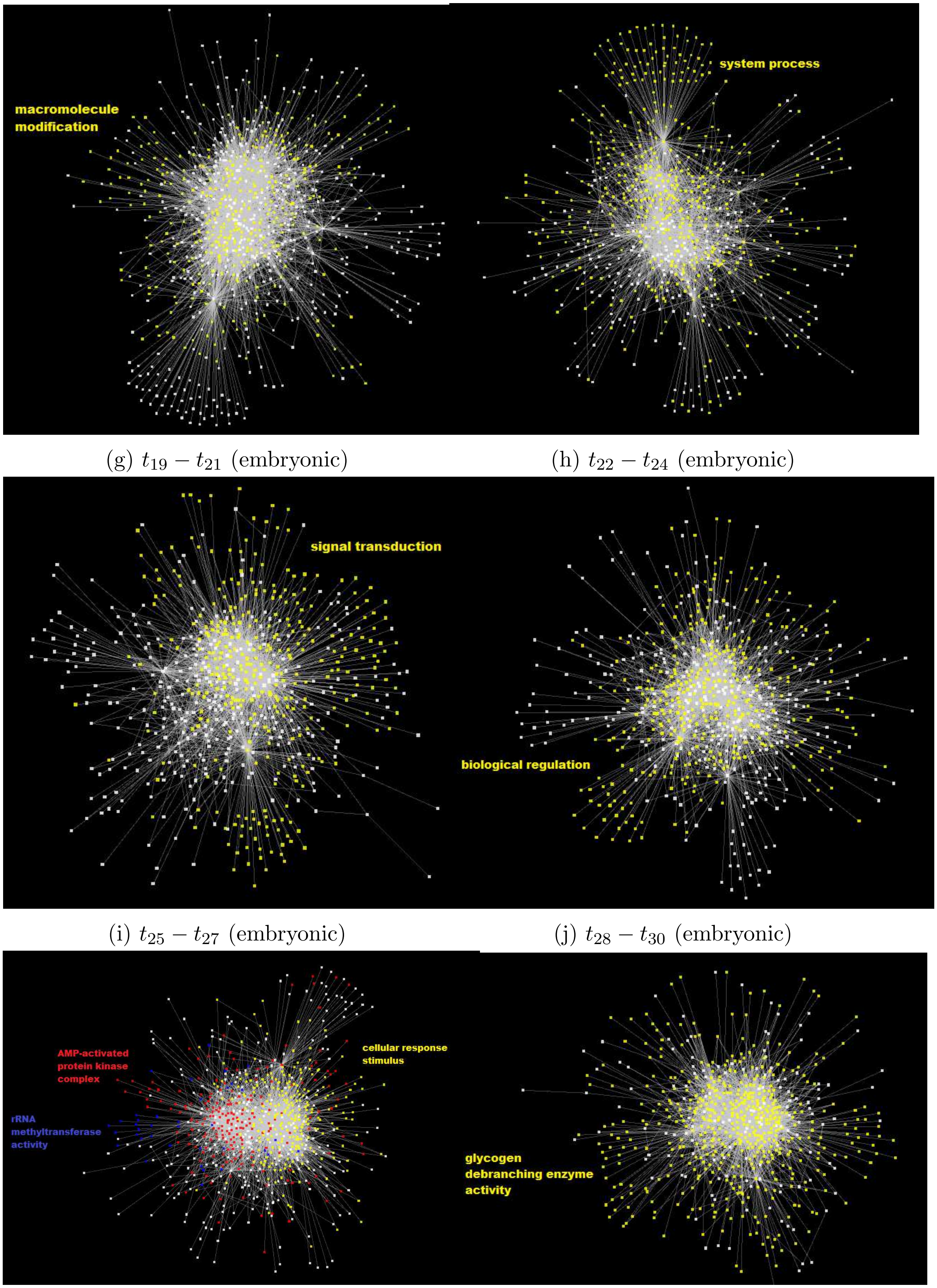
(b) $t_4 - t_6$ (embryonic)

(c) $t_7 - t_9$ (embryonic)

(d) $t_{10} - t_{12}$ (embryonic)

(e) $t_{13} - t_{15}$ (embryonic)

(f) $t_{16} - t_{18}$ (embryonic)


macromolecule modification
system process
signal transduction
biological regulation
AMP-activated protein kinase complex
cellular response stimulus
rRNA methyltransferase activity
glycogen debranching enzyme activity

(g) $t_{19} - t_{21}$ (embryonic)

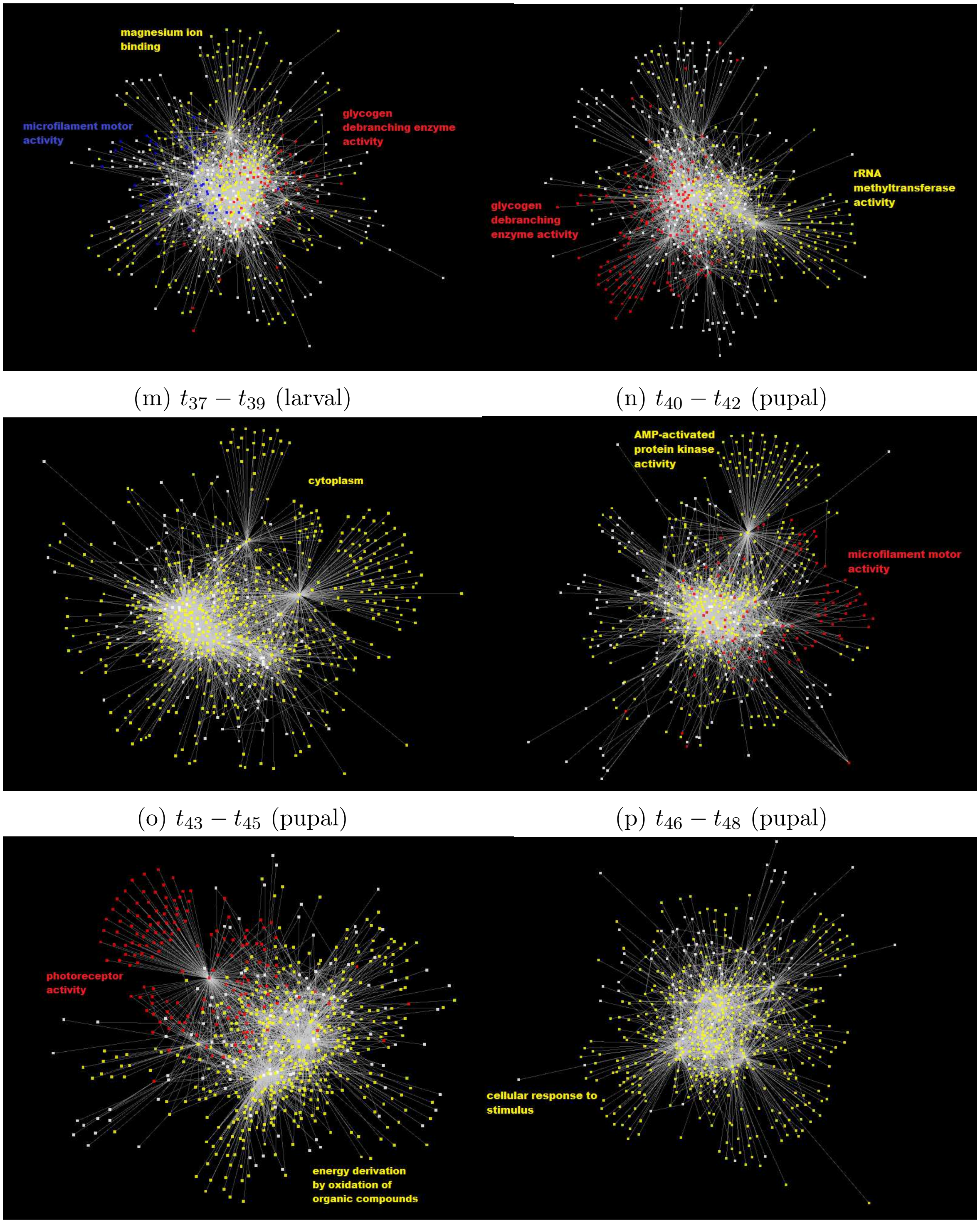
(h) $t_{22} - t_{24}$ (embryonic)

(i) $t_{25} - t_{27}$ (embryonic)

(j) $t_{28} - t_{30}$ (embryonic)

(k) $t_{31} - t_{33}$ (larval)

(l) $t_{34} - t_{36}$ (larval)


magnesium ion binding
microfilament motor activity
glycogen debranching enzyme activity
glycogen debranching enzyme activity
rRNA methyltransferase activity
cytoplasm
AMP-activated protein kinase activity
microfilament motor activity
photoreceptor activity
energy derivation by oxidation of organic compounds
cellular response to stimulus

(m) $t_{37} - t_{39}$ (larval)

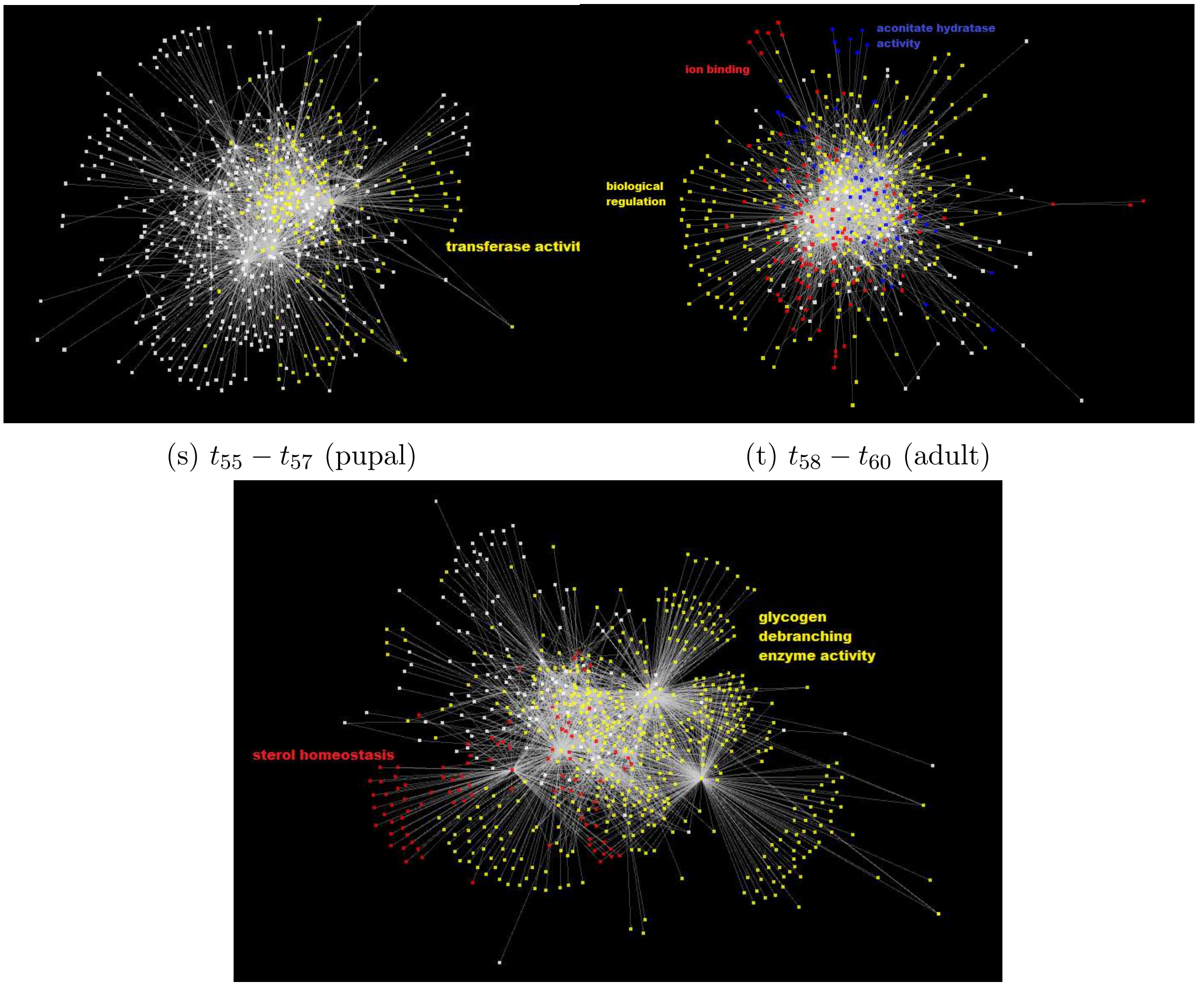
(n) $t_{40} - t_{42}$ (pupal)

(o) $t_{43} - t_{45}$ (pupal)

(p) $t_{46} - t_{48}$ (pupal)

(q) $t_{49} - t_{51}$ (pupal)

(r) $t_{52} - t_{54}$ (pupal)



(s) $t_{55} - t_{57}$ (pupal)

(t) $t_{58} - t_{60}$ (adult)

(u) $t_{61} - t_{63}$ (adult)

Figure 5.5: Snapshots of the time-varying networks at 21 time epochs (3 time points for every network) depicting the connectivity patterns between the 1863 genes of the *Drosophila melanogaster* during its development cycle. Genes are represented as nodes and interactions as edges. Colored nodes are sets of genes enriched for Gene Ontology summarized by the indicated terms. The nodes were distributed using a force-directed layout in Cytoscape.

Conclusion

Due to the dynamic nature of biological processes, biological networks undergo systematic rewiring in response to cellular requirements and environmental changes. These changes in network topology are imperceptible when estimating a static "average" network for all time points. The dynamic view of genetic regulatory networks reveals the temporal information about the onset and duration of genetic interactions; in particular showing that few genes are permanent players in the cellular function while others act transiently during certain phases or "regimes" of the biological process. It is, therefore, essential to develop methods that capture the temporal evolution of genetic networks, and allow the study of phase-specific genetic regulation and the prediction of network structures under given cellular and environmental conditions.

In this Thesis, we formulated the reverse-engineering of time-varying networks, from a limited number of observations, as a tracking problem in a compressed domain. Under the assumption of linear dynamics, we derived the stable *weighted* l_1 Kalman filter, which provides the optimal minimum mean-square sparse estimate of the connectivity structure. The estimated networks reveal that genetic interactions undergo significant rewiring during the developmental process of an organism such as the *Drosophila Melanogaster*. We anticipate that these topological changes and phase-specific interactions apply to other genetic networks underlying dynamic biological processes, such as cancer progression, therapeutic treatment and development.

Finally, we anticipate that the rapid breakthroughs in genomic technologies for measurement and data collection will make the static representation of biological networks obsolete and establish instead the dynamic perspective of biological interactions. The code and relevant data for this work is available at `http://users.rowan.edu/~bouaynaya/EURASIP2014.html`.

Bibliography

[1] A. Ahmed, L. Song, and E. P. Xing. Time-varying networks: Recovering temporally rewiring genetic networks during the life cycle of drosophila melanogaster. Technical report, Carnegie Mellon University, 2009.

[2] A. Ahmed and E. P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106(29):11878–11883, July 2009.

[3] M. Arbeitman, E. Furlong, F. Imam, E. Johnson, B. Null, B. Baker, M. Krasnow, M. Scott, R. Davis, and K. White. Gene expression during the life cycle of Drosophila melanogaster. *Science*, 297:2270–2275, 2002.

[4] J. Bobin, J.-L. Starck, and R. Ottensamer. Compressed sensing in astronomy. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):718–726, 2008.

[5] S. Boyd. Linear Dynamical Systems Lecture 12. `http://stanford.edu/class/ee363/lectures/lyap.pdf`. Accessed: 2014-06-18.

[6] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, chapter Lyapunov Stability Criteria, pages 437–445. Society for Industrial and Applied Mathematics - SIAM, 1994.

[7] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489 – 509, February 2006.

[8] J. Cao, X. Qi, and H. Zhao. Modeling gene regulation networks using ordinary differential equations. *Methods in Molecular Biology*, 802:185–197, 2012.

[9] V. Chaitankar, P. Ghosh, E. J. Perkins, P. Gong, Y. Deng, and C. Zhang. A novel gene network inference algorithm using predictive minimum description length approach. *BMC Systems Biology*, 4(Suppl 1)(S7), May 2010.

[10] J. C. Costello, M. M. Dalkilic, and J. R. Andrews. Microarray normalization protocols. Personal communication to FlyBase, 2008.

[11] E. H. Davidson, J. P. Rast, P. Oliveri, A. Ransick, C. Calestani, C.-H. Yuh, T. Minokawa, G. Amore, V. Hinman, C. Arenas-Mena, O. Otim, C. T. Brown, C. B. Livi, P. Y. Lee, R. Revilla, A. G. Rust, Z. J. Pan, M. J. Schilstra, P. J. C.

Clarke, M. I. Arnone, L. Rowen, R. A. Cameron, D. R. McClay, L. Hood, and H. Bolouri. A genomic regulatory network for development. *Science*, 295(5560):1669–1678, March 2002.

[12] F. Dondelinger, S. Lebre, and D. Husmeier. Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Machine Learning*, 90(2):191–230, February 2013.

[13] S. V. Dongen. Performance criteria for graph clustering and markov cluster experiments. Technical report, National Research Institute for Mathematics and Computer Science, 2000.

[14] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.

[15] J. Ernst, O. Vainas, C. T. Harbison, I. Simon, and Z. Bar-Joseph. Reconstructing dynamic regulatory maps. *Molecular Systems Biology*, 3(74), 2007.

[16] H. M. Fathallah-Shaykh, J. L. Bona, and S. Kadener. Mathematical model of the drosophila circadian clock: Loop regulation and transcriptional integration. *Biophysical Journal*, 97(9):2399–2408, November 2009.

[17] P. Fearnhead. Exact and efficient bayesian inference for multiple change point problems. *Statistics and Computing*, 16(2):203 – 213, July 2006.

[18] M. Fornasier and H. Rauhut. *Handbook of Mathematical Methods in Imaging*, chapter Compressive sensing. Springer, 2011.

[19] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4):1569 – 1585, April 2011.

[20] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, February 2004.

[21] N. Friedman, M. Linial, I. Nachman, and D. Peter. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[22] M. E. Futschik and T. Crompton. OLIN: optimized normalization, visualization and quality testing of two-channel microarray data. *Bioinformatics*, 21(8):1724–1726, 2005.

[23] J. D. Geeter, H. V. Brussel, J. D. Schutter, and M. Decreton. A smoothly constrained Kalman filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1171 – 1177, October 1997.

[24] S. Gillijns. *Kalman Filtering Techniques for System Inversion and Data Assimilation.* PhD thesis, Katholieke Universiteit Leuven (University of Leuven), Leuven, Belgium, December 2007.

[25] M. Grzegorczyk and D. Husmeier. Non-stationary continuous dynamic bayesian networks. In *Neural Information Processing Systems*, pages 682–690, December 2009.

[26] F. Guo, S. Hanneke, W. Fu, and E. P. Xing. Recovering temporally rewiring networks: a model-based approach. In *Proceedings of the international conference on Machine learning*, pages 321 – 328, 2007.

[27] H. Hache, H. Lehrach, and R. Herwig. Reverse engineering of gene regulatory networks: a comparative study. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009, April 2009.

[28] S. Hayward. *Mathematics in Signal Processing IV*, chapter Constrained Kalman filter for least-squares estimation of time-varying beamforming weights, pages 113–125. Oxford University Press, 1998.

[29] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A*, 454(1971):903–995, March 1998.

[30] S. J. Julier and J. J J LaViola. On Kalman filtering with nonlinear equality constraints. *IEEE Transactions on Signal Processing*, 55(6):2774–2784, June 2007.

[31] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

[32] E. E. Kuruoğlu, X. Yang, Y. Xu, and T. S. Huang. Time varying dynamic bayesian network for nonstationary events modeling and online inference. *IEEE Transactions on Signal Processing*, 59(4):1553 – 1568, April 2011.

[33] N. M. Luscombe, M. M. Babu, H. Yu, M. Snyder, S. A. Teichmann, and M. Gerstei. Genomic analysis of regulatory network dynamics reveals large topological changes. *Letters to Nature*, 431:308–312, September 2004.

[34] S. Maere, K. Heymans, and M. Kuiper. Bingo a cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16):3448–3449, 2005.

[35] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, and K. R. Allison. Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8):796–804, July 2012.

[36] P. E. Meyer, K. Kontos, F. Lafitte, and G. Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP Journal on Bioinformatics and Systems Biology*, 79879, 2007.

[37] H. Miyashita, T. Nakamura, Y. Ida, T. Matsumoto, and T. Kaburagi. Nonparametric Bayes-based heterogeneous Drosophila Melanogaster gene regulatory network inference: T-process regression. In *Proceedings of the* IASTED *International Conference on Artificial Intelligence and Applications*, pages 51–58, 2013.

[38] K. Murphy and S. Mian. Modeling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, 1999.

[39] S. H. Nielsen and T. D. Nielsen. Adapting Bayes network structures to non-stationary domains. *International Journal of Approximate Reasoning*, 49(2):379–397, October 2008.

[40] A. Noor, E. Serpedin, M. Nounou, and H. N. Nounou. Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity. *IEEE Transactions on Computational Biology and Bioinformatics*, 9(4):1203–1211, August 2012.

[41] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77:103 – 124, May 2008.

[42] R. Prado, G. Huerta, and M. West. Bayesian time-varying autoregressions: Theory, methods and applications. *Resenhas: Journal of the Institute of Mathematics and Statistics of the University of Sao Paolo*, 4:405–422, 2000.

[43] A. Rao, A. O. Hero, D. J. States, and J. D. Engel. Inferring time-varying network topologies from gene expression data. *EURASIP Journal on Bioinformatics and Systems Biology*, pages 1–12, 2007.

[44] G. Rasool, N. Bouaynaya, H. M. Fathallah-Shaykh, and D. Schonfeld. Inference of genetic regulatory networks using regularized likelihood with covariance estimation. In *IEEE Statistical Signal Processing Workshop*, August 2012.

[45] G. Rasool, N. Bouaynaya, H. M. Fathallah-Shaykh, and D. Schonfeld. Inference of genetic regulatory networks using regularized likelihood with covariance estimation. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 560–563. IEEE, 2012.

[46] J. W. Robinson and A. J. Hartemink. Learning non-stationary dynamic bayesian networks. *The Journal of Machine Learning Research*, 11:3647–3680, 2010.

[47] J. W. Robinson and E. J. Hartemink. Non-stationary dynamic bayesian networks. In *Neural Information Processing Systems*, pages 1369–1376, 2008.

[48] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, November 2003.

[49] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[50] D. Simon. *Optimal State Estimation*. Wiley, 2006.

[51] L. Song, M. Kolar, and E. Xing. Time-varying dynamic bayesian networks. In *The Neural Information Processing Systems*, December 2009.

[52] G. Tauböck, F. Hlawatsch, D. Eiwen, and H. Rauhut. Compressive estimation of doubly selective channels in multicarrier systems: Leakage effects and sparsity-enhancing processing. *IEEE Journal of selected topics in signal processing*, 4(2):255–271, 2010.

[53] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267 – 288, 1996.

[54] A. Tucker and X. Liu. A bayesian network approach to explaining time series with changing structure. *Intelligent Data Analysis*, 8(8):469 – 480, October 2004.

[55] Z. Wang, X. Liu, Y. Liu, J. Liang, and V. Vinciotti. An extended Kalman filtering approach to modeling nonlinear dynamic gene regulatory networks via short gene expression time series. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(3):410–419, July 2009.

[56] W. Wen and H. F. Durrant-Whyte. Model-based multi-sensor data fusion. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1720 – 1726, May 1992.

[57] K. W. Xidian, J. Zhang, F. Shen, and L. Shi. Adaptive learning of dynamic bayesian networks with changing structures by detecting geometric structures of time series. *Knowledge and Information Systems*, 17(1):121–133, October 2008.

[58] M. Yamashita, K. Fujisawa, and M. Kojima. Sdpara: Semidefinite programming algorithm parallel version. *Parallel Computing*, 29(8):1053 – 1067, 2003.

[59] M. M. Zavlanos, A. A. Julius, S. P. Boyd, and G. J. Pappas. Inferring stable genetic networks from steady-state data. *Automatica*, 47(6):1113 – 1122, 2011. Special Issue on Systems Biology.

[60] W. Zhao, E. Serpedin, and E. R. Dougherty. Inferring connectivity of genetic regulatory networks using information-theoretic criteria. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):262–274, April 2008.

[61] P. Zoppoli, S. Morganella, and M. Ceccarelli. Time delay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(154), 2010.