Estimating Time-Varying Directed Gene Regulation Networks : Paper Review

Ha Yeon Won

1 Introduction

In biology and genetics, the problem of modeling gene regulation networks (GRN) has been of a great interest for a long time. To deal with this problem, a variety of methods have been suggested 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 deal with static GRN, which is time-invariant topology. Moreover, a sparse model is suitable to model the GRN which have the large number of genes in the network. However, in high dimension circumstance, where the total number of genes available far exceeds the number of gene expression measures, model selection has not been well addressed.

In this article, authors suggested 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. Using this high dimensioanl nonlinear ODE model, they made two contributions. First is modeling the dynamical feature of directed GRN, 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. The other is they proposed functional smoothly clipped absolute deviation (fSCAD) method th 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.

2 Preliminaries

2.1 Preliminary: Smoothly clipped absolute deviation (SCAD)

The smoothly clipped absolute deviation (SCAD) is developed to improve the model variable selection performance in high dimension context. Pan and Li (2001) suggested the SCAD estimator.

Consider the linear regression model

$$\mathbf{y} = \mathbf{X} + \epsilon$$

where $\mathbf{y}$ is an $nx1$ vector and $\mathbf{X}$ is an nxd matrix.

A form of the penalized least squares is

$$\frac{1}{2} \parallel \mathbf{y} - \mathbf{X}\beta \parallel^2 + \lambda \sum_{j=1} dp_j(|_j|) = \frac{1}{2} \parallel \mathbf{y} - \hat{\mathbf{y}} \parallel^2 + \frac{1}{2} \sum_{j=1} d(z_j - \beta_j)^2 + \lambda \sum_{j=1} dp_j(|\beta_j|)$$

The penalty functions $p_j(\cdot)$ are not necessarily the same for all j. for simplicity of presentation, they assume that the penalty functions for all coefficients are the same, denoted by $p(|\cdot|)$. Furthermore, they denote $\lambda p(|\cdot|)$ by $p_\lambda(|\cdot|)$.

The minimization problem of equation (1) is equivalent to the following equation.

$$\frac{1}{2}(z-\theta)^2 + p_\lambda(|\theta|)$$

$$p_\lambda(|\theta|) = \lambda^2 - (|\theta| - \lambda)^2 I(|\theta| < \lambda)$$

A good penalty function should result in an estimator with three properties.

- Unbiaedness: The resulting estimator is nearly unbiased when the true unknown parameter is large to avoid unnecessary modeling bias.
- Sparsity: The resulting estimator is a thresholding rule, which automatically sets small estimated coefficients to zero to reduce model complexity.
- Continuity: The resulting estimator is contiuous in data z to avoid instability in model prediction.

The continuous differentiable penalty function defined by

$$p'_\lambda(\theta) = \lambda \left\{ I(\theta \le \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} + I(\theta > \lambda) \right\}, \; a > 2, \; \theta > 0$$

This penalty function leaves large values of θ not excessively penalized and makes the solution continuous. The resulting solution (Fan, 1997) is given by

$$\hat{\theta} = \begin{cases} sgn(z)(|z| - \lambda)_+, & \text{when } |z| \le 2\lambda, \\ (a-1)z - sgn(z)a\lambda/(a-2), & \text{when } 2\lambda < |z| \le a\lambda, \\ z, & \text{when } |z| > a\lambda \end{cases}$$

2.2 Quadratic Approximation

Quadratic approximation is an extension of linear approximation, which is add the second derivative term to the formula of linear approximation.

Definition 1. The linear approximation of a function $f(x)$ around a value $x = c$ is the following linear function. c is a constant that you have chosen, so this is just a function of x.

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \;\; (x \approx x_0)$$

Definition 2. The quadratic approximation of a function $f(x)$ for values of x near x_0 is:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 \ \ (x \approx x_0)$$

These are more complicated and so are only used when higher accuarcy is needed.

3 Method

3.1 An ODE model for Time-Varying Directed Gene Regulation Networks

Ordinary differential eqauation (ODE) is an equation for an unknown function that contains not only the function but also its derivatives. As the derivative of a function provides the rate at which that function is changing with respect to its independent variable, the equations describing these phenomena often involve one or more derivatives, and they refer to them as differential equations.

Suppose a time-varying directed GRN has G genes in total, and their expressions are measured in a certain time period.

$$\dot{X}_l(t) = \mu_l + \sum_{g=1}^{G} f_{gl}(X_g),\ l = 1, ..., G,\ \ t \in [0, T] \tag{1}$$

where $\dot{X}_l(t)$ denotes the first derivative of $X_l(t)$ at time t for the targe gene l, $\mu_l$is the intercept term, and $f_{gl}(X_g)$ represents the regulation function of gene g on gene l. Here, assume $\dot{X}_l(t)$ is known, and in Section 2.7, how to estimate it will be discussed.

In the ODE model (1), only a few regulation functions $f_{gl}(X_g) \neq 0$ and all others $f_{gl}(X_g) \equiv 0$ because they assume that only a few genes regulate the expression of the targe gene l when the number of genes, G, is large. Another assumption is the regulation effect of a particular regulatory gene might only be significant when its expression level is within a certain range. In other words, $f_{gl}(X_g) \neq 0$ when $X_g \in S_{gl}$ and $f_{gl}(X_g) = 0$ when $X_g \notin S_{gl}$ where S_{gl} is the support or nonzero intervals of the regulation function f_{gl}

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

where $\boldsymbol{\phi}_{glk}(X_g) = (\phi_{gl1}(X_g), \phi_{gl2}(X_g), ..., \phi_{glK_{gl}}(X_g))^T$ denotes the vector of B-spline basis functions, $\boldsymbol{\beta}_{gl} = (\beta_{gl1}, \beta_{gl2}, ..., \beta_{glK_{gl}})$ is the corresponding vector of B-spline basis coefficients, and K_{gl} denotes the number of basis functions. If all $\boldsymbol{\beta}_{gl}$ are estimated to be zero, the $f_{gl}(X_g) \equiv 0$, and the corresponding gene is omitted from the ODE model. On the other hand, if only a few elements of $\boldsymbol{\beta}_{gl}$ are estimated to be zero, then the corresponding regulation function $f_{gl}(X_g)$ will be strictly zero in certain intervals.

3.2 Sparsity Penalty

In this article, the functional SCAD was used to deal with sparsity in ODE model (1). The fSCAD penalty in our model is defined as

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

where x_{gl} and x_{gu} are the lower and upper bounds of the expression of the g-th gene $X_g(t)$, $t \in [0, T]$, $\Delta_{x_g} = x_{gl}$ - x_{gu}, and M_{gl} is the number of subintervals partitioned by the knots of B-spline basis functions (total number of interior knots plus one).

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

where $a > 2$, and $\lambda$is the tuning parameter, which controls the sparsity of the regulation functions.

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

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

3.3 Roughness Penalty

Roughness penalty is added to ODE model because of the assumption that the regulation function $f_{gl}(X_g)$ is a smooth function. It comes from the assumption the regulation effect is not expected to change dramatically when the regulatory gene's expression has a small change.

A roughness penalty based on B-spline basis function for a certain regulatory gene X_g is given as

$$\left\| \frac{df_{gl}^2(X_g(t))}{dt^2} \right\|^2 = \int_0^T \left(\frac{d^2 f_{gl}(X_g(t))}{dt^2} \right)^2 dt$$

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

where

$$d_{glk} = \frac{d^2 \phi_{glk}(X_g(t))}{dt^2} = \frac{d^2 \phi_{glk}}{dX_g^2} \left(\frac{dX_g}{dt} \right)^2 + \frac{d\phi_{glk}}{dX_g} \frac{d^2 X_g}{dt^2} \tag{4}$$

The following is the roughness penalty for all G regulation functions.

$$R_l = \sum_{g=1}^{G} \left\| \frac{df_{gl}^2(X_g(t))}{dt^2} \right\|^2 = \sum_{g=1}^{G} \int_0^T \left(\sum_{k=1} K_g l \beta_{glk} d_{glk} \right)^2 dt \tag{5}$$

3.4 Parameter Estimation

They estimated $f_{gl}(X_g)$ by minimizing the following loss function:

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

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

The first term of equation (6) can be expressed as

$$\frac{1}{n} \sum_{i=1}^{n} \left(\dot{\boldsymbol{X}}_{\boldsymbol{l}}(t_i) - \sum_{g=1}^{G} f_{gl}(X_g(t_i)) \right)^2 = \frac{1}{n} (\dot{\boldsymbol{X}}_l(t) - \boldsymbol{\Phi}_l^T \boldsymbol{\beta}_l)^T (\dot{\boldsymbol{X}}_l(t) - \boldsymbol{\Phi}_l^T \boldsymbol{\beta}_l) \tag{7}$$

where $\dot{\boldsymbol{X}}_l(t)$ = ($\dot{X}_l(t_1)$,$\dot{X}_l(t_2)$,...,$\dot{X}_l(t_n)$)T is a length n column vector, $\boldsymbol{\Phi}_l$ = [$\boldsymbol{\Phi}_{1ln}$,...,$\boldsymbol{\Phi}_{Gln}$] is a GKxn matrix, $\boldsymbol{\Phi}_{gln}$ = [$\boldsymbol{\phi}_{gl}(X_g(t_1))$,...,$\boldsymbol{\phi}_{gl}(X_g(t_n))$] is a K_{gl}xn matrix, and $\boldsymbol{\phi}_{gl}(X_g(t_1))$ = ($\phi_{gl1}(X_g(t_1))$,...,$\phi_{glK_{gl}}(X_g(t_1))$)^T.

The roughness penalty in (6) is transformed into the following form:

$$\gamma \sum_{g=1}^{G} \left\| \frac{df_{gl}^2(X_g(t))}{dt^2} \right\|^2 = \gamma \boldsymbol{\beta}_l \boldsymbol{V}_l \boldsymbol{\beta}_l \tag{8}$$

where $\mathbf{V_l} = diag(\mathbf{V_{1l}}, ..., \mathbf{V_{Gl}}$ is a matrix ($\mathbf{GK_{gl}} x \mathbf{GK_{gl}}$ with blocks $\mathbf{V_{1l}}, ..., \mathbf{V_{Gl}}$ in its main diagonal and zeros elsewhere.

The fSCAD penalty in (6) can be approximated as

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

The approximation process

1. Based on equation (3) the fSCAD penalty is approximated as

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

$$\| f_{gl[j]} \|_2^2 \stackrel{def}{=} \int_{x_{g,j-1}}^{x_{gj}} [f_{gl}(X_g)]^2 dX_g = \beta_{\mathbf{gl}}^{\mathbf{T}} M_{glj} \beta_{gl}$$

where M_{glj} is a $K_{gl} x K_{gl}$ matrix with entries $m_{glj,uv} = \int_{x_{g,j-1}}^{gj} \phi_{glu}(X_g)\phi_{glv}(X_g)dX_g$

2. Using the local quadratic approximation (LQA) porposed in Fan and Li (2001), given some initial estimate $\beta_{\mathbf{gl}}^{(\mathbf{o})}$, they can derive

$$\frac{M_{gl}}{\Delta_{xg}} \int_{x_{gl}}^{x_{gu}} p_\lambda(|f_{gl}(X_{gl})|)dX_g \approx \beta_{\mathbf{gl}}^{\mathbf{T}} \mathbf{W}_{\mathbf{gl}}^{(}\mathbf{0})\beta_{\mathbf{gl}} + \mathbf{G}(\beta_{\mathbf{gl}}^{(\mathbf{0})}),$$

where

$$\mathbf{W}_{gl}^{(0)} = \frac{1}{2} \sum_{j=1}^{M_{gl}} \left(\frac{\dot{p_\lambda}(\| f_{gl[j]} \|_2 \sqrt{M_{gl}/\Delta_{xg}})}{\| f_{gl[j]} \|_2 \sqrt{\Delta_{xg}/M_{gl}}} M_{glj} \right)$$

and

$$\mathbf{G}(\beta_{gl}^{(0)}) \equiv \sum_{j=1}^{M_{gl}} p_\lambda \left(\frac{\| f_{gl[j]} \|_2}{\sqrt{\Delta_{xg}/M_{gl}}} \right) - \frac{1}{2} \sum_{j=1}^{M_{gl}} \dot{p}_\lambda \left(\frac{\| f_{gl[j]} \|_2}{\sqrt{\Delta_{xg}/M_{gl}}} \right) \frac{\| f_{gl[j]} \|_2}{\sqrt{\Delta_{xg}/M_{gl}}}$$

By minimizing $\mathbf{Q}(\beta_{\mathbf{l}})$, they obtain the estimate for the basis coefficients

$$\hat{\boldsymbol{\beta}}_l = \frac{1}{n} \left(\frac{1}{n} \boldsymbol{\Phi}_l \boldsymbol{\Phi}_l^T + \gamma \boldsymbol{V}_l + \boldsymbol{W}_l^{(0)} \right)^{-1} \boldsymbol{\Phi}_l \dot{\boldsymbol{X}}_l$$

However, there is an identifiability issue. To solve this problem, they apply a similar strategy as in Wood (2006), which constrins the sum of $f_{1l}(\cdot)$ to zero over the entire time domain. That is,

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

The constriant (10) can be satisfied in a sample as

$$\sum_{g=1}^{G} \left(\sum_{k=1}^{K_{gl}} \beta_{glk} \omega_{glk} \right)^2 = \boldsymbol{\beta}_l^T \boldsymbol{\Omega}_l \boldsymbol{\beta}_l = 0 \tag{11}$$

where

$$\omega_{glk} = \sum_{i=1}^{n} \phi_{glk}(X_g(t_i)),\ \boldsymbol{\Omega}_l = diag(\boldsymbol{\Omega}_{1l}, \boldsymbol{\Omega}_{2l}, ..., \boldsymbol{\Omega}_{Gl})$$

and

$$\boldsymbol{\Omega}_{gl} = \begin{bmatrix} \omega_{gl1}^2 & \omega_{gl1}\omega_{gl2} & \cdots & \omega_{gl1}\omega_{glK_{gl}} \\ \omega_{gl2}\omega_{gl1} & \omega_{gl2}^2 & \cdots & \omega_{gl2}\omega_{glK_{gl}} \\ \vdots & \vdots & \cdots & \vdots \\ \omega_{glK_{gl}}\omega_{gl1} & \omega_{glK_{gl}}\omega_{gl2} & \cdots & \omega_{glK_{gl}}^2 \end{bmatrix}$$

By adding $\lambda_I \boldsymbol{\beta}_l^T \boldsymbol{\Omega}_l \boldsymbol{\beta}_l$ to the loss function (6), in which λ_I is a realtively large positive number, the estimator $\boldsymbol{\beta}_l$ becomes

$$\hat{\boldsymbol{\beta}}_l = \frac{1}{n}\left(\frac{1}{n}\boldsymbol{\Phi}_l\boldsymbol{\Phi}_l^T + \gamma \boldsymbol{V}_l + \boldsymbol{W}_l^{(0)} + \lambda_l \boldsymbol{\Omega}_l\right)^{-1} \boldsymbol{\Phi}_l \dot{\boldsymbol{X}}_l \tag{12}$$

3.5 Derivative Estimation

To estimate the expressions and derivatives of each gene in equation (1), $X(t)$ and $\dot{X(t)}$, they introduced a smoothing spline method.

Assumption: Y_i, $i = 1, ..., n$, which is from an unknown gene expression function $X(t)$

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

where

$$\epsilon_i \overset{iid}{\sim} N(0, \sigma_s^2)$$

and $X(t)$ can be expressed as a linear combination of B-spline basis functions:

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

They estimate the vector of coefficients θ can be obtained by minimizing the loss function:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{argmin} \textstyle\sum_{i=1}^{n} (Y_i - X(t_i))^2 + \lambda_o \int \left[\ddot{X}(t)\right]^2 dt, \lambda_0 > 0 \\ &= \underset{\boldsymbol{\theta}}{argmin} (\boldsymbol{Y} - \boldsymbol{\Psi\theta})^T (\boldsymbol{Y} - \boldsymbol{\Psi\theta}) + \lambda_o \boldsymbol{\theta^T R \theta} \end{aligned} \tag{13}$$

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \lambda_o \boldsymbol{R})^{-1}\boldsymbol{\Psi}^T\boldsymbol{Y}$$

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

$$\frac{1}{n}(\dot{\boldsymbol{X}}_l(t) - \boldsymbol{\Phi}_l^T\boldsymbol{\beta}_l)^T \left[\widehat{Cov}(\hat{\dot{\boldsymbol{X}}})\right]^{-1} (\dot{\boldsymbol{X}}_l(t) - \boldsymbol{\Phi}_l^T\boldsymbol{\beta}_l) \tag{14}$$

where

$$\widehat{Cov}(\hat{\dot{\boldsymbol{X}}}) = \dot{\boldsymbol{\Psi}}^T \widehat{Cov}(\hat{\theta})\dot{\boldsymbol{\Psi}} = \hat{\sigma}_s^2 \dot{\boldsymbol{\Psi}}^T (\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \lambda_0 \mathbf{R})^{-1}\boldsymbol{\Psi}^T\boldsymbol{\Psi}\boldsymbol{\Psi}(\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \lambda_0 \mathbf{R})^{-1}\dot{\boldsymbol{\Psi}} \tag{15}$$

in which $\hat{\dot{\boldsymbol{X}}} = (\hat{\dot{X}}(t_1), ..., \hat{\dot{X}}(t_n))^T$, $\dot{\boldsymbol{\Psi}}$ is a n x J matrix with entries $\dot{\psi_j}(t_i)$, $\mathbf{R}$ is a J x J matrix with entries $R_{ij} = \int \ddot{\psi}_i(t)\ddot{\psi}_t(t)dt$, $\boldsymbol{\Psi}$ is an n x J matrix with entries $\Psi_{ij} = \psi_j(t_i)$, and $\hat{\sigma}_s^2$ can be obtained by computing the sample variance of the residuals $\mathbf{e}_s = \mathbf{Y} - \boldsymbol{\Psi}\hat{\theta}$.

Consiquently, equation (12) becomes

$$\hat{\boldsymbol{\beta}}_l = \frac{1}{n}\left(\frac{1}{n}\boldsymbol{\Phi}_l\boldsymbol{L}_l^T\boldsymbol{L}_l\boldsymbol{\Phi}_l^T + \gamma\boldsymbol{V}_l + \boldsymbol{W}_l^{(0)} + \lambda_l\boldsymbol{\Omega}_l\right)^{-1}\boldsymbol{\Phi}_l\boldsymbol{L}_l^T\boldsymbol{L}_l\dot{\boldsymbol{X}}_l$$

where

$$\left[\widehat{Cov}(\hat{\dot{\boldsymbol{X}}})\right]^{-1} = \boldsymbol{L}_l^T\boldsymbol{L}_l$$

4 Simulation

4.1 Simulation Setting

To mimic the real gene regulation process, 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:

$$X_0(t) = \int_0^t \dot{X}_0(\tau)d\tau = \int_0^t \sum_{i=1}^{20} f_i(X_i(\tau))d\tau, \;\; \tau \in (0, 1, 2, ..., 23) \tag{16}$$

where $\dot{X}_0(\tau)$ denotes the derivaive of the expression for the target gene and $X_i(\tau)$ is the expression function of gene i at time τ.

- $X_0(t) + \epsilon, \;\; \epsilon \overset{iid}{\sim} N(0, \rho), \;\; \rho \in$(1,5)
- $X_0(t) \leftarrow \hat{X_0}(t) = \int_0^t \hat{f}_1(X_1(\tau)) + \hat{f}_2(X_2(\tau)) + \hat{f}_3(X_3(\tau))d\tau$
- Locally sparse method : $\lambda \in$ (10,1,$10^{-1}, 10^{-2}$), $\gamma \in$ (10,$10^{-1}, 10^{-3}, 10^{-5}$)
- Smoothing spline method : λ=0, $\gamma \in$ (10,$10^{-1}, 10^{-3}, 10^{-5}$)
- Linear fSCAD method : $\lambda \in$ (10,1,$10^{-1}, 10^{-2}$), γ=100

The model parameters, λ, γ, are determined by AICc. The number of basis function and the identifiability parameter are set as $K_{gl} = 5$ and $\lambda_I = 10^4$ to ease the computation.

4.2 Simulation

The simulation is repeated for 100 times and the results is followings. The model performance for each method is accessed using the false negative error (FN), the true positive error (TP) and the false positive error (FP).

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

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

$$\text{TP} = \frac{\#\text{ of correctly estimated regulatory genes}}{\#\text{ of all true regulatory genes}}$$

The following table shows the FN, FP, and TP error.

Table 1

		TP(%)		FP(%)		FN(%)	
Method	$\rho(\%)$	Mean	SD	Mean	SD	Mean	SD
Locally Sparse	1	98.0	4.4	3.5	1.5	1.7	4.4
	5	100.0	0.0	6.0	5.6	0.0	0.0
Smoothing Spline	1	100.0	0.0	35.0	0.0	0.0	0.0
	5	100.0	0.0	35.0	0.0	0.0	0.0
Linear fSCAD	1	92.4	3.1	16.0	3.7	7.6	3.2
	5	92.8	2.1	23.7	3.4	7.2	2.1

Table 1 shows TP, FP, and FN error. Locally Sparse method and smoothing spline method have well estimated the regulatory genes. On the other hand, smoothing spline method shows high FP error than locally sparse method. It means that prediction of smoothing spline method is poor than locally sparse method. Linear fSCAD method have lowest TP error among three methods and higher FP error than locally sparse method. FN error of locally sparse method and smoothing spline method both have lower FN error relatively compared to linear fSCAD method. In conclusion, locally sparse method is the best method to estimate the true regulation functions among three methods.

Figure 1. Estimated regulation coefficients from the simulated data with the noise-to-signal ratio of the simulated data $\rho = 1\%$ using the locally sparse method. The gray lines are estimated regulation coefficients in 100 simulation replicates.

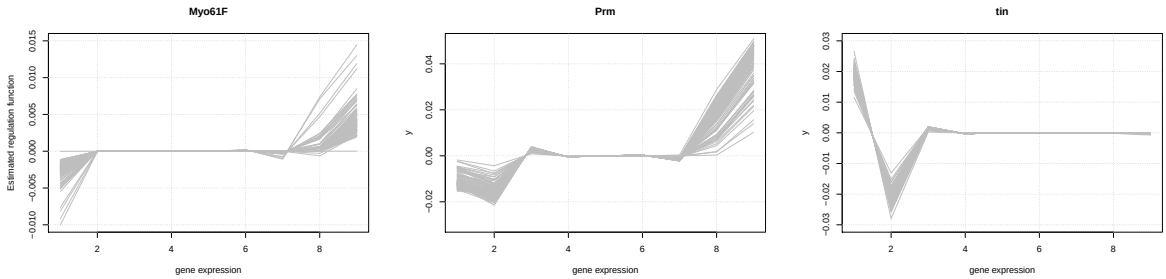
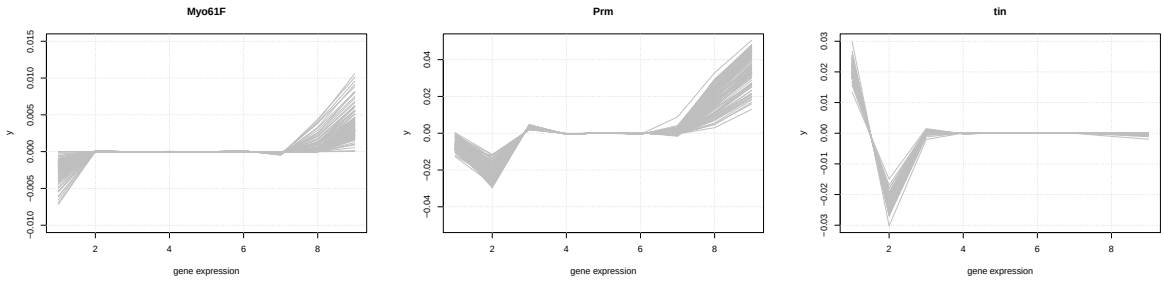


Figure 2. Estimated regulation coefficients from the simulated data with the noise-to-signal ratio of the simulated data $\rho = 5\%$ using the locally sparse method. The gray lines are estimated regulation coefficients in 100 simulation replicates.



The plots of estimated regulation coefficients from the simulation data with the noise-to-signal ratio 1 and 5 are shown Figure 1 and Figure 2. These plots have similar appearances with the real regulation regulation functions. Estimated regulation coefficients with $\rho = 5\%$ have high variance than estimated regulation coefficients with $\rho = 1\%$ because the lines in Figure 2 are more spread over than them in Figure 1.

5 Conclusion

In summary, the high dimensional nonlinear ODE model can be well applicable to the dynamical regulation process. This article suggested the method of the nonlinear ODE model with sparsity and roughness penalty. Furthermore, with simulation, we can check that this model is able to estimate the true regulation functions under different noise condition.

References

[1] P. Howard. (2009). Overview. *Modeling with ODE*, Texas AM University.

[2] Jianqing Fan and Runze Li. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. Penn State University. pp. 1348-1351.

[3] The Formula for Quadratic Approximation. 18.01SC Single Variable Calculus. MIT OpenCourseWare.