Afrika Statistika

**Afrika Statistika**
Vol. 12 (2), 2017, pages 1251–1271.
DOI: http://dx.doi.org/10.16929/as/1253.103

ISSN 2316-090X

# Network estimation in State Space Models with L1-regularization constraint

**Anani Lotsi**[1,2,*] **and Ernst Wit**[2]

[1]University of Ghana Legon, Department of Statistics, P.O.Box: 115 LG
[2]Department of Statistics and Probability, University of Groningen Nijenborgh 9, 9747 AG The Netherlands

Received April 25, 2017; Accepted June 08, 2017



**Abstract.** Microarray technologies and related methods coupled with appropriate mathematical and statistical models have made it possible to identify dynamic regulatory networks by measuring time course expression levels of many genes simultaneously. However one of the challenges is the high-dimensional nature of such data coupled with the fact that these gene expression data are known not to include various biological process. As genomic interactions are highly structured, the aim was to derive a method for inferring a sparse dynamic network in a high dimensional data setting. The paper assumes that the observations are noisy measurements of gene expression in the form of mRNAs, whose dynamics can be described by some partially observed process.

**Résumé.** Microarray technologies and related methods coupled with appropriate mathematical and statistical models have made it possible to identify dynamic regulatory networks by measuring time course expression levels of many genes simultaneously. However one of the challenges is the high-dimensional nature of such data coupled with the fact that these gene expression data are known not to include various biological process. As genomic interactions are highly structured, the aim was to derive a method for inferring a sparse dynamic network in a high dimensional data setting. The paper assumes that the observations are noisy measurements of gene expression in the form of mRNAs, whose dynamics can be described by some partially observed process.

**Key words:** genomic; gene expression; microarray; sparse; EM algorithm; state space model.
**AMS 2010 Mathematics Subject Classification :** Primary 62-09; Secondary 62H12; 62J07.

*alotsi@ug.edu.gh
e.c.wit@rug.nl

## 1. Introduction

Modelling differential gene expression as a function of time is providing new insights for biological research. Technology is now available to track the expression pattern of thousands of genes in a cell in a regulated fashion and to trace the interactions of many of the products of these genes (Bower *et al.*, 2001). However, the sheer dimensionality of all possible networks combined with the noisy nature of the observations and the complex structure of genomic regulation and signaling have meant that simply reading off a network from the data turned out to be somewhat optimistic. Instead, only statistical models of sufficient biological relevance are capable of discovering direct and indirect interactions between genes, proteins and metabolites. The last decade has seen an explosion of techniques to infer network structure from microarray data. Models have now been developed to capture how information is stored in DNA, transcribed to mRNA, translated to proteins and then from protein structure to function. These models include Boolean networks based on Boolean logic (Kauffman, 1993) and (Dhaeseleer *et al.*, 2000), where each gene is assumed to be in one of two states "expressed" or "not expressed", Graphical Gaussian models (Schafer *et al.*, 2005; Abegaz *et al.*, 2013), Dynamic Bayesian Networks (Perrin *et al.*, 2003), vector autoregressive models (Fujita *et al.*, 2007), ordinary differential equation models (Quach *et al.*, 2007; Cao *et al.*, 2008) or (Khanin *et al.*, 2006, 2007) in which the state is a list of the concentrations of each chemical species and the concentrations are assumed to be continuous, stochastic differential equation model (Chen *et al.*, 2005; Purutcuoglu *et al.*, 2008) and finally state space models (Rangel *et al.*, 2004; Beal *et al.*, 2005).

Integrating these models in mainstream statistics is an exciting challenge from a theoretical, computational, and applied perspective. Among the above mentioned network modelling techniques, ordinary differential equations have been established in recent years to model gene regulatory, or, more generally biochemical-networks, since they provide a detailed quantitative description of transcription regulatory network. On the downside, they contain a large number of model parameters and are not well suited to deal with noisy data. Current methods for estimating parameters in ODEs from noisy data are very computationally intensive (Ramsay *et al.*, 2007).

In our work, we consider a penalized state space model (SSM) framework which consists of two different spaces, i.e a latent "protein" space and an observed "mRNA" space. SSM's are special cases of dynamic Bayesian networks (DBNs) and include hidden factors into the model, eg. genes whose protein expression values are not measured. The standard SSM (Fahrmeir *et al.*, 2009, 1997), in the context of time-series gene expression, assumes that the observed time series expression data, $y_t$, represent a $p$-dimensional vector of gene expression observations of $p$ genes at time $t$. The $y_t$ are assumed to have come from an underlying sequence of $k$ unobserved (hidden) state variables $\theta_t$ that evolve according to Markovian dynamics across successive time points. In a essence the model consists of a linear observation equation in states and is supplemented by a linear transition equation. A linear SSM as in (Koopman *et al.*, 2001) can be written :

$$\begin{aligned} \theta_t &= F\theta_{t-1} + \eta_t \\ y_t &= Z\ \theta_t + \xi_t \end{aligned} \tag{1}$$

,
where $F$ and $Z$ represent the model coefficients of dimensions $k \times k$ and $p \times k$ respectively. The two terms $\eta_t$ and $\xi_t$ are zero-mean independent system noise and measurement noise, respectively with

$$E(\eta_t \eta_t^{'}) = Q, \qquad E(\xi_t \xi_t^{'}) = R. \tag{2}$$

Choosing SSMs to model network kinetics has a number of advantages. Most importantly, it allows the inclusion of hidden regulators which can either be unobserved gene expression values or transcription factor protein levels. The assumption of having incomplete data is quite realistic in the sense that in a microarray experiment, we usually do not observe protein concentrations together with mRNA concentrations due to the technical difficulty involved in performing such experiments. Thus we see the data as noisy measurements of mRNA concentrations, whose dynamics can be described by some hidden process which involves protein transcriptions factors and mRNA concentrations.

Several authors have exploited Kalman filtering (Shumway *et al.*, 2005; Meinhold *et al.*, 1983) in the context of a SSM for gene expression modelling and have used them to reverse engineer transcriptional networks. To this effect, Wu *et al.* (2004), in modelling gene regulatory networks, used a two step approach. In the first step, factor analysis is employed to estimate the state vector $\theta$ and the design matrix $Z$; the optimum dimension of the state vector $k$ was determined by minimizing the Bayesian information criterion (BIC). In the second step, the state dynamic matrix $F$ is estimated using least squares regression. Rangel *et al.* (2004) applied SSM to t-cell activation data in which a bootstrap procedure was used to derive a classical confidence interval for parameters representing gene-gene interaction through a re-sampling technique. Beal *et al.* (2005) approached the problem of inferring the model structure of the SSM using variational approximations in the Bayesian context. Recently, Bremer *et al.* (2009) used SSM to rank observed genes in gene expression time series experiments according to their degree of regulation in a biological process. Their technique is based on Kalman smoothing and maximum likelihood estimation techniques to derive optimal estimates of the model parameters; however, in their work, little attention has been paid to determining the dimension of the hidden state.

In microarray analysis, the number of predictors usually in the form of genes far exceed the number of observations ($p >> n$). Faced with such explosion of predictors, regularization has become an important ingredient and is fundamental to high-dimensional statistical modelling. The Lasso of Tibshirani *et al.* (1996) is one of the methods for shrinkage and selection in regression analysis that incorporates an $L_1$ regularization constraint to yield a sparse solution. A considerable amount of literature has been published on regularization methods in areas with large data sets such as genomics. These studies include, the regularization paths for the support-vector machine (Hastie *et al.*, 2004), the elastic net (Zou *et al.*, 2005) for applications with unknown groups of predictors

and useful for situations where variables operate in correlated groups, $L_1$ regularization paths for generalized linear models (Park *et al.*, 2007) and the graphical lasso (Friedman *et al.*, 2008) for sparse covariance estimation and undirected graphs to mention but few.

Gene regulatory networks are usually sparse. For that reason, we will expect many of the gene interaction parameters to be zero leading to a sparse solution. It is in this context that we employ an $L_1$ regularization approach for the estimation of the parameters. The proposed method in the maximization step of the EM-algorithm is the $L_1$ penalty through a simple modification of the Least Angle Regression (LARS) algorithm by Efron *et al.* (2004). LARS is an efficient algorithm for computing the entire regularization path for the Lasso.

State space models are robust candidates to represent interactions between biological components in the form of mRNA concentrations and protein transcriptions factors. We present a statistical method that infers the complexity, the dependence structure of the network topology and the functional relationships between the genes, and deduce the kinetic structure of the network. We estimate all network interaction parameters in order to clarify and describe the complex transcriptional response of a biological system and to clarify interactions between components. By so doing, we are able to add some useful interpretations to the model. We use the minimum Akaike information criterion (AIC) to determine the optimum level of sparsity.

The rest of this article is organized as follows. In section 2, we introduce the model, and give it a precise mathematical and biological interpretation. Section 3 describes the inference method and the model selection technique. We perform a simulation study and an *in silico* validation experiment in order to evaluate the performance of our method in section 4. Section 5 contains the application of our model to a real t-cell experiment. We conclude with a discussion of the method used, possible extensions, and a summary of related work in section 6.

## 2. GENOMIC STATE SPACE MODEL

We extend the SSM model in equation 1 by considering an input dependent SSM for gene expression times series data, as in (Beal *et al.*, 2005). The framework captures the stochastic nature of our biological process and its dynamics. To define the model, we start with the definition of the state variables $\theta_t$ represented as hidden process and the observation measurements are assumed to be produced by these hidden processes. The model assumes that the evolution of the hidden variables $\theta_t$ is governed by the state dynamics which follow an input dependent first-order Markov process. The hidden variables $\theta_t$ are classically used to represent genes that have not been included in the microarray experiment such as unmeasured protein regulators or transcription factors. The hidden variables are further corrupted by Gaussian intrinsic biological noise $\eta_t$. The hidden variables are not directly accessible but their presence can be induced through the observed data vector, $y_t$, namely the quantity of mRNA produced by the gene at time $t$. In essence we build a dynamic model that connects the observed variables $y_t$ (RNA transcripts) to the $k$-dimensional real valued unobserved quantities $\theta_t$ such as un-

measured typically protein regulators. Our model is defined through the following dynamics:

First the state dynamics or the state of the network satisfies an input dependent first-order Markov process

$$\theta_{tr} = F\theta_{t-1,r} + Ay_{t-1,r} + \eta_{tr}, \tag{3}$$

where $F$ is a regulatory matrix that quantifies the effect of the latent variables at consecutive time points and is of dimension $k \times k$. The quantity $A$ represents the input-to-state matrix whose dimension is $k \times p$. The quantity $r = \{1, 2, ..., n_R\}$ denotes biological replicates and $\eta_{tr}$ is the Gaussian noise with mean 0 and variance-covariance matrix $Q$. The initial state $\theta_0$ is Gaussian distributed with mean $a_0 = 0$ and variance-covariance $Q_0$.

Second, the $p$ dimensional-observation $y_t$ is a possibly time-dependent linear transformation of the $k$ dimensional real-valued $\theta_t$ with observational Gaussian noise $\xi_t$ and is given by

$$y_{tr} = Z\theta_{t,r} + By_{t-1,r} + \xi_{tr}, \tag{4}$$

where $Z$ describes how the latent variables regulate the transcription of $p$ genes and is of dimension $k \times p$. The matrix $B$ represents either degradation or interaction of mRNAs also known as input-to-observation matrix whose dimension is $p \times p$ and $\xi_{tr}$ is the measurement Gaussian noise with mean 0 and variance-covariance matrix $R$.

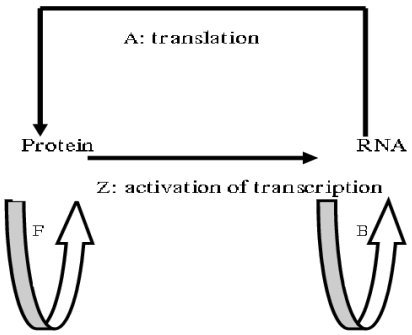


**Fig. 1.** Biological interpretation of the input SSM.

The input dependent state space model defined by equations 3 and 4 is an extension of the central SSM defined in equation 1 and has been exploited in reverse-engineering transcriptional network (Beal *et al.*, 2005; Rangel *et al.*, 2004).

The model indicates two networks, one in the protein space and another in the mRNAs space, across consecutive time points. It assumes RNA-protein translation at two consecutive time points through the matrix $A$, and instantaneous protein-RNA transcription through $Z$. A biological interpretation of the model network is also represented in Figure 1, which describes two fundamental stages in gene regulation which are in conformity with the central dogma which states that DNA does not code for protein directly but rather acts through

2 stages, namely, transcription and translation. The observation-to-state matrix $A$ models the influence or the effects of the gene expression values from previous time steps on the hidden states. The matrix $B$ indicates the direct gene-gene interactions. The state dynamic matrix $F$ describes the temporal development of the regulators or the evolution of the hidden variables from previous time step $t-1$ on the current time step $t$. It provides key information on the influences of the hidden regulators on each other. The observation dynamics matrix $Z$ relates the latent variables to the RNAs at a given time point. We now collect the model interaction parameters into a single vector $\varphi$ i.e $\varphi = \{G, Q, R, Q_0\}$ where $G = \begin{bmatrix} B & Z \\ A & F \end{bmatrix}$ is interpreted as a directed and weighted adjacency matrix of the graph of interactions.

## 3. LEARNING STATES AND PARAMETERS

### *3.1. Identifiability issues*

Briefly speaking, a parameter of a dynamic system is said to be identifiable given some data if only one value of this parameter maximizes the observed likelihood. The identifiability property is important because it guarantees that the model parameter can be determined uniquely from the available data. Identifiability issues of the SSM stems from the fact that given the original model equations 3 and 4, and with the linear transformation of the state vector $\theta_t^* = T\theta_t$, where $T$ is a non-singular square matrix, we can find a different set of parameter vector say,

$$\hat{\varphi}^* = \left\{\hat{G}^*, \hat{Q}^*, \hat{R}^*\right\} \tag{5}$$

that gives rise to the same observation sequence $\{y_t, t = 1, 2, ..., T\}$ having the same likelihood as the one generated by the parameter vector $\varphi$. Hence, if we place no constraints on $F$, $A$, $Z$, $B$ and possibly $Q$ and $R$, there exists an infinite space of equivalent solutions $\hat{\varphi}$ all with the same likelihood value. To minimize such identifiability issues, further restrictions have to be imposed on the model. In our work, we assume $Q$ and $Q_0$ to be identity matrices and $R$ is set to be diagonal matrix. Assuming $Q$ to be identity only affects the scale of $\theta$ and matrices $A$ and $Z$. On the individual parameters $Z$, $A$, $B$ and $F$, Wild *et al.* (2004) showed that only the matrix $ZA + B$ is identifiable. However our $L_1$ constraint on the likelihood identifies the individual parameters. We further assume that the errors $\{\eta_t; t = 1, ..., T\}$ and $\{\xi_t; t = 1, ..., T\}$ are uncorrelated.

### *3.2. The likelihood function*

The model interaction parameters are now restricted to $\varphi = \{Z, B, F, A\}$ which is equivalent to our graph of interactions $G$. As can be seen from the model, the observations at time $t$, $y_{tr}$ are conditioned on the past observations, $y_{(t-1)r}$ and on the regulators $\theta_{tr}$ and also to infer for instance $\theta_{tr}$, we need $\theta_{(t-1)r}$ and $y_{(t-1)r}$. To that effect, under the Gaussian assumption we have the following:

$$\begin{aligned} \theta_{0r} &\sim \mathcal{N}_k(0, I) \\ \theta_{tr}|\theta_{(t-1)r}, y_{(t-1)r} &\sim \mathcal{N}_k(\tilde{\theta}_{tr}, I) \\ y_{tr}|\theta_{tr}, y_{(t-1)r} &\sim \mathcal{N}_p(\tilde{y}_{tr}, R), \end{aligned}$$

where

$$\begin{aligned} \tilde{\theta}_{tr} &= F\theta_{(t-1)r} + Ay_{(t-1)r}, \\ \tilde{y}_{tr} &= Z\theta_{tr} + By_{(t-1)r}, \end{aligned}$$

and $\mathcal{N}(\mu, \Sigma)$ is the normal density with mean $\mu$ and variance covariance matrix $\Sigma$.

We now write the marginal likelihood function $l_y^m(\varphi)$ of the data $y$. This is given by

$$\begin{aligned} l_y^m(\varphi) &= \int \prod_{t=1}^{T} P(\theta_t|F, A, \theta_{t-1}, y_{t-1}) P(y_t|B, Z, \theta_t, y_{t-1}) d\theta \\ &= \int \prod_{t=1}^{T} \mathcal{N}_k(\theta_t|\tilde{\theta}_t, \sigma_\eta^2 I) \mathcal{N}_p(y_t|\tilde{y}_t, \sigma_\xi^2 I) d\theta. \end{aligned} \tag{6}$$

Learning the parameters of a state space model including the hidden variable can be tackled from different approaches. Beal *et al.* (2005) inferred the parameters in the SSM using a Bayesian approach through a Variational Bayes Method (VBM) that approximates the posterior quantities required for Bayesian learning. As a probabilistic model, Wild *et al.* (2004) estimated the parameters trough a frequentist approach using maximum likelihood inference in the context of EM algorithm.

In our context, the number of parameters to be estimated $P = k^2 + 2kp + p^2$ far exceeds the number of observations. Thus we want to shrink unnecessary coefficients to zero. This will make interpretation of results easier and probably reflects the true underlying situation by introducing some level of sparsity. The formulation of our problem becomes

$$\hat{\varphi} = \arg\max_{\varphi}(l_y^m) \tag{7}$$

subject to the constraints

$$||Z||_1 \leq s_1, \qquad ||B||_1 \leq s_2, \qquad ||A||_1 \leq s_3, \qquad ||F||_1 \leq s_4, \tag{8}$$

where $s_i$ represents the regularization parameters or penalty parameters and we allow different penalty parameters for different coefficients. equations (7) and (8) are called a constrained regression problem also called a penalized state space models. Our $L_1$ constraint not only promotes sparsity but also minimizes identifiability problems.

To find the solution to the above problem, many well developed procedures can be used. For example, quadratic programming (Tibshirani *et al.*, 1996), the shooting algorithm (Fu *et al.* ,

1998), local quadratic approximation (Fan *et al.*, 2001) and most recently, the LARS method by Efron *et al.* (2004) can all be employed. Our proposed method adapt the later procedure, optimization under $L_1$ constraint, where a penalty term is added to the likelihood function giving rise to a penalized likelihood criterion. LARS or optimization with $L_1$-regularization constraint turns out to be helpful and computationally feasible approach for finding sparse solutions in high dimension and by so rendering model interpretation easier.

### *3.3. The EM algorithm*

It is important to realize that the integration in equation 6 involves the hidden component $\theta$, thus making the integration difficult.The Expectation-Maximization (EM) algorithm stems from the fact that if we did have the complete data $(y_t, \theta_t)$ it will be straight forward to obtain maximum likelihood estimators (MLEs) of $\varphi$ using multivariate normal theory. In this case, we do not have the complete data. Therefore we use the EM algorithm. The latter is an iterative method for finding the MLEs of $\varphi$ using the observed data $y_t$, by successively maximizing the conditional expectation of the complete data likelihood given the observed values.

The EM algorithm for SSM was formulated by Stoffer *et al.* (1982) and Shumway (2000). To this effect the algorithm requires the computation of the log-likelihood of the complete data $l_{\theta,y}(\varphi)$ and compute the conditional expectation of the log-likelihood given the data. The algorithm is a two-stage procedure in which we begin with a set of trial initial values for the model parameter to calculate the Kalman smoother $E(\theta_r)$ and $E(\theta_r\theta_r^{'})$. The Kalman output are then input into the M-step to update parameter estimates subject to the constraints in equation 8, giving rise to the EM for penalized likelihood estimation (Green *et al.*, 1990). The algorithm alternates recursively between an expectation step followed by a maximization step.

The full log-likelihood function of the complete data $(y_{tr}, \theta_{tr})$ denoted by $l_{y,\theta}(\varphi)$ is for simplicity given by

$$l_{y,\theta}(F, A, Z, B) = \sum_{r=1}^{n_R} l^r_{y_r\theta_r}(F, A, Z, B), \tag{9}$$

where $l^r_{y_r\theta_r}(F, A, Z, B)$ is the complete log-likelihood of the $r^{th}$ replicate and is given by

$$\begin{aligned} l^r_{y_r\theta_r}(F, A, Z, B) \;=\; & \sum_{t=1}^{T} l_{y_t|\theta_t, y_{(t-1)}}(Z, B) + \\ & \sum_{t=1}^{T} l_{\theta_t|\theta_{(t-1)}, y_{(t-1)}}(F, A). \end{aligned} \tag{10}$$

From now onwards for a given replicate and for simplicity, we will write the unpenalized complete log-likelihood as:

$$l_{y,\theta}(F, A, Z, B) = \sum_{r=1}^{n_R} l^r_{y_r\theta_r}(F, A) + \sum_{r=1}^{n_R} l^r_{y_r\theta_r}(Z, B) \tag{11}$$

3.3.1. The expected log-likelihood function: The E-step.

The E-step of the EM algorithm involves the calculation of the first two moments of $\theta$ of the hidden states i.e $E(\theta)$ and $E(\theta^{'}\theta)$. Let $\mathbf{Q}$ denote the expected log-likelihood. Then from equation 11, dropping the replicate index, $\mathbf{Q}$ becomes

$$\begin{aligned}\mathbf{Q}(\varphi|\varphi^*) &= E_{\theta,\varphi^*}\left[l_{y_t,\theta_t}(\varphi)|\varphi^* y\right] \\ &= \sum_{t=1}^{T} E_{\theta,\varphi^*}\left[l_{y_t,\theta_t}(\varphi)|\varphi^* y\right] \\ &= \sum_{t=1}^{T} E_{\theta,\varphi^*}\left[l_{y_t}(Z,B)|y\right] + \sum_{t=1}^{T} E_{\theta,\varphi^*}\left[l_{\theta_t}(F,A)|y\right] \\ &= \mathbf{Q}_1(Z,B) + \mathbf{Q}_2(F,A), \end{aligned} \tag{12}$$

where $\varphi^* = (Z^*, B^*, F^*, A^*)$ is the estimate obtained from the previous M-step,

$$\begin{aligned}\mathbf{Q}_1(Z,B) &= C_1 + 2\sum_{t=1}^{T} E(\theta_t^{'}) Z y_t + 2\sum_{t=1}^{T} y_{t-1}^{'} B^{'} y_t \\ &- \sum Z^{'} E(\theta_t^{'}\theta_t) Z - 2\sum_{t=1}^{T} E(\theta_t^{'}) Z B y_{t-1} \\ &- \sum_{t=1}^{T} B^{'} y_{t-1}^{'} y_{t-1} B \end{aligned} \tag{13}$$

and

$$\begin{aligned}\mathbf{Q}_2(F,A) &= C_2 + 2\sum_{t=1}^{T} y_{t-1}^{'} A^{'} E(\theta_t) + 2\sum_{t-1}^{T} E(\theta_t^{'}) F E(\theta_{t-1}) \\ &- \sum_{t=1}^{T} F^{'} E(\theta_{t-1}^{'}\theta_{t-1}) F - 2\sum_{t=1}^{T} y_{t-1}^{'} A^{'} F E(\theta_{t-1}) \\ &- \sum_{t=1}^{T} A^{'} y_{t-1}^{'} y_{t-1} A \end{aligned} \tag{14}$$

$C_1$ and $C_2$ are known constants and assuming that the prime denotes transposition.

The first two moments needed in the E-step are supplied by the Kalman smoothing algorithm through a forward filtering pass and a backward smoothing pass as in (Briers *et al.* , 2010). The above implies that for each replicate we run the Kalman smoothing algorithm to find the expected hidden states and their variance-covariance components and these are joined together to get $\mathbf{Q}(\varphi|\varphi^*)$.

Now equation 12 is the sum of two quadratic functions $\mathbf{Q}_1$ and $\mathbf{Q}_2$ that does not depend on $\theta$ but rather depend on the parameters and the data $y$ in a quadratic way. We maximize these two functions during the maximization step.

3.3.2. The update equations: The M-step.

At this stage we solve for

$$\hat{\varphi} = \arg\max_{\varphi} \left[\mathbf{Q}(\varphi|\varphi^*)\right] \tag{15}$$

subject to the constraints defined in equation 8. In essence we maximize the quadratic function $\mathbf{Q}$ given in equation 12 across $\varphi$ using LARS algorithm, where each coefficient is assigned a tuning parameter $s$. This breaks down to two maximization problems, one for $\mathbf{Q}_1$ across $(Z, B)$ and the other for $\mathbf{Q}_2$ across $(F, A)$. The iterative maximization process is similar in both cases.

We now show the maximization process for $\mathbf{Q}_1$. To do that, the following lemma is needed. The proof is found in the appendix.

**Lemma 1.** *The solution that maximizes the quadratic function*

$$\mathbf{Q}(X) = 2d^{'}X - X^{'}SX \qquad subject \quad to \quad ||X||_1 \quad \leq s \tag{16}$$

*is given by the lasso solution*

$$(\mathbf{y} - \mathbf{C}\beta)^{'}(\mathbf{y} - \mathbf{C}\beta) \qquad subject \quad to \quad ||X||_1 \quad \leq s \tag{17}$$

*where*

$$\mathbf{C}^{'}\mathbf{C} = S, \qquad \beta = Vec(X), \qquad \mathbf{y} = \mathbf{C}S^{-1}d. \tag{18}$$

*and the LARS solution of (17) is a function of $S$ and $d$.*

Now from equation 13, given $B$ we carry the maximization process of $\mathbf{Q}_1$ across $Z$. Therefore, we can write $\mathbf{Q}_1(Z, B)$ as

$$\mathbf{Q}_1(Z) = c_1 + 2b_1^{'}Z - Z^{'}S_1Z \tag{19}$$

subject to

$$\sum_{j=1}^{kp} |z_j| \leq s_2 \tag{20}$$

where $S_1 = E(\theta_t^{'}\theta_t)$, $b_1$ is just a function of $(y, B, \theta)$, and $c_1$ is a constant.

Applying lemma 1, the update maximum likelihood estimators $\hat{Z}$ from equation 19 are just a function of $S_1$ and $b_1$. We therefore obtained the updates estimates $\hat{Z}$ by supplying the

LARS function, the quantities $S_1$ and $b_1$ with a given tuning parameter where $b_1$ becomes the new data and $S_1$ the data matrix.

Next, given $\hat{Z}$, we maximize the quadratic function

$$\mathbf{Q}_1(B) = c_2 + 2b_2^{'}B - B^{'}S_2B \tag{21}$$

subject to

$$\sum_{j=1}^{p^2} |b_j| \leq s_1, \tag{22}$$

where $S_2 = \sum_{t=1}^{T} y_{t-1}^{'} y_{t-1}$, $b_2 = f(y, Z, \theta)$, $c_2$ is a constant. With the same analysis, the updates estimates $\hat{B}$ are obtained by supplying the LARS function, the quantities $S_2$ and $b_2$ with a given tuning parameter. Similar analysis is conducted for the estimation of $(F, A)$.

The advantage of this approach is that we see the LARS update as functions not of the raw data, but instead as functions of $S$ and $b$. This enables us to avoid first, the Cholesky decomposition of $S$ and second, computing $S^{-1}$ which are both time consuming and computationally inefficient.

1. Iterate across penalty parameters $s \in S$
   a. Start with initial values of $\varphi$
      i. Do the E-step by calculating the Kalman smoother
      ii. Perform the M-step via LARS algorithm
   b. Repeat (i) and (ii) until convergence
2. Across $S$ select model with minimum AICc

**Table 1.** Summary of the EM for Penalized Likelihood inference method

### *3.4. Model selection: Choice of regularization parameter s.*

Determining the optimal SSM tuning parameter $s$ is an important issue. Popular model selection criteria include the Mallow's $C_p$ (Mallows , 1973), the Akaike's Information Criterion (AIC) (Akaike , 1974) and the Bayesian Information Criterion (BIC) (Schwarz , 1978). We apply Akaike's Information Criterion (AIC) method for our model selection. We generate a vector of values for the tuning parameters $s_i (i = 1, ..., 4)$. For each combination of the values of the tuning parameters we run the EM algorithm and obtain

$$\hat{\varphi}(s) = \arg\max_{\varphi} [Q(\varphi)]$$

subject to constraints in equation 8. AIC is aimed at finding the best approximating model to the unknown data generating process via minimizing the estimated expected K-L divergence,


**Fig. 2.** The full true network $G$ (left) and the full recovered network $\hat{G}$ (right)


**Fig. 3.** Gene-gene interactions $ZA + B$. The true network(left) and the estimated network(right)

i.e. AIC's try to find the best approximation among the models we actually look at. As recommended by Burnham *et al.* (2002), we have applied $AIC_c$ (AIC with a correction for finite sample size) for our model selection procedure. The reason being that $AIC_c$ estimates the expected discrepancy with less bias than AIC. [1] The $AIC_c$ is given by

$$AIC_c(\hat{\varphi}(s)) = -2l(y_t) + 2P\left[\frac{N}{N-P-1}\right], \tag{23}$$

where $N = pTn_R$ represents total number of observations and $P = p^2 + 2kp + k^2$ is the total number of estimated parameters. Then for each model, the AICc is computed and the model with the minimum AICc is selected. In essence, minimizing AICc, we obtained the optimal tuning parameters which is given by

$$\hat{s} = \arg\min_{s}[AICc(\hat{\varphi}(s))]$$

and the selected model parameters are given by $\hat{\varphi}(\hat{s})$. Table 1 summarizes the general formulation of the EM-$L_1$ penalized inference method:

[1] In the framework of normal linear regression models (both univariate and multivariate), the penalty term of AICc provides an exact expression for the bias adjustment.

| $p$ | 10 | 20 | 30 | 40 |
|---|---|---|---|---|
| TPR | 0.41 | 0.45 | 0.25 | 0.52 |
| | (0.14) | (0.04) | (0.12) | (0.00) |
| FPR | 0.03 | 0.04 | 0.09 | 0.01 |
| | (0.01) | (0.00) | (0.03) | (0.00) |
| $F_1$-score | 0.47 | 0.39 | 0.10 | 0.49 |
| | (0.10) | (0.03) | (0.16) | (0.04) |

**Table 2.** Simulation results showing the average scores for true positive rates (TPR), false positive rates, (FPR) and $F_1$-scores as $p$, the number of nodes increase. For each $p$, we performed 50 different simulations and TPR, FPR and $F_1$ are the average scores. The numbers in parenthesis represent the standard deviations. We fixed $T$, the number of time points to be 10, and the number of replicates is chosen to be 50. The number of hidden states is assumed to be 2.

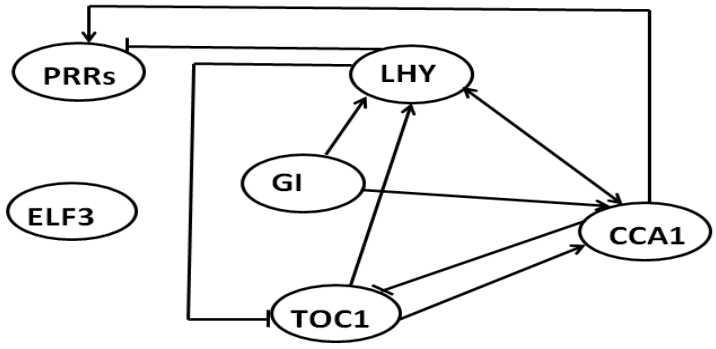


**Fig. 4.** The Arabidopsis thaliana clock network.

## 4. Validation of Method

### *4.1. Simulation*

In this section we evaluate our method on a simulated data based on the model described in equations 3 and 4 with 10 different time points, $p = 3$ as number of genes and $k = 2$ as the number of hidden variables. In applying our algorithm, parameters were initialized as follows: $Z$ and $F$ are assumed to be identity matrices whiles we initialize $A$ to be zero. For $B$ we perform a simple linear regression where we regress current genes on its previous ones and $R$ assumes the usual variance estimate from the regression.

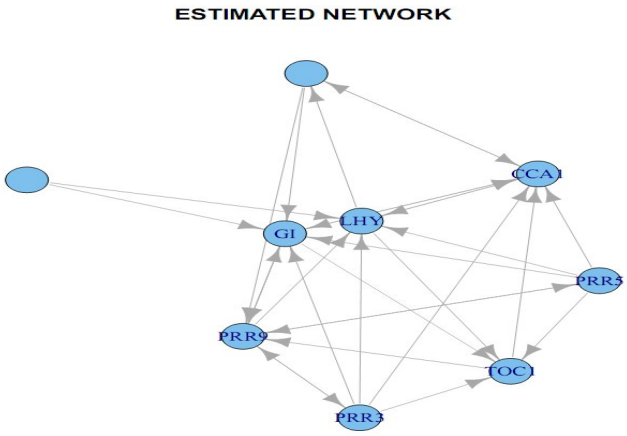


**Fig. 5.** Reverse genetic appraoch on Arabidopsis thaliana clock. Nodes refers to genes expression whiles empty nodes indicates latent variables in the form TFs.

The true (left) and the recovered (right) networks are depicted in Figure 2. The efficiency of our method is seen from the number of true links recovered. We obtained 71% as true positive rate and 27% as false positive rate with an $F_1$ score of about 58%. We also reported the gene-gene interaction matrix $\hat{Z}\hat{A} + \hat{B}$ in Figure 3. The matrix $\hat{Z}\hat{A} + \hat{B}$ captures all information related to gene-gene interaction for consecutive time points. It relevance stems from the fact it is identifiable.

Table 2 depicts the performance of our method as the network increases. For a fixed $k$, we increase $p$ and monitor how best the network is recovered through true positive rates, (TPR), false positive rates, (FPR), and $F_1$ scores. We experience a relatively stable $F_1$ score and TPR but a decrease in FPR as the number of nodes increase. To this, our method performs quite well even on a large network.

### *4.2. In silico experiment: Arabidopsis thaliana clock*

The question we address here is to check whether the links recovered by our method are actually reproducible. To do that we want to validate the model of the Arabidopsis thaliana oscillator by Salome *et al.* (2004). The Arabidopsis is a model plant system that results from a combination of forward and reverse genetic approaches together with transcriptome-scale gene expression analyses. We consider a simple model of Arabidopsis clock made up of 9 genes namely CCA1, LHY, TOC1, ELF4, ELF3, GI, PRR9, PRR5, PRR3 with 3 replicates. Most importantly Salome *et al.* (2004) focus on the interaction between 4 out the 9 genes, namely LHY, CCA1, TOC1, GI. The gene *CCA1* has its corresponding protein named CIRCADIAN CLOCK ASSOCIATED 1. The gene *LHY* encodes a single Myb domain protein and is closely related to CCA1, both are important for proper clock function. Figure 4 depicts the genomic interaction in the Arabidopsis thaliana clock recovered by Salome *et al.* (2004). It reveals a regulation activities between CCA1 and LHY, this link came about

| | | True network | | |
|---|---|---|---|---|
| | | Links | No Links | Total |
| Estimated network | Links | 6 | 2 | 8 |
| | No Links | 2 | 2 | 4 |
| | Total | 8 | 4 | 12 |

**Table 3.** Comparison of a model of the Arabidopsis thaliana oscillator network (True network) and the network recovered by our method (Estimated network). The true network comprises of 8 links out of which our method recovered 6 correctly.

with the analysis of the TOC1 promoter and closed the loop of the Arabidopsis clock. The model from Salome *et al.* (2004) reveals that TOC1 acts as a positive regulator of the expression levels of CCA1 and LHY. The model also posits the repressive activity of CCA1 and LHY on TOC1. The gene GI is also necessary for high-level expression from the CCA1 and LHY.
In an attempt to recover the model of Arabidopsis thaliana clock by Salome *et al.* (2004), we applied our method to the data with 2 hidden or latent variables in the form of transcription factors. Our recovered network is shown in Figure 5.

For easy comparison purpose, we also focus on the interaction between two subnetworks representing the interaction between CCA1, TOC1, GI, LHY. The two subnetworks are subnetworks from Figure 4 and Figure 5. Both subnetworks support the hypothesis that TOC1 is a positive regulator of the expression level of CCA1. Our result also confirms the interaction between CCA1 and LHY. However our method has failed to recover the negative regulatory activity of CCA1 and LHY on TOC1. Table 3 compares the two networks in terms of how many correct links we have recovered.

## 5. Application

To demonstrate the application of our reverse engineering method, we used publicly available data, the results of two experiments used to investigate the expression response of human T-cells to PMA and ionomicin treatment. The data is a combination of two data set namely tcell.34 and tcell.10. The first data set tcell.34 contains the temporal expression levels of 58 genes for 10 unequally spaced time points. At each time point there are 34 separate measurements. The second data set tcell.10 comes from a related experiment considering the same genes and identical time points, and contains 10 further measurements per time point.

After pre-processing the data, genes found to have few interactions were eliminated leaving us a total of 45 genes. At each time point there are 44 separate measurements or replicates. It was assumed that the 44 replicates have a similar underlying distribution. See Rangel *et al.* (2004) for more details. Given that the t-cell is a time course gene expression data with technical replicates we expect more reliable estimation and inference results by applying our method. Corresponding to each gene expression $y_{tr}$, we also generated technical replicates for the hidden variables $\theta_{tr}$. In essence, we treated the data as a time series measurement

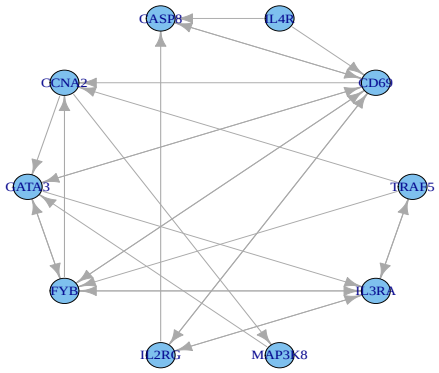


**Fig. 6.** Subnetwork found representing the topology of gene FYB in connection with some selected genes

$y_{t_r}$, $t = 1, 2, ..., 10$ and $r = 1, 2, ..., 44$. Based on our previous work Lotsi *et al.* (2016) and that of Rau *et al.* (2010), we assumed the dimension of the hidden state $k$, to be 4. For each replicate, $y_t$ and $\theta_t$ consist of 45 genes and 4 transcriptions factors respectively, each, measured at 10 different time points, i.e for each replicate $r$, $y_t$ and $\theta_t$ are of dimension $(45 \times 10)$, $(4 \times 10)$ respectively. Some of these genes include RB1, CCNG1, TRAF5, CLU.... We applied our $L_1$ penalized inference method to the time course gene expression data and estimated a total number of 2401 parameters consisting of $B$, $A$, $Z$ and $F$. To do that, we iterate across the penalty parameters namely 4 different tuning parameters $s_B$, $s_A$, $s_Z$ and $s_F$. While LARS produces the entire path of solutions, we make prediction or extract coefficients from the fitted LARS model using the `predict` function in LARS. The `predict` function allows one to extract a prediction at a particular point along the path. This procedure is repeated until convergence. We then have different set of estimated model parameters corresponding to each set of tuning parameters. At this stage, we applied model selection technique via minimum AICc described in section 3.4 to select the optimum parameters. At the end, we obtained the connectivity matrix of the directed genomic graph. The optimum estimated tuning parameters has given rise to fairly sparse networks.

The output are graphs showing connections from one gene expression variable at a given time point $t$ to another gene expression variable whose expression it influences at the next time point, $t+1$. The output depicted in Figure 6 is a sub-network that shows the topology of gene FYB. We found that the genes such as CCNA2, FYB, and CASP8 are mostly activation genes. Specifically, FYB activates the expression level of genes such as GATA3, CCNA2, CD69, IL3RA while CASP8 activates genes such as: JUND, CDC2, CD69. Figure 7 recovers the interactions between the Jun proteins family and other genes. It identifies JUND to have significant number of connections in the form of activation and inhibitions.

The structure of the network is visualized using the R package for Network analysis and visualization `igraph`.

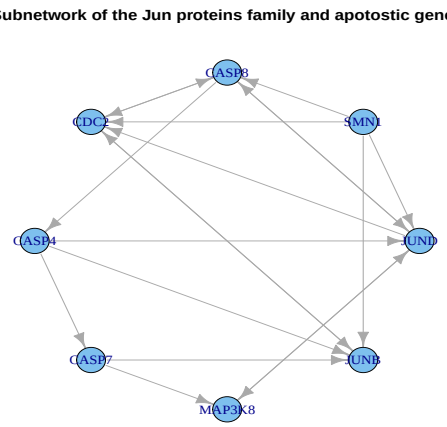


**Fig. 7.** Subnetwork found representing the interactions between Jun proteins family and other genes

Our method has resulted in relatively sparse networks as compared to Lotsi *et al.* (2016). In all, the following genes were found to have the highest number of interactions in terms of inwards directed connections: TRAF5, C3X1, CASP4, CDK4 and IL3RA. In addition, from a topological point of view genes such as JUND, AKT1, FYB, and CCNA2 occupy a crucial position in the recovered networks. We recommend these genes to be object of further study by biologist. These results supports the works of Rangel *et al.* (2004) and Wild *et al.* (2004). Both found gene FYB to occupy an important position in their respective graphs. At the optimum turning parameter, we found JUNB interacting directly with CASP4 through JUND; a result also supported by Beal *et al.* (2005). The unpenalized inference of Lotsi *et al.* (2016) approach has indicated that JUNB activates directly CDC2. This work also supports the same interaction. A portion of the subnetwork found by Beal *et al.* (2005) and Rau *et al.* (2010) representing the interactions between CASP4 and JUND is also found in our network through Figure 7. Another interesting interactions that were supported by previous literature were interactions between JUND and CASP7 on one hand and interactions between JUND and CDC2 both in the form of inhibitions. JUND is predicted to repress the expression level of the cell cycle regulator CDC2. This clearly supports the hypothesis that JUND negatively regulates cell growth and acts as anti-proliferative and anti-apoptotic signal gene.

Our work reveals that genes such as AKT1 and MLC1 also occupy a crucial position. AKT1 is found to influence the expression level of many genes. Of these, include the JUN proteins JUNB , one interleukin receptor gene, IL3RA, one apoptosis-related cysteine protease,

CASP4, the cell division cycle, CDC2. AKT1 is also seen to activate the expression level of a transcription factor. In our model, MLC1 is found to regulate positively the expression level of one of the transcription factors. Also MLC1 represses several genes including CASP7, CDK4, C3X1 to mention but few. Rau *et al.* (2010) found that CAPS4 inhibits the expression level of CAPS8. This interaction was supported by our work. Beal *et al.* (2005) did not recover such interaction. Also some findings, Figure 7 of our current study do not support the work of Beal *et al.* (2005) in the sense that we found no interactions between JUND and Caspase-7 and also between JUNB and MAPK8. Thus, the results based on our methodology suggest some findings that are supported by the current literature and are biologically interpretable, while some other findings have not been documented yet in biological literature and we hope these new findings will be confirmed in the near future. A comparison of our proposed model and method to alternative models and methods for dynamic network construction would be desirable, but is beyond the scope of this article.

## 6. Conclusion

In this paper, we have inferred a sparse dynamic network by using an input dependent linear state space model. We have assumed that the true biological process is not fully observed and the hidden variables were first calculated using a Kalman filtering and smoothing algorithm via an E-step. We then proceed to update the model parameters through an $L_1$ regularization constraint via LARS algorithm in the maximization step. We used AICc to determine the optimum combination of tuning parameters and hence the model parameters. The proposed method offers significant advantages over other methods that have recently appeared in the literature. For example, Beal *et al.* (2005) inferred regulatory interactions from expression data by maximizing the marginal likelihood with a modification of the EM algorithm. His approach was based on variational Bayesian methodology which is an approximation of the posterior distribution of the parameters while we did exact inference of the parameters. Rangel *et al.* (2004) used cross validation as model selection technique which is quite slow as compared to AIC. Also our model allows for dynamic correlation over time, as each observation and hidden state depend explicitly on some function of previous observations as opposed to the models described by Perrin *et al.* (2003); Wu *et al.* (2004). Their models do not allow for RNA-protein translation and RNA-RNA interactions through the matrix $A$ and $B$ respectively in our model. Most importantly, the LARS algorithm adopted guarantees us interpretable model, and accurate predictors.

One fundamental assumption in our proposed model is the first-order linear dynamics in the state and observation equations of the SSM. The advantages of using linear SSM stems from the fact that the linearity assumption has resulted in a more stable network and has enabled us to recover the dynamics of the network easily as compared to the nonlinear relationships. The inference method via LARS is potentially revolutionary, offering interpretable models, relative stability, accurate predictions, unbiased inferences, and a nice graphical display of coefficient paths that indicates the key tradeoff in model complexity. We used the R package for Network analysis and visualization `igraph` to display simple, and easy to understand graph through which the whole system under study can be ascertained quite easily.

Future works will encompass extending the linear SSM into a Non Linear State Space Model (NLSSM) (Quach *et al.*, 2007) whose hidden process will be defined through an integration of an Ordinary Differential Equation (ODE) and estimate both parameters and hidden variables through the same inference technique. We also plan to overcome the ODE limitations namely ability to handle noisy data and the high number of model parameters by integrating a sparse ODE model into a graphical model framework, thus taking noisy measurement into account, and the resulting model will then be embedded into a penalized maximum likelihood learning set-up.

## 7. Appendix

We outline here the proof of lemma 1

*Proof.* Properties of Gaussian distribution and Gaussian processes suggest that the quadratic $\mathbf{Q}(X)$ corresponds to a Gaussian $N(S^{-1}d, S^{-1})$. Therefore

$$\begin{aligned}
\mathbf{Q}(\beta) &= (\beta - S^{-1}d)^{'}\Sigma^{-1}(\beta - S^{-1}d) \\
&= (\beta - S^{-1}d)^{'}S(\beta - S^{-1}d) \\
&= (\beta - S^{-1}d)^{'}\mathbf{C}^{'}\mathbf{C}(\beta - S^{-1}d) \\
&= (\mathbf{C}S^{-1}d - \mathbf{C}\beta)^{'}(\mathbf{C}S^{-1}d - \mathbf{C}\beta) \\
&= (\mathbf{y} - \mathbf{C}\beta)^{'}(\mathbf{y} - \mathbf{C}\beta). \qquad (24)
\end{aligned}$$

Suppose we have a set of linearly independent covariates $(\mathbf{c}_1, ..., \mathbf{c}_k)$ and we define the matrix

$$\mathbf{C} = (..., s_i\mathbf{c}_i, ...)_{\mathbb{A}}, \qquad (25)$$

where $\mathbb{A}$ is the subset of the indices $\{1, ..., k\}$ of the active set and $s_i$ is the signs and equals $\pm 1$. Suppose

$$\mathbb{G}_{\mathbb{A}} = (\mathbf{C}_{\mathbb{A}}^{'}\mathbf{C}_{\mathbb{A}})^{-1} \quad \text{and} \quad A_{\mathbb{A}} = \left[1_{\mathbb{A}}^{'}\mathbb{G}_{\mathbb{A}}1_{\mathbb{A}}\right]^{-\frac{1}{2}} \qquad (26)$$

where $1_{\mathbb{A}}$ is a vector of 1's of length equals the size of $\mathbb{A}$.

The unit vector making equal angles, less than $90^0$, with the active columns of $\mathbf{C}_{\mathbb{A}}$ is given by

$$w_{\mathbb{A}} = A_{\mathbb{A}}\mathbb{G}_{\mathbb{A}} \qquad (27)$$

Let

$$\hat{\gamma} = \underset{i \in \mathbb{A}^c}{min^{+}} \left\{ \frac{(Q_{max} - \hat{q}_i)}{A_{\mathbb{A}} - a_i}, \frac{(Q_{max} + \hat{q}_i)}{A_{\mathbb{A}} + a_i} \right\}, \qquad (28)$$

where $\mathbf{q} = \mathbf{C}^{'}\mathbf{y}$, representing vector of current correlations, $Q_{max}$ is the maximum absolute value from the set $\mathbf{q}$, $\mathbf{a} = (\mathbf{C}_{\mathbb{A}^c})^{'}\mathbf{C}_{\mathbb{A}}w_{\mathbb{A}}$.
Now the next step of LARS algorithm updates the coefficient $\hat{\beta}_{k-1}$, say to

$$\hat{\beta}_k = \hat{\beta}_{k-1} + \hat{\gamma} w_{\mathbb{A}} \tag{29}$$

We need to show that $\hat{\beta}_k$ is a function of $\mathbf{C}^{'}\mathbf{C}$ and $\mathbf{C}^{'}y$.
The above definitions of $\mathbf{q}$ and $\mathbf{a}$ indicate that $\hat{\gamma}$ is a function of $\mathbf{C}^{'}\mathbf{C}$ and $\mathbf{C}^{'}y$. Therefore $\hat{\beta}_k$ is also a function of $\mathbf{C}^{'}\mathbf{C}$ and $\mathbf{C}^{'}y$

## References


Bower, M., and Bolouri, H.(2001).*Computational Modeling of Genetic and Biochemical Networks*

Kauffman, S.A.(1993).*The origins of order*. J. Theor. Biol, 22:437.

Dhaeseleer, P., Liang, S., and Somogyi, R.(2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*.16,18, p.707-726.

Schafer, J., and Strimmer, K.(2005). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*.21,6, p.754-764.

Abegaz, F. and Wit, E.(2013). Sparse time series chain graphical models for reconstructing genetic networks. *Journal of Biostatistics*.

Perrin, B.E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J. and d'Alche Buc, F.(2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics 19 (Suppl.2)*. p.138–148.

Fujita, A. Sato, J. Garay-Malpartida, H., Yamaguchi, R., Miyano, S., Sogayar, M., and Ferreira, C.(2007). Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Systems Biology*. 1, 1 p.39.

Quach, M., Brunel, N., and d'Alché-Buc., F.(2007). Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference. *Bioinformatics*. 23, 23, p.3209-3216

Cao, J., and Zhao, H.(2008). Estimating dynamic models for gene regulation networks. *Bioinformatics*. 24, 14 p.1619-1624

Khanin, R., Vinciotti, V., and Wit, E.(2006). Reconstructing repressor protein levels from expression of gene targets in Escherichia coli. *Proceedings of the National Academy of Sciences*. 103, 49, p.18592-18596

Khanin, R., Vinciotti, V., Mersinias, V., Smith, C. P. and Wit, E.(2007). Statistical Reconstruction of Transcription Factor Activity Using Michaelis-Menten Kinetics. *Biometrics*. 63, 3, p.816–823

Chen, Kuang-Chi., Wang, Tse-Yi., Tseng, Huei-Hun., Huang, Chi-Ying, F., and Kao, Cheng-Yan.(2005). A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae. *Bioinformatics*. 21, 12, p.2883–2890

Purutcuoglu, V. and Wit, E.(2008). Bayesian inference for the MAPK/ERK pathway by considering the dependency of the kinetic parameters. *Bayesian analysis 2008*. 3, 4, p.851-886

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., Wild, D. L., and Falciani, F.(2004). Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics*. 9, 20.

Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C., and Wild, D. L.(2005). A Bayesian Approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*. 21, p.349-356

Ramsay, J. O., Hooker, G., Campbell, D., Cao, J.(2007). Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 69, 5. p.1467-98686

Fahrmeir, L. and Kunstler, R.(2009). Penalized likelihood smoothing in robust state space models. *Biometrika (1999)*. 49, 5. p.173–191

Fahrmeir, L. and Wagenpfeil,S.(1997). Penalized likelihood estimation and iterative Kalman smoothing for non-Gaussian dynamic regression models. *Computational Statistics & Data analysis (1997)*. 24. p.295–320

Durbin, J., and Koopman, S. J.(1997). *Time series analysis by state space models*. Oxford University Press.

Shumway, R. H., and Stoffer, D. S.(2005). *Time series analysis and its applications with R examples*.

Meinhold, R. J., and Singpurwalla, N. D. (1983). Understanding the Kalman Filter. *The Americam Statistician*. 37, N0. 2. p.123-127

Fang-Xiang, Wu., Wen-Jun, Zhang., and Kusalik, J.A.(2004). Modelling Gene expression from microarray expression data with state-space equations. *Biocomputing*. 9. p.588–592

Bremer, M. and Doerge, R. W. (2009). The KM-algorithm identifies regulated genes in time series expression data.*Advances in Bioinformatics (in press)*

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso.*Journal of the Royal Statistics Society*, series B, 58. p. 267-288

Zou, H. and Hastie, T. (2005).Regularization and variable selection via the elastic net.*Journal of the Royal Statistical Society*, Series B, 67(2). p. 301-320

Hastie, S. Rosset., Tibshirani, R. and Zhu, J. (2004). The entire regularization for the support vector machine.*Journal of Machine Learning Research*, 8. p. 1519-1555

Park, M. Y. and Hastie, T.(2007). $l_1$ regularization path algorithm for generalized linear models.*Journal of the Royal Statistical Society*, Series B,69. p.659-677

Friedman, T., Hastie, T. and Tibshirani, R. (2008).Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9. p.432-441

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Annals of statistics*, 32. p.407-451.

Fan, J., and Li, R. (2001).Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, 96, 456.

Stoffer, D.S., and Shumway, R.H. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *J.Time series Analysis*. 3, p. 253-264

Shumway, R.H. (2000).Dynamic mixed models for irregularly observed time series. *Resenhas-Reviews of the Institute of Mathematics and Statistics,University of Sao Paulo,USP Press, Brazil*. 4,No.4, p.433-456.

Green, P. J. (1990).On Use of the EM for Penalized Likelihood Estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*. 52,3, p.443-452

Briers, Mark., Doucet, Arnaud., and Maskell, Simon. (2010). Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*. 62,1, p.61-89

Schwarz, G. Estimating the dimension of a model. (1978).*The Annals of Statistics*. 6(2), p.461-464.

Burnham, K. P., and Anderson, D. R. (2002).*Model Selection and Multi Model Inference*. Annals of the Institute of Statistical Mathematics. 2.

Mallows, C. L. Some Comments on CP. (1973).*Technometrics*. 15,4. p.661-675.

Akaike, H. A. (1974). New look at the statistical model identification. *Automatic Control, IEEE Transactions on*. 19,6. p.16 - 723.

Wild, D. L., Rangel, C., Angus, J., and Ghahramani, Z. (2004). Modeling Genetic Regulatory Networks using Gene Expression Profiling and State Space Models. *Probabilistic Modelling in Bioinformatics and Medical informatics. Springer-Verlag (in press)*.

Salome, Patrice. A., and McClung, Robertson. C. (2004). The Arabidopsis thaliana clock. *Journal of Biological Rhythms*. 17 (4). p.1-18.

Lotsi, A. and Wit, E. (2016). State space Modelling of Replicated Dynamic Genetic Networks. *British Journal of Applied Science and Technology*.

Rau, Andrea., Foulley, Jean-Louis., Jaffrézic, Florence., and Doerge, Rebecca, W. (2010). An Empirical Bayesian Method for Estimating Biological Networks from Temporal Microarray Data. *Statistical Applications in Genetics and Molecular Biology*. 9, Issue 1 2010.

Fu, Wenjiang. J. (1998). Penalized Regressions: The Bridge versus the Lasso. *Journal of Computational and Graphical Statistics*. 7, 3, p.397-416.