Biometrics 73, 1231–1242
December 2017

DOI: 10.1111/biom.12685

Estimating Time-Varying Directed Gene Regulation Networks

Yunlong Nie, LiangLiang Wang, and Jiguo Cao*

Department of Statistics and Actuarial Science, Simon Fraser University, British Columbia, Canada
**email:* jiguo_cao@sfu.ca

Summary. The problem of modeling the dynamical regulation process within a gene network has been of great interest for a long time. We propose to model this dynamical system with a large number of nonlinear ordinary differential equations (ODEs), in which the regulation function is estimated directly from data without any parametric assumption. Most current research assumes the gene regulation network is static, but in reality, the connection and regulation function of the network may change with time or environment. This change is reflected in our dynamical model by allowing the regulation function varying with the gene expression and forcing this regulation function to be zero if no regulation happens. We introduce a statistical method called *functional SCAD* to estimate a time-varying sparse and directed gene regulation network, and simultaneously, to provide a smooth estimation of the regulation function and identify the interval in which no regulation effect exists. The finite sample performance of the proposed method is investigated in a Monte Carlo simulation study. Our method is demonstrated by estimating a time-varying directed gene regulation network of 20 genes involved in muscle development during the embryonic stage of Drosophila melanogaster.

Key words: Ordinary differential equation; Smoothing spline; Sparse estimation; System identification.

1. Introduction

Gene regulation networks (GRN) have gained a lot of attention from biologists, geneticists, and statisticians in recent years. A variety of methods have been developed to infer GRN based on gene expression data such as Boolean networks (Thomas, 1973; Laubenbacher and Stigler, 2004; Mehra, Hu, and Karypis, 2004), information theory models (Steuer et al., 2002; Stuart et al., 2003), and Bayesian networks (Jensen, 1996; Needham et al., 2007). However, these methods only focus on static GRN, that is, the network with the time-invariant topology given a set of genes. In fact, the regulation effect between a given pair of genes may change dramatically over the course of a biological process (Luscombe et al., 2004). Consequently, the GRN topology may be time-varying.

Ordinary differential equation (ODE) models (Cao and Zhao, 2008; Lu et al., 2011; Wu et al., 2014) have become popular to model the dynamical changes (both decreasing and increasing) of a target gene expression as a function of expression levels of all regulatory genes. The estimated regulation effect is also time-varying due to the variation of the regulatory gene expression. For instance, Cao and Zhao (2008) focused on parameter estimation for the ODE model when the type of regulation effect between two genes is known.

When the number of genes in the network is large, a sparse model is often preferable. But model selection (identification of true regulatory genes) has not been well addressed in the high-dimension context, where the total number of genes available far exceeds the number of gene expression measures. To solve this problem, Lu et al. (2011) reduced the dimension by first clustering genes into modules, then estimating a linear additive ODE model on the module level instead of the gene level. However, this method fails to capture the dynamical regulation effect at the gene level. In addition, the linear assumption on the regulation function may be impractical in many complex scenarios. Wu et al. (2014) modeled the regulation effect using a nonlinear function and solved the curse of dimensionality by adopting shrinkage techniques such as group LASSO (Yuan and Lin, 2006) and adaptive LASSO (Zou, 2006). On the other hand, once the regulatory genes are selected, the global topology of the GRN will stay constant during the whole process. However, in reality, the regulation effect from one gene might exist only in a certain time period rather than during the whole biological process.

Thus, we would prefer a flexible model in which the global topology of the estimated GRN is time-varying. Several methods have been proposed to estimate time-varying networks. For instance, Hanneke, Fu, and Xing (2010) extended exponential random graph models (ERGMs) to model the topology change of a time-varying social network based on a number of evolution statistics such as edge-stability, reciprocity, and transitivity. However, their method can only recover the undirected interactions between the nodes, and can only be scaled up to small-scale networks because of the sampling algorithm. Song, Kolar, and Xing (2009) and Kolar et al. (2010) proposed a kernel-reweighted logistic regression model with the L_1 penalty to estimate a time-varying GRN, which can be scaled up to large networks. Another advantage of their method is to allow both smoothing and sudden changes in the network topology. Kolar and Xing (2009) established the consistency of the kernel-smoothing L_1 regularized method. But both Song et al. (2009) and Kolar et al. (2010) only took binarized gene expression data as the input, and were also limited to undirected interactions between the genes. To the best of our knowledge, no existing methods use differential equations to model directed time-varying networks and estimate directed time-varying networks from continuous gene expression data. This is the main focus of this article.

Our article makes two crucial contributions. First, we model the dynamical feature of directed GRN using a high-dimensional nonlinear ODE model, in which the regulation function is a nonlinear function of the regulatory gene expression and is exactly zero in those intervals when no regulation effect happens. Hence our model allows the global topology of the directed GRN to be time-varying. Second, we propose a carefully-designed shrinkage technique called the functional smoothly clipped absolute deviation (fSCAD) method to do three tasks simultaneously: detecting significant regulatory genes for any given gene, identifying the intervals in which the significant regulatory genes have the regulation effect, and estimating the nonlinear regulation function without any parametric assumption. In addition, an R package called "flyfuns" is developed to implement our proposed method, and is available at https://github.com/YunlongNie/flyfuns.

The rest of this article is organized as follows: Details of our method are introduced in Section 2. Our method is demonstrated with a real data example in Section 3, where, we estimate a time-varying directed GRN among 20 Drosophila melanogaster genes during the embryonic stage. Section 4 presents a simulation study to investigate the finite sample performance of our method in comparison with conventional methods. Conclusions are given in Section 5.

2. Method

2.1. *An ODE Model for Time-Varying Directed Gene Regulation Networks*

Suppose a time-varying directed GRN has G genes in total, and their expressions are measured in a certain time period. The following ODE model relates the rate of change of one target gene expression to the expression of all genes in the network:

$$\dot{X}_\ell(t) = \mu_\ell + \sum_{g=1}^{G} f_{g\ell}(X_g),\ \ \ell = 1, \dots, G, \quad t \in [0, T], \tag{1}$$

where $\dot{X}_\ell(t)$ denotes the first derivative of $X_\ell(t)$ at time t for the target gene ℓ, μ_ℓ is the intercept term, and $f_{g\ell}(X_g)$ represents the regulation function of gene g on gene ℓ. Here, we assume $\dot{X}_\ell(t)$ is known, and in Section 2.7, we will discuss how to estimate it. Note that our approach also belongs to the framework of the two-step estimation for ODE parameters (Ramsay and Silverman, 2002; Chen and Wu, 2008).

When the number of genes, G, is large, we assume that only a few genes regulate the expression of the target gene ℓ. In other words, in the ODE model (1), only a few regulation functions $f_{g\ell}(X_g) \neq 0$ and all others $f_{g\ell}(X_g) \equiv 0$. This assumption implies the sparsity of the underlying directed GRN structure.

In addition, we assume that the regulation effect of a particular regulatory gene might only be significant when its expression level is within a certain range. We use $S_{g\ell}$ to denote the support or nonzero intervals of the regulation function $f_{g\ell}$. In other words, $f_{g\ell}(X_g) \neq 0$ when $X_g \in S_{g\ell}$ and $f_{g\ell}(X_g) = 0$ when $X_g \notin S_{gl}$. This assumption results in a dynamical directed GRN with a time-varying topology, because some regulation functions may be nonzero at one time and become zero at some other time.

Without any parametric assumption on the regulation function $f_{g\ell}(X_g)$, we represent $f_{g\ell}(X_g)$ as a linear combination of basis functions

$$f_{g\ell}(X_g) = \sum_{k=1}^{K_{g\ell}} \beta_{g\ell k}\phi_{g\ell k}(X_g) = \boldsymbol{\phi}_{g\ell}^T(X_g)\boldsymbol{\beta}_{g\ell}, \tag{2}$$

where $\boldsymbol{\phi}_{g\ell}(X_g) = (\phi_{g\ell 1}(X_g), \phi_{g\ell 2}(X_g), \dots, \phi_{g\ell K_{g\ell}}(X_g))^T$ denotes the vector of basis functions, $\boldsymbol{\beta}_{g\ell} = (\beta_{g\ell 1}, \dots, \beta_{g\ell K_{g\ell}})^T$ is the corresponding vector of basis coefficients, and $K_{g\ell}$ denotes the number of basis functions. If all the elements of $\boldsymbol{\beta}_{g\ell}$ are estimated to be zero, then $f_{g\ell}(X_g) \equiv 0$, and the corresponding gene is omitted from the ODE model. On the other hand, if only a few elements of $\boldsymbol{\beta}_{g\ell}$ are estimated to be zero, then the corresponding regulation function $f_{g\ell}(X_g)$ will be strictly zero in certain intervals.

In this method, we choose B-spline basis functions due to their compact support property, they are only nonzero in a local interval (de Boor, 2001). This property is crucial for the computation efficiency and the sparse estimation of our fSCAD method. Figure 1 shows an example of the ten cubic B-spline basis functions, defined by six interior knots. We can see that each of the six basis functions in the center is nonzero over four adjacent sub-intervals. In addition, the three left-most basis functions and the three right-most basis functions are nonzero over no more than four adjacent sub-intervals.

The method proposed in the rest of this section is trying to achieve the following three tasks simultaneously: detecting those significant regulatory genes whose regulation function $f_{g\ell}(X_g) \neq 0$, identifying the nonzero intervals, $S_{g\ell}$, of these regulation functions, and estimating the nonlinear regula-

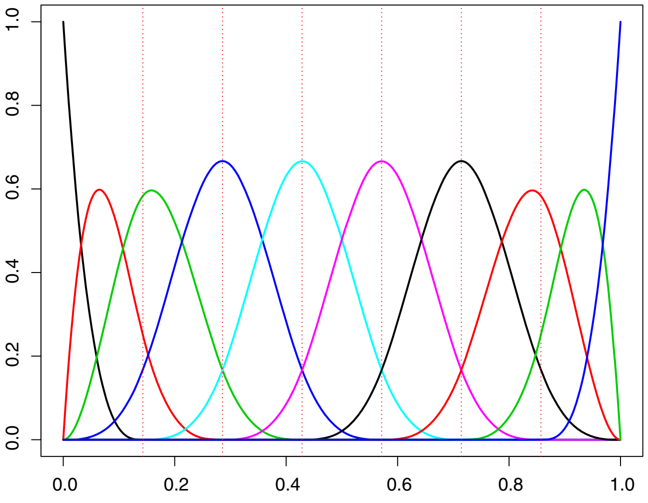

Figure 1. Ten cubic B-spline basis functions, defined by six interior knots. The locations of interior knots are indicated by vertical dashed lines. This figure appears in color in the electronic version of this article.

tion function, that is, $f_{g\ell}(X_g)$, in the corresponding nonzero intervals.

2.2. *Sparsity Penalty*

The most common way to achieve sparsity is to add a penalty term to the loss function. Our method belongs to this fashion by carefully choosing the penalty composition. Generally speaking, the main idea is to first partition each regulatory gene's whole expression domain into several subintervals. The penalty term depends on the magnitude of the regulation effect in each subinterval instead of in the entire expression domain.

The functional SCAD method was first proposed by Lin et al. (2016), which could be considered as a functional generalization of the SCAD (Fan et al., 2004). In Lin et al. (2016), they used fSCAD to estimate the coefficient function in the functional linear regression. However, they only had one functional predictor in their model, and did not consider the variable selection problem. In our work, we extend this method to do the variable selection in the high-dimensional differential equation model. At the same time, we use the fSCAD method to identify the nonzero intervals, $S_{g\ell}$, of these regulation functions and estimate the nonlinear regulation function in the estimated nonzero intervals simultaneously. Now we introduce our fSCAD penalty as follows:

The fSCAD penalty in our model is defined as

$$\sum_{g=1}^{G} \frac{M_{g\ell}}{\Delta_{x_g}} \int_{x_{gl}}^{x_{gu}} p_\lambda(|f_{g\ell}(X_g)|)dX_g,$$

where x_{gl} and x_{gu} are the lower and upper bounds of the expression of the g-th gene $X_g(t)$, $t \in [0, T]$, $\Delta_{x_g} = x_{gu} - x_{gl}$, and $M_{g\ell}$ is the number of subintervals partitioned by the knots of B-spline basis functions, so $M_{g\ell}$ is the total number of interior knots plus one. We use the cubic B-spline basis functions in our simulation and application, then $M_{g\ell} = K_{g\ell} - 3$, where $K_{g\ell}$ is the number of cubic spline basis functions. Inside the integral, $p_\lambda(\cdot)$ denotes the SCAD penalty function defined in Fan and Li (2001):

$$p_\lambda(u) = \begin{cases} \lambda u & \text{if } 0 \le u \le \lambda, \\ -\frac{u^2 - 2a\lambda u + \lambda^2}{2(a-1)} & \text{if } \lambda < u < a\lambda, \\ \frac{(a+1)\lambda^2}{2} & \text{if } u \ge a\lambda, \end{cases}$$

where a is 3.7, as suggested by Fan and Li (2001), and λ is the tuning parameter, which controls the sparsity of the regulation functions.

Let $x_0, x_1, \ldots, x_{M_{g\ell}}$ denote the sequence of the knots of B-spline basis function. Lin et al. (2016) has shown that

$$\frac{1}{\Delta_{x_g}} \int_{x_{gl}}^{x_{gu}} p_\lambda(|f_{g\ell}(X_g)|)dX_g = \frac{1}{M_{g\ell}} \lim_{M_{g\ell} \to +\infty} \times \sum_{j=1}^{M_{g\ell}} p_\lambda\left(\sqrt{\frac{M_{g\ell}}{\Delta_{x_g}} \int_{x_{g,j-1}}^{x_{gj}} [f_{g\ell}(X_g)]^2 dX_g}\right). \qquad (3)$$

From (3), one can see that the fSCAD penalty is essentially the summation of penalties in each subinterval $[x_{g,j-1}, x_{gj}]$. In each subinterval $[x_{g,j-1}, x_{gj}]$, the penalty is governed by the magnitude of the regulation effect quantified by $\int_{x_{g,j-1}}^{x_{gj}} [f_{g\ell}(X_g)]^2 dX_g$. Thus fSCAD compares all gene regulation effects on each subinterval and tends to shrink those insignificant regulation effects towards zero without over shrinking those significant regulation effects. The value of λ determines the size of the shrinkage effect. For instance, a larger value of λ will lead to smaller nonzero region for $f_{g\ell}(X_g)$. If the nonzero region of the regulation function does not exist, the corresponding gene is omitted from the ODE model. Thus we identify those genes that have no regulation effects. On the other hand, if the nonzero region does exist, the corresponding gene will have a significant regulation effect when its expression level is within the estimated nonzero region.

In comparison with group LASSO, the advantage of fSCAD is that it is able to discover a strong regulation effect even when this effect only exists for a small subinterval. Essentially, fSCAD penalizes the gene regulation function based on their regulation effects on each subinterval, whereas group LASSO cannot achieve this because its penalty depends on the regulation effect in the whole interval. For instance, if the regulation effect of one gene only exists in a short interval, group LASSO will still shrink the effect to zero and ignore its regulation effect completely even though the magnitude of the regulation effect is quite large in that short interval.

2.3. *Roughness Penalty*

We assume that the regulation function $f_{g\ell}(X_g)$ is a smooth function of X_g because the regulation effect is not expected to change dramatically when the regulatory gene's expression has a small change.

In order to obtain a smooth regulation function, we introduce a roughness penalty. For a certain regulatory gene X_g, we define the roughness penalty as:

$$\left\|\frac{df^2_{g\ell}(X_g(t))}{dt^2}\right\|^2 = \int_0^T \left(\frac{d^2 f_{g\ell}(X_g(t))}{dt^2}\right)^2 dt.$$

Based on the basis function expansion for the regulation function $f_{g\ell}(X_g(t))$ defined in equation (2), one can show that the second derivative of $f_{g\ell}(X_g(t))$ can be calculated as

$$\frac{d^2 f_{g\ell}(X_g(t))}{dt^2} = \sum_{k=1}^{K_{g\ell}} \beta_{g\ell k} \frac{d^2 \phi_{g\ell k}(X_g(t))}{dt^2} = \sum_{k=1}^{K_{g\ell}} \beta_{g\ell k} d_{g\ell k},$$

where

$$d_{g\ell k} = \frac{d^2\phi_{g\ell k}(X_g(t))}{dt^2} = \frac{d^2\phi_{g\ell k}}{dX_g^2}\left(\frac{dX_g}{dt}\right)^2 + \frac{d\phi_{g\ell k}}{dX_g}\frac{d^2 X_g}{dt^2}. \qquad (4)$$

The roughness penalty for all G regulation functions is given as

$$R_\ell = \sum_{g=1}^{G} \left|\left| \frac{df_{g\ell}^2(X_g(t))}{dt^2} \right|\right|^2 = \sum_{g=1}^{G} \int_0^T \left(\sum_{k=1}^{K_{g\ell}} \beta_{g\ell k} d_{g\ell k} \right)^2 dt. \tag{5}$$

2.4. *Parameter Estimation*

Combining the fSCAD penalty (3) and the roughness penalty (5), we estimate $f_{g\ell}(X_g)$ via minimizing the following loss function:

$$\begin{aligned} Q(\boldsymbol{\beta}_\ell) = \frac{1}{n} \sum_{i=1}^{n} \left(\dot{X}_\ell(t_i) - \sum_{g=1}^{G} f_{g\ell}(X_g(t_i)) \right)^2 \\ + \gamma \sum_{g=1}^{G} \left|\left| \frac{df_{g\ell}^2(X_g(t))}{dt^2} \right|\right|^2 \\ + \sum_{g=1}^{G} \frac{M_{g\ell}}{\Delta_{x_g}} \int_{x_{gl}}^{x_{gu}} p_\lambda(|f_{g\ell}(X_g)|) dX_g, \end{aligned} \tag{6}$$

where $\boldsymbol{\beta}_\ell = (\boldsymbol{\beta}_{1\ell}^T, \boldsymbol{\beta}_{2\ell}^T, \dots, \boldsymbol{\beta}_{G\ell}^T)^T$, is a length GK column vector of all basis function coefficients.

The first term in (6) quantifies the goodness of fit to the derivative. The second term is the summation of the roughness penalty for each regulation function, and γ is the smoothing parameter which controls the smoothness of all regulation functions. The last term corresponds to the fSCAD penalty.

For simplicity, we recast each part of the loss function in (6) into a matrix form. Following the notations in equation (2), the first term of the loss function can be expressed as

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^{n} \left(\dot{X}_\ell(t_i) - \sum_{g=1}^{G} f_{g\ell}(X_g(t_i)) \right)^2 \\ = \frac{1}{n} (\dot{\mathbf{X}}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{\beta}_\ell)^T (\dot{\mathbf{X}}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{\beta}_\ell), \end{aligned} \tag{7}$$

where $\dot{\mathbf{X}}_\ell = (\dot{X}_\ell(t_1), \dot{X}_\ell(t_2), \dots, \dot{X}_\ell(t_n))^T$ is a length n column vector, $\boldsymbol{\Phi}_\ell = [\boldsymbol{\Phi}_{1\ell n}, \boldsymbol{\Phi}_{2\ell n}, \dots, \boldsymbol{\Phi}_{G\ell n}]^T$ is a $GK \times n$ matrix, $\boldsymbol{\Phi}_{g\ell n} = [\boldsymbol{\phi}_{g\ell}(X_g(t_1)), \boldsymbol{\phi}_{g\ell}(X_g(t_2)), \dots, \boldsymbol{\phi}_{g\ell}(X_g(t_n))]$ is a $K_{g\ell} \times n$ matrix, and $\boldsymbol{\phi}_{g\ell}(X_g(t_1)) = (\phi_{g\ell 1}(X_g(t_1)), \phi_{g\ell 2}(X_g(t_1)), \dots, \phi_{g\ell K_{g\ell}}(X_g(t_1)))^T$. Let $\mathbf{V}_{g\ell}$ be a $K_{g\ell} \times K_{g\ell}$ matrix with entries $v_{g,ij} = \int_0^T d_{g\ell i} d_{g\ell j} dx$, where $1 \leq i, j \leq K_{g\ell}$ and $d_{g\ell i}$ is expressed using (4).

Let $\mathbf{V}_\ell = \text{diag}(\mathbf{V}_{1\ell}, \mathbf{V}_{2\ell}, \dots, \mathbf{V}_{G\ell})$ be a matrix $(GK_{g\ell} \times GK_{g\ell})$ with blocks $\mathbf{V}_{1\ell}, \mathbf{V}_{2\ell}, \dots, \mathbf{V}_{G\ell}$ in its main diagonal and zeros elsewhere. Then the roughness penalty in (6) is transformed into the following form:

$$\gamma \sum_{g=1}^{G} \left|\left| \frac{df_{g\ell}^2(X_g(t))}{dt^2} \right|\right|^2 = \gamma \boldsymbol{\beta}_\ell^T \mathbf{V}_\ell \boldsymbol{\beta}_\ell. \tag{8}$$

Based on equation (3), the fSCAD penalty can be approximated as

$$\begin{aligned} \frac{M_{g\ell}}{\Delta_{x_g}} \int_{x_{gl}}^{x_{gu}} p_\lambda(|f_{g\ell}(X_g)|) dX_g \\ \approx \sum_{j=1}^{M_{g\ell}} p_\lambda \left(\sqrt{\frac{M_{g\ell}}{\Delta_{x_g}} \int_{x_{g,j-1}}^{x_{gj}} [f_{g\ell}(X_g)]^2 dX_g} \right). \end{aligned}$$

In addition, we define

$$||f_{g\ell[j]}||_2^2 \overset{def}{=} \int_{x_{g,j-1}}^{x_{gj}} [f_{g\ell}(X_g)]^2 dX_g = \boldsymbol{\beta}_{g\ell}^T \mathbf{M}_{g\ell j} \boldsymbol{\beta}_{g\ell},$$

where $\mathbf{M}_{g\ell j}$ is a $K_{g\ell} \times K_{g\ell}$ matrix with entries $m_{g\ell j,uv} = \int_{x_{g,j-1}}^{x_{gj}} \phi_{g\ell u}(X_g) \phi_{g\ell v}(X_g) dX_g$, if $j \leq u, v \leq j + d$ and zero otherwise. Using the local quadratic approximation (LQA) proposed in Fan and Li (2001), given some initial estimate $\boldsymbol{\beta}_{g\ell}^{(0)}$, we can derive that

$$\frac{M_{g\ell}}{\Delta_{x_g}} \int_{x_{gl}}^{x_{gu}} p_\lambda(|f_{g\ell}(X_g)|) dX_g \approx \boldsymbol{\beta}_{g\ell}^T \boldsymbol{W}_{g\ell}^{(0)} \boldsymbol{\beta}_{g\ell} + G(\boldsymbol{\beta}_{g\ell}^{(0)}),$$

where

$$\boldsymbol{W}_{g\ell}^{(0)} = \frac{1}{2} \sum_{j=1}^{M_{g\ell}} \left(\frac{\dot{p}_\lambda(||f_{g\ell[j]}||_2 \sqrt{M_{g\ell}/\Delta_{x_g}})}{||f_{g\ell[j]}||_2 \sqrt{\Delta_{x_g}/M_{g\ell}}} \mathbf{M}_{glj} \right),$$

and

$$\begin{aligned} G(\boldsymbol{\beta}_{g\ell}^{(0)}) \equiv \sum_{j=1}^{M_{g\ell}} p_\lambda \left(\frac{||f_{g\ell[j]}||_2}{\sqrt{\Delta_{x_g}/M_{g\ell}}} \right) \\ - \frac{1}{2} \sum_{j=1}^{M_{g\ell}} \dot{p}_\lambda \left(\frac{||f_{g\ell[j]}||_2}{\sqrt{\Delta_{x_g}/M_{g\ell}}} \right) \frac{||f_{g\ell[j]}||_2}{\sqrt{\Delta_{x_g}/M_{g\ell}}}. \end{aligned}$$

Adding all the fSCAD penalty for each gene, we have

$$\sum_{g=1}^{G} \frac{M_{g\ell}}{\Delta_{x_g}} \int_{x_{gl}}^{x_{gu}} p_\lambda(|f_{g\ell}(X_g)|) dX_g \approx \boldsymbol{\beta}_\ell^T \mathbf{W}_\ell^{(0)} \boldsymbol{\beta}_\ell + \sum_{g=1}^{G} G(\boldsymbol{\beta}_\ell^{(0)}), \tag{9}$$

where $\mathbf{W}_\ell^{(0)} = \text{diag}(\mathbf{W}_{1\ell}^{(0)}, \mathbf{W}_{2\ell}^{(0)}, \dots, \mathbf{W}_{G\ell}^{(0)})$. Putting (7), (8), and (9) together, we obtain

$$\begin{aligned} Q(\boldsymbol{\beta}_\ell) = \frac{1}{n} (\dot{\mathbf{X}}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{\beta}_\ell)^T (\dot{\mathbf{X}}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{\beta}_\ell) + \gamma \boldsymbol{\beta}_\ell^T \mathbf{V}_\ell \boldsymbol{\beta}_\ell \\ + \boldsymbol{\beta}_\ell^T \mathbf{W}_\ell^{(0)} \boldsymbol{\beta}_\ell + \sum_{g=1}^{G} G(\boldsymbol{\beta}_{g\ell}^{(0)}). \end{aligned}$$

By minimizing $Q(\boldsymbol{\beta}_\ell)$, we obtain the estimate for the basis coefficients

$$\widehat{\boldsymbol{\beta}}_\ell = \frac{1}{n}\left(\frac{1}{n}\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \gamma\mathbf{V}_\ell + \mathbf{W}_\ell^{(0)}\right)^{-1}\boldsymbol{\Phi}_\ell\dot{\mathbf{X}}_\ell.$$

Then, we can plug the estimate, $\widehat{\boldsymbol{\beta}}_\ell$, into (2) to obtain the estimates for all regulation functions:

$$\hat{f}_{g\ell}(X_g) = \phi_{g\ell}^T(X_g)\widehat{\beta}_{g\ell}, \quad g = 1, \ldots, G, \quad \ell = 1, \ldots, G.$$

2.5. *Identifiability Issue*

Modeling multiple regulatory genes introduces an identifiability problem. For instance, suppose there are only two regulatory genes such that the ODE (1) is reduced to

$$\dot{X}_\ell(t) = \mu_\ell + f_{1\ell}(X_1(t)) + f_{2\ell}(X_2(t)).$$

Since simultaneously adding any constant to $f_{1\ell}(\cdot)$ and subtracted it from $f_{2\ell}(\cdot)$ does not affect the model prediction, $f_{1\ell}(\cdot)$ and $f_{2\ell}(\cdot)$ are only estimable up to an additive constant.

To address this issue, we apply a similar strategy as in Wood (2006), which constrains the sum of $f_{g\ell}(\cdot)$ to zero over the entire time domain. That is,

$$E(f_{g\ell}(X_g(t))) = 0, g = 1, \ldots, G. \tag{10}$$

This also implies that $\hat{\mu}_\ell = E(\dot{X}_\ell(t))$. In the rest part of this section, we briefly discuss how to include constraints (10) into the parameter estimation.

Denote $w_{g\ell k} = \sum_{i=1}^n \phi_{g\ell k}(X_g(t_i))$. The constraint (10) can be satisfied in a sample as

$$\sum_{g=1}^G\left(\sum_{k=1}^{K_{g\ell}}\beta_{g\ell k}w_{g\ell k}\right)^2 = 0. \tag{11}$$

Next, we can recast the left side of (11) into a matrix form

$$\sum_{g=1}^G\left(\sum_{k=1}^{K_{g\ell}}\beta_{g\ell k}w_{g\ell k}\right)^2 = \boldsymbol{\beta}_\ell^T\boldsymbol{\Omega}_\ell\boldsymbol{\beta}_\ell,$$

where $\boldsymbol{\Omega}_\ell = \text{diag}(\boldsymbol{\Omega}_{1\ell}, \boldsymbol{\Omega}_{2\ell}, \ldots, \boldsymbol{\Omega}_{G\ell})$ and

$$\boldsymbol{\Omega}_{g\ell} = \begin{bmatrix} w_{g\ell 1}^2 & w_{g\ell 1}w_{g\ell 2} & \cdots & w_{g\ell 1}w_{g\ell K_{g\ell}} \\ w_{g\ell 2}w_{g\ell 1} & w_{g\ell 2}^2 & \cdots & w_{g\ell 2}w_{g\ell K_{g\ell}} \\ \vdots & \vdots & \cdots & \vdots \\ w_{g\ell K_{g\ell}}w_{g\ell 1} & w_{g\ell K_{g\ell}}w_{g\ell 2} & \cdots & w_{g\ell K_{g\ell}}^2 \end{bmatrix}.$$

We add $\lambda_I\boldsymbol{\beta}_\ell^T\boldsymbol{\Omega}_\ell\boldsymbol{\beta}_\ell$ to the loss function (6), in which λ_I is a relatively large positive number to make sure that (11) holds. Consequently, the estimator $\widehat{\boldsymbol{\beta}}_\ell$ becomes

$$\widehat{\boldsymbol{\beta}}_\ell = \frac{1}{n}\left(\frac{1}{n}\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \gamma\mathbf{V}_\ell + \mathbf{W}_\ell^{(0)} + \lambda_I\boldsymbol{\Omega}_\ell\right)^{-1}\boldsymbol{\Phi}_\ell\dot{\mathbf{X}}_\ell. \tag{12}$$

Note that $\boldsymbol{\Omega}_\ell$ is a singular matrix so that a very large value of λ_I might cause $\frac{1}{n}\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \gamma\mathbf{V}_\ell + \mathbf{W}_\ell^{(0)} + \lambda_I\boldsymbol{\Omega}_\ell$ in (12) to be almost singular. If that is the case, we recommend to try a new value of λ_I, for instance, half of the previous value.

Below we give the details of our algorithm to compute the estimated coefficients $\widehat{\boldsymbol{\beta}}_\ell$:

Step 1: Compute the initial estimate $\widehat{\boldsymbol{\beta}}_\ell^{(0)} = \frac{1}{n}(\frac{1}{n}\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \lambda_I\boldsymbol{\Omega}_\ell)^{-1}\boldsymbol{\Phi}_\ell\dot{\mathbf{X}}_\ell$.

Step 2: In each iteration, given $\widehat{\boldsymbol{\beta}}_\ell^{(i)}$, compute the corresponding $\boldsymbol{W}_\ell^{(i)}$. Then $\widehat{\boldsymbol{\beta}}_\ell^{(i+1)} = \frac{1}{n}(\frac{1}{n}\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \gamma\mathbf{V}_\ell + \boldsymbol{W}_\ell^{(i)} + \lambda_I\boldsymbol{\Omega}_\ell)^{-1}\boldsymbol{\Phi}_\ell\dot{\mathbf{X}}_\ell$. If a variable is very small in magnitude such that it makes $(\frac{1}{n}\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \gamma\mathbf{V}_\ell + \boldsymbol{W}_\ell^{(i)} + \lambda_I\boldsymbol{\Omega}_\ell)$ almost singular or badly scaled so that inverting $(\frac{1}{n}\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \gamma\mathbf{V}_\ell + \boldsymbol{W}_\ell^{(i)} + \lambda_I\boldsymbol{\Omega}_\ell)$ is unstable, then we manually shrink it into zero.

Step 3: Repeat Step 2 until $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\ell}}^{(i)}$ converges.

2.6. *Choose Tuning Parameters*

We need to specify four tuning parameters in (12): the total number of basis functions used to represent each regulation function, $K_{g\ell}$; the smoothing parameter in the roughness penalty for each regulation function, γ; the fSCAD penalty for sparsity, λ; and the identifiability parameter, λ_I.

First of all, a large value of $K_{g\ell}$ is chosen to obtain a good approximation for each regulation function $f_{g\ell}(\cdot)$. This will not result in a saturated model since the smoothing parameter, γ, and fSCAD penalty parameter, λ, will control the roughness of the regulation functions. Second, $\lambda_I \in [10^4, 10^9]$ generally works well according to our experience and this choice is not crucial. We note that the value of λ_I only affects the convergence speed. Once $K_{g\ell}$ and λ_I are determined, one can use a popular selection criterion such as information criterion (AICc, BIC) or cross validation to search the optimal values for γ and λ on a discrete grid. Our experience from the real data application suggests that the AICc information criterion tends to work well from a practical perspective.

2.7. *Derivative Estimation*

The ODE model in equation (1) uses the derivatives of each gene as the response. In this section, we introduce a smoothing spline method to estimate the derivative of each gene based on the its own observed expression values. Other methods for the derivative estimation can also be used in our framework.

Let Y_i denote the measurement for a particular gene at time t_i, $t_i \in [0, T]$. Suppose that $Y_i, i = 1, \ldots, n$, is from an unknown gene expression function $X(t)$. That is,

$$Y_i = X(t_i) + \epsilon_i, i = 1, \ldots, n,$$

where ϵ_i is independently and identically distributed from a normal distribution $N(0, \sigma_s^2)$. Our goal is to estimate $X(t)$ and $\dot{X}(t)$ from $Y_i,\ i = 1, \ldots, n$.

We first represent $X(t)$ using a linear combination of B-spline basis functions:

$$X(t) = \sum_{j=1}^{J} \theta_i \psi_j(t) = \boldsymbol{\psi}(t)^T \boldsymbol{\theta},$$

in which $\boldsymbol{\theta}$ is the length J vector of coefficients, and $\boldsymbol{\psi}(t)$ is the length J vector of basis functions. Then, we estimate the vector of coefficients $\boldsymbol{\theta}$ by minimizing the following loss function:

$$Q_0(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(Y_i - X(t_i) \right)^2 + \lambda_0 \int \left[\ddot{X}(t) \right]^2 dt, \ \lambda_0 > 0. \quad (13)$$

Intuitively, the first term in $Q_0(\boldsymbol{\theta})$ quantifies the goodness of fit to the data, and the second one controls the roughness of the estimated function. The relative importance between these two terms is controlled by λ_0. For instance, a larger value of λ_0 will lead to a smoother estimate for $X(t)$. Here, we suggest using the generalized cross validation (GCV) score in Wahba and Craven (1978) to determine the value of λ_0.

To estimate the vector of the basis coefficients $\boldsymbol{\theta}$, we can rewrite (13) into a matrix form:

$$Q_0(\boldsymbol{\theta}) = (\mathbf{Y} - \boldsymbol{\Psi}\boldsymbol{\theta})^T (\mathbf{Y} - \boldsymbol{\Psi}\boldsymbol{\theta}) + \lambda_0 \boldsymbol{\theta}^T \mathbf{R} \boldsymbol{\theta},$$

where $\mathbf{R}$ is a $J \times J$ matrix with entries $R_{ij} = \int \ddot{\psi}_i(t) \ddot{\psi}_j(t) dt$ and $\boldsymbol{\Psi}$ is an $n \times J$ matrix with entries $\Psi_{ij} = \psi_j(t_i)$. Taking the derivative $Q_0(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, one can obtain

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{\Psi}^T \boldsymbol{\Psi} + \lambda_0 \mathbf{R})^{-1} \boldsymbol{\Psi}^T \mathbf{Y}.$$

Thus, the estimated trajectory for $X(t)$ and the derivative $\dot{X}(t)$ can be expressed as $\hat{X}(t) = \boldsymbol{\psi}(t)^T \hat{\boldsymbol{\theta}}$ and $\hat{\dot{X}}(t) = \dot{\boldsymbol{\psi}}(t)^T \hat{\boldsymbol{\theta}}$.

Because the estimated derivatives for gene ℓ at observed time points are essentially correlated across time, equation (7) should take this correlation into consideration and be replaced by

$$\frac{1}{n} (\hat{\dot{\mathbf{X}}}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{\beta}_\ell)^T \left[\widehat{\mathrm{Cov}}(\hat{\dot{\mathbf{X}}}) \right]^{-1} (\hat{\dot{\mathbf{X}}}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{\beta}_\ell),$$

where the estimated variance-covariance matrix of the derivatives $\widehat{\mathrm{Cov}}(\hat{\dot{\mathbf{X}}})$ can be obtained with the delta method,

$$\widehat{\mathrm{Cov}}(\hat{\dot{\mathbf{X}}}) = \dot{\boldsymbol{\Psi}}^T \widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}) \dot{\boldsymbol{\Psi}} = \hat{\sigma}_s^2 \dot{\boldsymbol{\Psi}}^T (\boldsymbol{\Psi}^T \boldsymbol{\Psi} + \lambda_0 \mathbf{R})^{-1} \boldsymbol{\Psi}^T \boldsymbol{\Psi} (\boldsymbol{\Psi}^T \boldsymbol{\Psi} + \lambda_0 \mathbf{R})^{-1} \dot{\boldsymbol{\Psi}}, \quad (14)$$

in which $\hat{\dot{\mathbf{X}}} = (\hat{\dot{X}}(t_1), \ldots, \hat{\dot{X}}(t_n))^T$, $\dot{\boldsymbol{\Psi}}$ is a $n \times J$ matrix with entries $\dot{\psi}_j(t_i)$ and $\hat{\sigma}_s^2$ can be obtained by computing the sample variance of the residuals $\mathbf{e}_s = \mathbf{Y} - \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}$.

In fact, as one reviewer suggests, our proposed algorithm given at the end of Section 2.5 is still applicable by simply letting $\left[\widehat{\mathrm{Cov}}(\hat{\dot{\mathbf{X}}}) \right]^{-1} = \mathbf{L}_\ell^T \mathbf{L}_\ell$ be the Cholesky decomposition of the inverse variance-covariance matrix and then pre-conditioning both $\hat{\dot{\mathbf{X}}}_\ell$ and $\boldsymbol{\Phi}_\ell^T$ with $\mathbf{L}_\ell$. Consequently, equation (12) becomes

$$\widehat{\boldsymbol{\beta}}_\ell = \frac{1}{n} \left(\frac{1}{n} \boldsymbol{\Phi}_\ell \mathbf{L}_\ell^T \mathbf{L}_\ell \boldsymbol{\Phi}_\ell^T + \gamma \mathbf{V}_\ell + \mathbf{W}_\ell^{(0)} + \lambda_I \boldsymbol{\Omega}_\ell \right)^{-1} \boldsymbol{\Phi}_\ell \mathbf{L}_\ell^T \mathbf{L}_\ell \hat{\dot{\mathbf{X}}}_\ell.$$

3. Application

We consider a data set of 20 Drosophila melanogaster genes involved in the muscle development during the embryonic stage (see Bar-Joseph (2004) for details). The time-course gene expressions are measured at 30 time points in the embryonic stage (Arbeitman et al., 2002).

The time-varying directed GRN of these 20 genes are modeled using the nonlinear ODE model (1). The time-varying regulation functions $f_{g\ell}(X_g)$ in equation (1) for each of those 20 genes are estimated in two steps. In the first step, we obtain the estimate for the trajectory of each gene and its derivatives using the smoothing spline method, as introduced in Section 2.7. In the second step, we treat the derivative estimates for each gene as the response and all genes' trajectory estimates as the covariates in ODE model (1). We then estimate the basis coefficients for each regulation function via (12). The smoothing parameter γ and the sparsity parameter λ are both determined simultaneously using AICc criterion. The smoothing parameter γ is chosen from four candidate values: 10^{-5}, 10^{-3}, 10^{-1}, and 10. The sparsity parameter λ is selected from five candidate values: $10^{-2}, 10^{-1}$, 1, and 10. Since the results are not sensitive to specific values of the number of basis functions, $K_{g\ell}$, and the identifiability parameter, λ_I, we set their values to be $K_{g\ell} = 5$ and $\lambda_I = 10^4$ to ease the computation.

Figure 2 shows the estimated regulation functions for gene *Myo31DF*. It can be seen that 3 out of 20 genes are selected, which means that the regulation functions of the other 17 genes are estimated to be strictly zero during the entire embryonic stage. Those three estimated regulation functions shown in Figure 2, all have non-linear trends and show local sparsity to some extent.

We compare the prediction performance of our proposed method with the group Lasso method in the real data application. To be more specific, we remove the last observation for all genes in the network and estimate the regulation functions for the target gene *Myo31DF* using the remaining observations only. Then, we use our method and the group Lasso method to estimate the gene regulation functions in the ODE model (1). We then use the estimated ODE model (1) to predict the expression of the target gene at the last time point. We also compare their prediction performances with two other methods: the constant expression method and an autoregressive model, AR1. The constant expression model simply takes the sample mean from previously observed trajectories values of *Myo31DF* as the prediction value. The AR1 method is fitted using the maximum likelihood approach. The detailed results, presented in Web Table S1 in the supplementary file, show that the locally sparse method has the most accurate prediction among all methods.

To check whether our finding for gene *Myo31DF* makes biology sense, we conduct a literature search for studies on gene interactions using the Drosophila Interactions Database

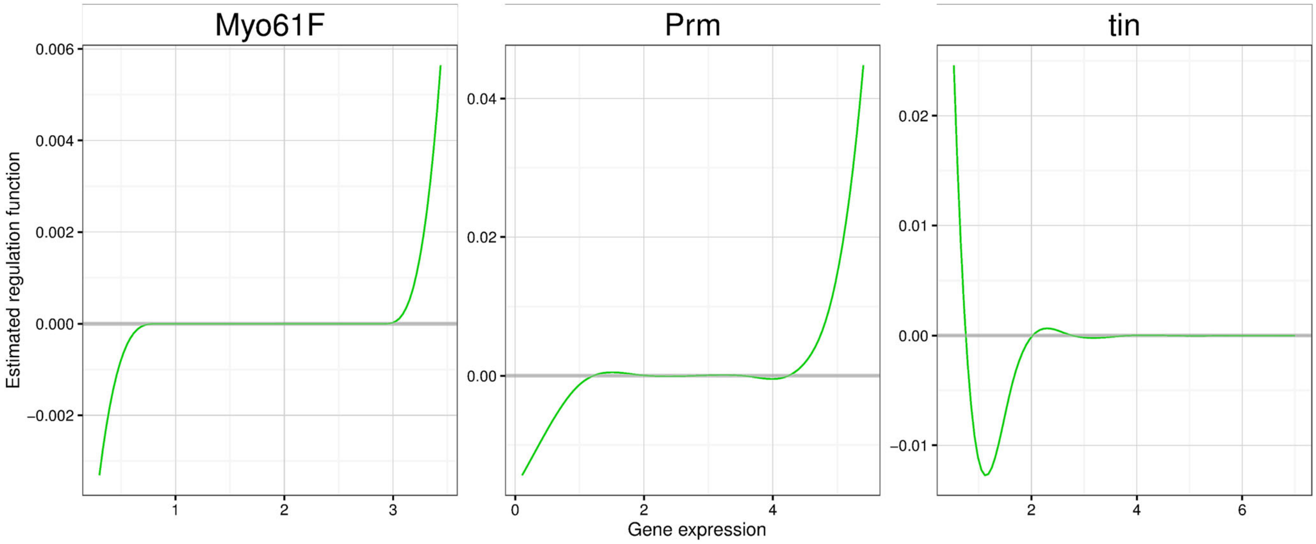

Figure 2. Estimated regulation functions on gene *Myo31DF* based in the ODE model (1). Three regulatory genes, that is, *Prm*, *tin*, and *Myo61DF* are selected out of 20 genes. All the regulation functions of the rest 17 genes are estimated to be strictly zero during the whole embryonic stage. This figure appears in color in the electronic version of this article.

(Murali et al., 2011) and the GeneMANIA tool (Warde-Farley et al., 2010). We find evidences in the literature about all three regulatory genes *Myo61F*, *Prm*, and *tin* on gene *Myo31DF*. For instance, Hozumi et al. (2006) suggested that both *Myo61F* and *Myo31DF* played a crucial role in generating left-right asymmetry of the embryonic gut. They found that *Myo31DF* was required in the hindgut epithelium for normal embryonic handedness and the overexpression of *Myo61F* reversed the handedness of the embryonic gut, and its knockdown also caused a left-right patterning defect. These two unconventional myosin I proteins might have antagonistic functions in left-right patterning. The results obtained from our analysis match these insights. For instance, Figure 2 shows that gene *Myo61F* only regulates gene *Myo31DF* when its expression level is either less than 1 or greater than 2. Thus, either the knockdown or overexpression of *Myo61F* will cause a left-right patterning defect. In addition, Lewis, Burge, and Bartel (2005), Ruby et al. (2007), Ruby, Jan, and Bartel (2007) and Kheradpour et al. (2007) suggested that gene *Prm* and gene *Myo31DF* shared two common miRNAs, that is, *mir-iab-4* and *mir-999*. As for gene *tin*, even though there was no direct evidence showing its regulation effect on *Myo31DF*, Fu et al. (1997) found out that *tin* was critical in determining the patterning of the Drosophila heart. Because of gene *Myo31DF*'s role in generating the left-right asymmetry gut, our hypothesis is that *tin* regulates *Myo31DF* to insure the left-right asymmetry formation in the heart. This hypothesis needs to be further investigated in real genetic studies.

Once the regulation functions for all 20 genes are estimated, we can visualize the whole GRN at any given time point. Figure 3 shows the estimated GRN at different selected time points during the embryonic stage. One important feature of the estimated GRN is that the regulation effects between genes are time-varying. For example, *Prm* regulates *sls* at the beginning of the embryonic stage, that is, $t = 3h$, however, *Mef2* replace *Prm*'s role in regulating *sls* in the middle stage. In addition, from the whole network point of view, we observe that genes interact with each other more frequently in the beginning than in the middle or at the end of the embryonic stage. Finally, we find some strong regulators such as *Mef2*, *Myo61F*, *Prm*, and *Mhc*, which act as hubs in our estimated GRN.

In Figure 3, we highlight those interactions that have been verified in the literature. Details of the references for each interaction are provided in the supplementary materials. A solid line indicates the corresponding directed regulation effect between genes has been verified; a dashed line means the corresponding gene-to-gene interaction has been discovered before but the exactly direction is unclear; and a dotted line means the corresponding interaction has not been found so far. Most regulation effects estimated using our method have been verified previously. Those regulation effects that have not been discovered may be candidate hypotheses for future investigation. It is worth mentioning that the total number of known interactions in the literature is 158 out of 400 possible interactions. In other words, the background interaction rate is 39.5%(=158/400). Using our method, wIt is woe estimate 67 interactions, 58 of which are verified in the literature. The discovery rate for our method is 86.6%(=58/67), which is more than twice the background interaction rate.

Another very important feature of our estimated GRN shown in Figure 3 is that the estimated network is sparsely connected. In other words, only a limited number of genes regulate a target gene. Table 1 displays a complete list of estimated regulatory genes for all genes. The number of regulatory genes ranges from 2 to 6 with an average 3.35. Furthermore, we prioritize those selected regulatory genes based on their estimated signal strength. The signal strength is defined using the functional L_2 norm of the estimated regulation functions in the entire time domain considered. For example, for gene *Actn*, *Mef2* is a stronger regulator in the

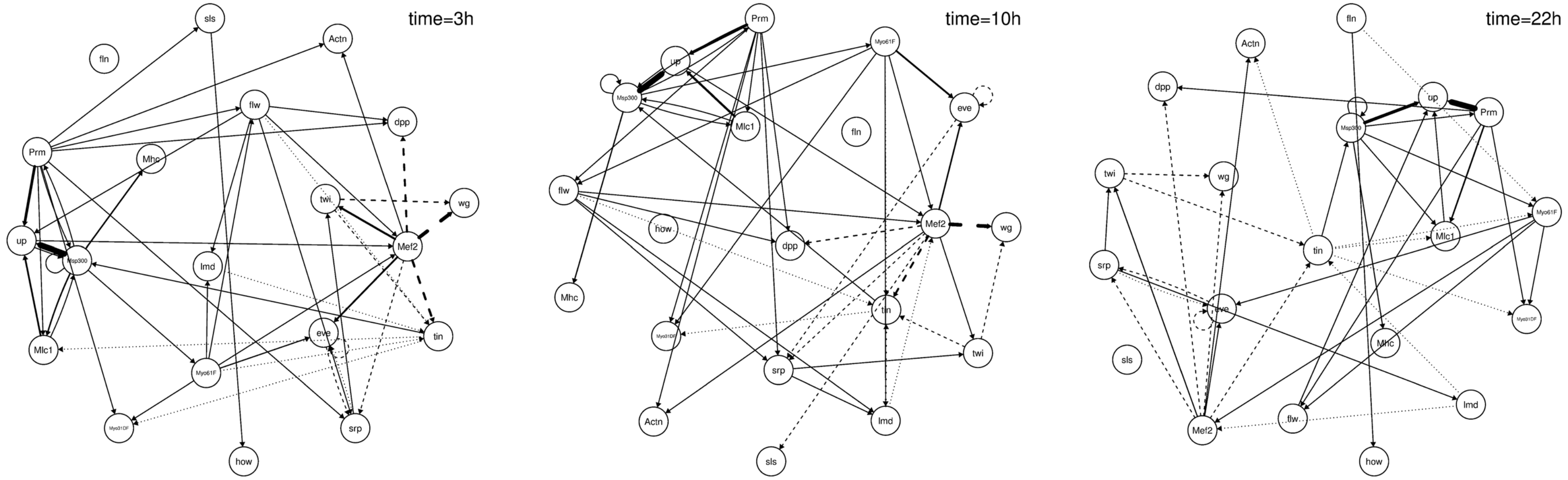

Figure 3. The estimated time-varying GRN of 20 genes in the muscle development pathway at three time points during the embryonic stage. The connection lines represent the existence of regulation effects between genes. The line thickness corresponds to the magnitude of the regulation function. The line type indicates whether the regulations have been verified in the literature: solid (verified regulation effects), dashed (verified gene-to-gene interactions), and dotted (unverified regulation effects). Detailed reference can be found with this article at the *Biometrics* website on Wiley Online Library. This figure is generated using the qgraph package (Epskamp et al., 2012).

Table 1
The regulatory genes for all 20 genes selected by our method. The regulatory genes are sorted by their overall regulation effect on the corresponding target gene. For example, Mef2 *has the largest the overall regulation effect on* Actn *in comparison with* Prm *and* tin*.*

Target Gene	Regulatory Genes
Actn	Mef2 Prm tin
dpp	Mef2 Prm flw
eve	Myo61F Mef2 srp eve
fln	tin Prm Actn Mhc Msp300 flw
flw	Myo61F Prm
how	sls Mlc1 fln
lmd	srp Myo61F flw
Mef2	Myo61F up flw lmd
Mhc	Msp300
Mlc1	Msp300 Prm tin
Msp300	tin up Mlc1 Prm Msp300
Myo31DF	Prm tin Myo61F
Myo61F	Msp300 tin fln
Prm	Msp300
sls	Prm tin how Mef2
srp	Mef2 Prm flw eve twi Msp300
tin	Mef2 lmd Myo61F flw twi
twi	Mef2 srp
up	Msp300 Prm Mlc1 flw
wg	Mef2 twi

whole time interval compared to *Prm* and *tin*, as shown in the first row of Table 1.

4. Simulation

In this section, we assess the performance of our fSCAD method using a simulation study. To mimic the real gene regulation process, we use the ODE model for the target gene *Myo31DF* estimated from the real data analysis to generate the true trajectory of the target gene as follows:

$$X_0(t) = \int_0^t \dot{X}_0(\tau)d\tau = \int_0^t \sum_{i=1}^{20} f_i(X_i(\tau))d\tau, \qquad (15)$$

where $\dot{X}_0(\tau)$ denotes the derivative of the expression for the target gene and $X_i(\tau)$ is the expression function of gene i at time τ, that is, what was observed empirically at τ. Here, we take $\tau \in \{0, 1, 2, \ldots, 23\}$. The three true regulation functions $f_i(X_i), i = 1, 2, 3$ are the same as the estimated regulation functions from the real data shown in Figure 2, and all the remaining 17 true regulation functions are strictly zero in the whole interval. That is, $f_i(X_i) \equiv 0, i = 4, \ldots, 20$. For simplicity, we use X_1, X_2, and X_3 to denote gene *Prm*, *tin*, and *Myo61F*, respectively. In addition, we refer to genes with nonzero regulation functions as regulatory genes and genes with strictly-zero regulation functions as non-regulatory genes. To account for the estimation error in estimating the derivative function $\dot{X}_0(t)$ in the first step, as one reviewer suggests, we generate the noisy data by adding a white noise ϵ to the true $X_0(t)$. The noise level is controlled by the noise-to-signal ratio ρ as $\epsilon \overset{i.i.d}{\sim} N(0, \rho\sigma_x^2)$, where σ_x is the sample standard deviation of the true trajectory $X_0(t)$ empirically observed at $\tau \in \{0, 1, 2, \ldots, 23\}$.

We estimate ODE model (1) using the group Lasso method and the following three methods:

Locally sparse method: the loss function defined in equation (7) with both fSCAD penalty and roughness penalty;

Smoothing spline method: the loss function defined in equation (7) with roughness penalty only, that is, $\lambda = 0$;

Linear fSCAD method: the loss function defined in equation (7) with the fSCAD penalty and a very large roughness penalty. More specifically, we fix $\gamma = 100$ to force the estimated regulation functions to be almost linear.

For the locally sparse method and the smoothing spline method, the smoothing parameter γ is chosen from four candidate values: $10, 10^{-1}, 10^{-3}$, and 10^{-5} using AICc. For both the locally sparse method and the linear fSCAD method, the sparsity parameter λ is selected from five candidate values: $10, 1, 10^{-1}$, and 10^{-2} using AICc. In addition, the number of basis function $K_{g\ell}$ and the identifiability parameter λ_I remain the same as in the real data analysis, that is, $K_{g\ell} = 5$ and $\lambda_I = 10^4$. For the group Lasso method, we use the 5-fold cross-validation to choose the penalty parameter.

We access the variable-selection accuracy for each method using the false negative error (FN) and the false positive error (FP), which are defined in the gene regulation scenario as follows:

$$\text{FN} = \frac{\text{\# of incorrectly estimated non-regulatory genes}}{\text{\# of all true regulatory genes}},$$

$$\text{FP} = \frac{\text{\# of incorrectly estimated regulatory gene}}{\text{\# of all estimated regulatory gene}}.$$

The simulation is repeated for 100 times and the results are presented in Table 2. First of all, we can see that the locally sparse method yields the lowest FN error among all the methods given the same noise-to-signal level. To be more specific, when the noise-to-signal level is only 1%, the locally sparse method only misselects 25% of all the estimated regulatory

Table 2
The means and standard deviations (SD) of the false positive errors (FP) and the false negative errors (FN) of the four methods in 100 simulation replicates. Here ρ represents the noise-to-signal ratio in the simulated data.

Method	ρ (%)	FP Mean (%)	FP SD (%)	FN Mean (%)	FN SD (%)
Locally Sparse	1	25.0	0.0	0.0	0.0
	5	30.0	11.5	9.0	15.6
Smoothing Spline	1	85.0	0.0	0.0	0.0
	5	85.0	0.0	0.0	0.0
Linear fSCAD	1	63.9	5.5	4.8	11.7
	5	63.9	5.5	4.8	11.7
Group Lasso	1	94.9	8.2	88.1	19.5
	5	99.5	3.2	99.3	4.7

genes. In comparison, with no sparsity penalty, smoothing spline method is not able to produce a parsimony model such that the FP error keeps at 85% even when the noise-to-signal level is only 1%. The group Lasso method also fails to detect the true regulatory genes and over 90% of the estimated regulatory genes are actually non-regulatory genes in our simulation settings. This is because the group Lasso method cannot detect local sparsity and always penalize the regulation function in the whole domain. In addition, the linear fSCAD yields the second lowest FP error because of the fSCAD penalty, however, as the large roughness penalty forces the regulation functions to be linear, the resulting model are not as parsimony as the true model is. On the other hand, the smoothing spline method does not make any FN error due to the fact that it estimates all the genes as regulatory genes. Both the linear fSCAD and the locally sparse method yield similar FN errors, which indicates that they seldom select true non-regulatory genes as regulatory genes. Lastly, the performance of group Lasso is still very poor, because over 85% of the estimated non-regulatory genes are actually true regulatory genes even when the noise-to-signal ratio level is 1%.

Next, we assess the each method's ability to detect the sparsity of the regulation functions. For each regulation function, we divide the corresponding entire interval into 100 subintervals equally. The false positive rate for each estimated regulation function in one simulation run is calculated as the percentage of those strictly-zero subintervals which are falsely estimated as nonzero. Then, we take the average false positive rate for each method across all regulation functions in 100 simulation replicates. The complete results are shown in Web Table S2 of the supplementary file. We find that the locally sparse method yields the lowest false positive rate among all methods considered. Less than 10% of the true strictly-zero subintervals are incorrectly estimated as nonzero even when the noise-to-signal ratio is high. The smoothing spline method cannot produce sparse regulation function estimations, therefore its false positive rate is always 1. With a very large roughness penalty, the linear fSCAD method forces the estimated regulation function to be close to linear forms, and this method fails to detect the change points between zero regions and nonzero regions. Therefore, the linear fSCAD method yields the second highest false positive rate among all four methods. With the group Lasso penalty, the group Lasso method tends to shrink the entire regulation function to zero and the corresponding false positive rate is about 16%. On the other hand, we find that the the group Lasso method always shrinks those three true regulation function into strictly zero even when the noise-to-ratio level is only 1%. In contrast, the locally sparse method estimated regulation function are much closer to the the true regulation functions. The average of estimated regulation functions compared to the true regulation functions along with the experimental point-wise confidence bands using the locally sparse method are shown in Web Figures S3 and S4 in the supplementary file.

We also compare the prediction performance of our proposed method with the group Lasso method. More specifically, we hold out the last observation, that is, $X_i(23), i = 0, 1, \ldots, 20$, for all the genes in the network and estimate the regulation functions using the first 23 observations only. Then, we use our method and the group Lasso method to estimate the gene regulation functions in the ODE model (1) in the main manuscript. We then use the estimated ODE model (1) to predict the value of $X_0(\tau)$ at $\tau = 23$ and compute the squared prediction error. We also compare their prediction performances with two other methods: the constant expression method and an autoregressive model, AR1. The constant expression model simply takes the sample mean from previously observed trajectories values as the prediction value. The AR1 method is fitted using the maximum likelihood approach. Web Table S3 shows that the locally sparse method yields the lowest mean squared prediction error among all methods, which is only about 10% compared to the group Lasso method and the AR1 model.

In summary, our proposed method can correctly select the true regulatory genes without misselecting those true non-regulatory genes in the ODE model compared to other alternative methods. In addition, it can also successfully identify the strictly-zero subregions of all regulation functions. Finally, it outperforms popular method such as group Lasso in term of the forward prediction accuracy.

5. Conclusions

ODE models are widely used to model a dynamical system in many fields such as biology, economics, and physics. In this article, we use a high-dimensional nonlinear ODE model to describe a time-varying direct GRN. It is worth mentioning, as one reviewer suggests, the ODE model itself is time-stationary in the sense that all the regulation functions are deterministic functions of the regulatory gene expressions, but the edges may implicitly emerge or disappear over time, and the strength of the edge may vary with time, because the expressions of regulatory genes change with time. We propose the fSCAD method to estimate the unknown regulation functions in the high-dimensional ODE model from the time-course gene expression data.

In the real data application, we show that our method can simultaneously detect the significant regulatory genes, estimate the nonlinear regulation functions without any parametric assumption, and identify the intervals with no regulation effects. The resulting GRN with the estimated regulation functions has many potential implications. First, based on the estimated edges and their corresponding directions, new hypotheses for gene regulation mechanism can be proposed as candidate relationships for future investigations. For those edges that have already been verified in the literature, we can prioritize them based on the estimated signal strength. In addition, when no prior knowledge about the direction of the regulation effect is available, our method can be a good starting point for the direction detection. Furthermore, our method can not only suggest potentially unverified regulation relationships between genes, but also give clues in which time periods the regulation effects are most likely to be detected. This advantage can greatly facilitate the future biology experiment designs for detecting gene regulation effects.

Furthermore, our simulation study shows that our method is able to estimate the true regulation functions under different levels of noises in the data more accurately in comparison with the group Lasso method. Finally, our method avoids solving the ODEs numerically, making it computational efficient and feasible in the high-dimensional context. Our method can be extended to model and estimate other high-dimensional directed networks from time-course or longitudinal data.

6. Supplementary Materials

Web Figures, Tables, and the reference details for gene interactions referenced in Sections 3 and 4 are available at the *Biometrics* website on Wiley Online Library. The R code is available at https://github.com/YunlongNie/flyfuns.

Acknowledgements

The authors are grateful for the invaluable comments and suggestions from the editor, Dr Yi-Hau Chen, an associate editor, and two reviewers. The authors also thank Prof Eric P. Xing and Prof. Le Song for kindly providing us the data and their computing codes. This research was supported by Nie's Postgraduate Scholarship-Doctorial (PGS-D) from the Natural Sciences and Engineering Research Council of Canada (NSERC), and the NSERC Discovery grants of Wang and Cao.

References

Arbeitman, M. N., Furlong, E. E., Imam, F., Johnson, E., Null, B. H., Baker, B. S., et al. (2002). Gene expression during the life cycle of drosophila melanogaster. *Science* **297**, 2270–2275.

Bar-Joseph, Z. (2004). Analyzing time series gene expression data. *Bioinformatics* **20**, 2493–2503.

Cao, J. and Zhao, H. (2008). Estimating dynamic models for gene regulation networks. *Bioinformatics* **24**, 1619–1624.

Chen, J. and Wu, H. (2008). Estimation of time-varying parameters in deterministic dynamic models. *Statistica Sinica* **18**, 987–1006.

de Boor, C. (2001). *A Practical Guide to Splines*. Applied Mathematical Sciences. New York: Springer.

Epskamp, S., Cramer, A. O. J., Waldorp, L. J., Schmittmann, V. D., and Borsboom, D. (2012). qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software* **48**, 1–18.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**, 1348–1360.

Fan, J. and Peng, H. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *Annals of Statistics* **32**, 928–961.

Fu, Y., Ruiz-Lozano, P., and Evans, S. M. (1997). A rat homeobox gene, rnkx-2.5, is a homologue of the tinman gene in drosophila and is mainly expressed during heart development. *Development Genes and Evolution* **207**, 352–358.

Hanneke, S., Fu, W., and Xing, E. P. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics* **4**, 585–605.

Hozumi, S., Maeda, R., Taniguchi, K., Kanai, M., Shirakabe, S., Sasamura, T., etal. (2006). An unconventional myosin in drosophila reverses the default handedness in visceral organs. *Nature* **440**, 798–802.

Jensen, F. V. (1996). *An introduction to Bayesian networks*, volume 210. London: UCL Press.

Kheradpour, P., Stark, A., Roy, S., and Kellis, M. (2007). Reliable prediction of regulator targets using 12 drosophila genomes. *Genome research* **17**, 1919–1931.

Kolar, M., Song, L., Ahmed, A., and Xing, E. P. (2010). Estimating time-varying networks. *Annals of Applied Statistics* **4**, 94–123.

Kolar, M. and Xing, E. P. (2009). Sparsistent estimation of time-varying discrete markov random fields. *arXiv:0907.2337* .

Laubenbacher, R. and Stigler, B. (2004). A computational algebra approach to the reverse engineering of gene regulatory networks. *Journal of Theoretical Biology* **229**, 523–537.

Lewis, B., Burge, C., and Bartel, D. (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets. *Cell* **120**, 15.

Lin, Z., Cao, J., Wang, L., and Wang, H. (2016). Locally sparse estimator for functional linear regression models. *Journal of Computational and Graphical Statistics* doi:10.1080/10618600.2016.1195273, 1–41.

Lu, T., Liang, H., Li, H., and Wu, H. (2011). High-dimensional odes coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *Journal of the American Statistical Association* **106**, 1242–1258.

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A., and Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature* **431**, 308–312.

Mehra, S., Hu, W.-S., and Karypis, G. (2004). A boolean algorithm for reconstructing the structure of regulatory networks. *Metabolic Engineering* **6**, 326–339.

Murali, T., Pacifico, S., Yu, J., Guest, S., Roberts, G. G., and Finley, R. L. (2011). Droid 2011: A comprehensive, integrated resource for protein, transcription factor, rna and gene interactions for drosophila. *Nucleic Acids Research* **39**, D736–D743.

Needham, C. J., Bradford, J. R., Bulpitt, A. J., and Westhead, D. R. (2007). A primer on learning in bayesian networks for computational biology. *PLoS Computational Biology* **3**, 129.

Ramsay, J. O. and Silverman, B. W. (2002). *Applied functional data analysis: Methods and case studies*, volume 77. New York: Springer.

Ruby, J. G., Jan, C. H., and Bartel, D. P. (2007). Intronic microrna precursors that bypass drosha processing. *Nature* **448**, 83–86.

Ruby, J. G., Stark, A., Johnston, W. K., Kellis, M., Bartel, D. P., and Lai, E. C. (2007). Evolution, biogenesis, expression, and target predictions of a substantially expanded set of drosophila micrornas. *Genome Research* **17**, 1850–1864.

Song, L., Kolar, M., and Xing, E. P. (2009). Keller: Estimating time-varying interactions between genes. *Bioinformatics* **25**, i128–i136.

Steuer, R., Kurths, J., Daub, C. O., Weise, J., and Selbig, J. (2002). The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics* **18**, S231–S240.

Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A Gene-Coexpression Network For Global Discovery Of Conserved Genetic Modules. *Science* **302**, 249–255.

Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology* **42**, 563–585.

Wahba, G. and Craven, P. (1978). Smoothing noisy data with spline functions. estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik* **31**, 377–404.

Warde-Farley, D., Donaldson, S. L., Comes, O., Zuberi, K., Badrawi, R., Chao, P., et al. (2010). The genemania prediction server: Biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Research* **38**, W214–W220.

Wood, S. (2006). *Generalized additive models: An introduction with R*. London: Chapman and Hall/CRC.

Wu, H., Lu, T., Xue, H., and Liang, H. (2014). Sparse additive ordinary differential equations for dynamic gene regulatory network modeling. *Journal of the American Statistical Association* **109**, 700–716.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**, 49–67.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* **101**, 1418–1429.

Received March 2016. *Revised February* 2017.
Accepted February 2017.