

RESEARCH

Open Access

Gene regulatory network inference by point-based Gaussian approximation filters incorporating the prior information

Bin Jia¹ and Xiaodong Wang^{2*}

Abstract

The extended Kalman filter (EKF) has been applied to inferring gene regulatory networks. However, it is well known that the EKF becomes less accurate when the system exhibits high nonlinearity. In addition, certain prior information about the gene regulatory network exists in practice, and no systematic approach has been developed to incorporate such prior information into the Kalman-type filter for inferring the structure of the gene regulatory network. In this paper, an inference framework based on point-based Gaussian approximation filters that can exploit the prior information is developed to solve the gene regulatory network inference problem. Different point-based Gaussian approximation filters, including the unscented Kalman filter (UKF), the third-degree cubature Kalman filter (CKF₃), and the fifth-degree cubature Kalman filter (CKF₅) are employed. Several types of network prior information, including the existing network structure information, sparsity assumption, and the range constraint of parameters, are considered, and the corresponding filters incorporating the prior information are developed. Experiments on a synthetic network of eight genes and the yeast protein synthesis network of five genes are carried out to demonstrate the performance of the proposed framework. The results show that the proposed methods provide more accurate inference results than existing methods, such as the EKF and the traditional UKF.

Keywords: Gene regulatory network; Point-based Gaussian approximation filters; Network prior information; Sparsity; Iterative thresholding

1 Introduction

Inferring gene regulatory network (GRN) has become one of the most important missions in system biology. Genome-wide expression data is widely used due to the development of several high-throughput experimental technologies. The gene regulatory network can be inferred from a number of gene expression samples taken over a period of time. Modeling of GRN is required before its structure can be inferred. Common dynamical modeling methods of GRN include Bayesian networks [1], Boolean networks [2], ordinary differential equations [3], state-space models [4,5], and so on. Various approaches based on different models have been used to infer the network from observed gene expression data, such as the

Markov Chain Monte Carlo (MCMC) methods for the dynamic Bayesian network model [6] and the ordinary differential equation model [7], as well as the Kalman filtering methods for the state-space model [4,8] and the ordinary differential equation model [3]. Some survey papers can be found in [9-12].

Due to the 'stochastic' nature of the gene expression, the Kalman filtering approach based on the state-space model is one of the most competitive methods for inferring the GRN. The Kalman filter is optimal for linear Gaussian systems. However, the GRN is generally highly nonlinear. Hence, advanced filtering methods for nonlinear dynamic systems should be considered. The extended Kalman filter (EKF) is probably the most widely used nonlinear filter which uses the first-order Taylor series expansion to linearize the nonlinear model. However, the accuracy of the EKF is low when the system is highly nonlinear or contains large uncertainty. The point-based

*Correspondence: wangx@ee.columbia.edu

²Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

Full list of author information is available at the end of the article

Gaussian approximation filters have been recently proposed to improve the performance of the EKF, which employ various quadrature rules to compute the integrals involved in the exact Bayesian estimation. Many filters fall into this category, such as the unscented Kalman filter (UKF) [13], the Gauss-Hermite quadrature filter [14], the cubature Kalman filter (CKF) [15], and the sparse-grid quadrature filter [16]. Besides the point-based Gaussian approximation filters, the particle filter has drawn much attention recently [17]. The particle filter uses random particles with weights to represent the probability density function (pdf) in the Bayesian estimation and provides better estimation result than the EKF. The main problem of the particle filter is that the computational complexity is high, and therefore, it is hard to use for high-dimensional problems, such as the problem considered in this paper.

The EKF and the particle filter have been used for the inference of GRN [4,8,18]. In this paper, we consider the point-based Gaussian approximation filters. Our main objective is to provide a framework of incorporating network prior information into the filters. For example, some gene regulations may be known [19] from literature and the inference accuracy of GRN can be improved by incorporating the known regulations of the GRN [20]. Integration of the prior knowledge or constraints with the GRN inference algorithm has been introduced to improve the inference result. The DNA motif sequence in gene promoter regions is incorporated in [21] while modeling of transcription factor interactions is incorporated in [22]. As mentioned in [20], experimentally determined physical interactions can be obtained. In addition, the sparsity constraint is frequently used in the inference of the GRN. To the best of the authors' knowledge, the most related work in incorporating the prior information in Bayesian filters is [8]. In that work, rather than directly getting the inference results from the filter, an optimization method is used. In particular, a cost function is used in which the sparsity constraint is enforced. However, the cost function in [8] does not consider the uncertainty of the state in the filtering. That cost function in fact is not coupled well with the filtering algorithm. In addition, it did not consider other kinds of prior information. In this paper, we propose a new framework that incorporates the prior information effectively in the filtering algorithm by solving a constrained optimization problem. Efficient recursive algorithms are provided to solve the associated optimization problem.

The remainder of this paper is organized as follows. In Section 2, the modeling of gene regulatory network is introduced. The point-based Gaussian approximation filters are briefly introduced in Section 3. The proposed new filtering framework is described in Section 4. In Section 5, experimental results are provided. Finally, concluding remarks are given in Section 6.

2 State-space modeling of gene regulatory network

The GRN can be described by a graph in which genes are viewed as nodes and edges depict causal relations between genes. The structure of GRN reveals the mechanisms of biological cells. Analyzing the structure of GRN will pave the way for curing various diseases [23]. The learning of GRN has drawn much attention recently due to the availability of the microarray data. By analyzing collected gene expression levels over a period of time, one can identify various regulatory relations between different genes. To facilitate the analysis of the GRN, modeling of GRN is required. Different models can be used, such as Bayesian networks [1], Boolean networks [2], ordinary differential equation [3], and state-space model [4,5]. The state-space model has been widely used because it incorporates noise and can make use of computationally efficient filtering algorithms [5]. Thus, we also use the state-space modeling of GRN in this paper.

Under the discrete-time state-space modeling of the gene regulatory networks, the network evolution from time k to time $k - 1$ can be described by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_k, \quad (1)$$

where $\mathbf{x}_k = [x_{1,k}, \dots, x_{n,k}]^T$ is the state vector and $x_{i,k}$ denotes the gene expression level of the i -th gene at time k . \mathbf{f} is a nonlinear function that characterizes the regulatory relationship among the genes. \mathbf{v}_k is the state noise and it is assumed to follow a Gaussian distribution with mean $\mathbf{0}$ and covariance matrix \mathbf{Q}_k , i.e., $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$.

Following [8], we use the following nonlinear function in the state Equation (1):

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{g}(\mathbf{x}), \quad (2)$$

with

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(x_1) \\ \vdots \\ g_n(x_n) \end{bmatrix} \quad (3)$$

and

$$g_i(x) = \frac{1}{1 + e^{-\mu_i x}}. \quad (4)$$

In (2), \mathbf{A} is the regulatory coefficient matrix with a_{ij} denoting the regulation coefficient from gene j to gene i . Note that a positive coefficient a_{ij} indicates that gene j activates gene i and a negative a_{ij} indicates that gene j represses gene i . In (4), μ_i is a parameter. Note that \mathbf{A} and μ_i are unknown parameters. The discrete-time nonlinear stochastic dynamic system [24] shown in Eqs. (1)-(3) have been successfully used to describe the GRN [4,8]. Equation (4) is also called Sigmoid function which is frequently used since it is consistent with the fact that all concentrations get saturated at some point in time [25].

The Sigmoid function has been used in modeling GRN to verify various methods, such as artificial neural network [26], simulated annealing and clustering algorithm [27], extended Kalman filter [4], particle filter [8], and Genetic programming and Kalman filtering [25].

For the measurement model, we consider the following general nonlinear observation equation

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k, \quad (5)$$

where $\mathbf{h}(\cdot)$ is some nonlinear function, \mathbf{n}_k is the measurement noise, which is assumed to follow the Gaussian distribution with mean $\mathbf{0}$ and covariance matrix \mathbf{R}_k , i.e., $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. For example, if the noise corrupted expression levels are observed, then $\mathbf{h}(\mathbf{x}) = \mathbf{x}$.

3 Network inference using point-based Gaussian approximation filters

3.1 Gaussian approximation filters

In this section, the framework of point-based Gaussian approximation filters for the state-space dynamic model is briefly reviewed. We consider the state-space model consisting of the state Equation (1) and the measurement Equation (5). We denote $\mathbf{y}^k \triangleq [\mathbf{y}_1, \dots, \mathbf{y}_k]$.

The optimal Bayesian filter is composed of two steps: prediction and filtering. Specifically, given the prior pdf $p(\mathbf{x}_{k-1}|\mathbf{y}^{k-1})$ at time $k-1$, the predicted conditional pdf $p(\mathbf{x}_k|\mathbf{y}^{k-1})$ is given by

$$p(\mathbf{x}_k|\mathbf{y}^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}^{k-1})d\mathbf{x}_{k-1}. \quad (6)$$

After the measurement at time k becomes available, the filtered pdf is given by

$$p(\mathbf{x}_k|\mathbf{y}^k) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}^{k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}^{k-1})d\mathbf{x}_k}. \quad (7)$$

The pdf recursions in (6) and (7) are in general computationally intractable unless the system is linear and Gaussian. The Gaussian approximation filters approximate (6) and (7) by invoking Gaussian assumptions. Specifically, the first assumption is that given \mathbf{y}^{k-1} , \mathbf{x}_{k-1} has a Gaussian distribution, i.e., $\mathbf{x}_{k-1}|\mathbf{y}^{k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$. The second assumption is that $(\mathbf{x}_k, \mathbf{y}_k)$ are jointly Gaussian given \mathbf{y}^{k-1} .

It then follows from the second assumption that given \mathbf{y}^{k-1} , \mathbf{x}_k has a Gaussian distribution, i.e., $\mathbf{x}_k|\mathbf{y}^{k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$. Using (1) and the first assumption, we have the predicted mean $\hat{\mathbf{x}}_{k|k-1}$ and covariance $\mathbf{P}_{k|k-1}$ given respectively by

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &\triangleq \mathbb{E}\{\mathbf{x}_k|\mathbf{y}^{k-1}\} = \mathbb{E}_{\mathbf{x}_{k-1}|\mathbf{y}^{k-1}}\{\mathbf{f}(\mathbf{x}_{k-1})\} \\ &= \int \mathbf{f}(\mathbf{x})\phi(\mathbf{x}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})d\mathbf{x}, \end{aligned} \quad (8)$$

and

$$\begin{aligned} \mathbf{P}_{k|k-1} &\triangleq \text{Cov}\{\mathbf{x}_k|\mathbf{y}^{k-1}\} \\ &= \mathbb{E}_{\mathbf{x}_{k-1}|\mathbf{y}^{k-1}}\left\{(\mathbf{f}(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1})\right. \\ &\quad \left. \times (\mathbf{f}(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1})^T\right\} + \mathbf{Q}_{k-1} \\ &= \int (\mathbf{f}(\mathbf{x}) - \hat{\mathbf{x}}_{k|k-1}) \\ &\quad \times (\mathbf{f}(\mathbf{x}) - \hat{\mathbf{x}}_{k|k-1})^T \phi(\mathbf{x}; \hat{\mathbf{x}}_{k-1|k-1}, \\ &\quad \mathbf{P}_{k-1|k-1})d\mathbf{x} + \mathbf{Q}_{k-1}, \end{aligned} \quad (9)$$

where $\phi(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})$ denotes the multivariate Gaussian pdf with mean $\hat{\mathbf{x}}$ and covariance \mathbf{P} .

Then, following the second assumption, given $\mathbf{y}^k = [\mathbf{y}^{k-1}, \mathbf{y}_k]$, \mathbf{x}_k is Gaussian distributed, i.e., $\mathbf{x}_k|\mathbf{y}^k \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$. Using the conditional property of the multivariate Gaussian distribution, the filtered mean $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ are given respectively by

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &\triangleq \mathbb{E}\{\mathbf{x}_k|\mathbf{y}_k, \mathbf{y}^{k-1}\} \\ &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \end{aligned} \quad (10)$$

$$\begin{aligned} \text{and } \mathbf{P}_{k|k} &\triangleq \text{Cov}\{\mathbf{x}_k|\mathbf{y}_k, \mathbf{y}^{k-1}\} \\ &= \mathbf{P}_{k|k-1} - \mathbf{L}_k\mathbf{P}_k^{xy}, \end{aligned} \quad (11)$$

with

$$\begin{aligned} \hat{\mathbf{y}}_{k|k-1} &= \mathbb{E}_{\mathbf{x}_k|\mathbf{y}^{k-1}}\{\mathbf{h}(\mathbf{x}_k)\} \\ &= \int \mathbf{h}(\mathbf{x})\phi(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}, \end{aligned} \quad (12)$$

$$\mathbf{L}_k = \mathbf{P}_k^{xy}(\mathbf{R}_k + \mathbf{P}_k^{yy})^{-1}, \quad (13)$$

$$\begin{aligned} \mathbf{P}_k^{xy} &= \mathbb{E}_{\mathbf{x}_k|\mathbf{y}^{k-1}}\left\{(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})(\mathbf{h}(\mathbf{x}) - \hat{\mathbf{y}}_{k|k-1})^T\right\} \\ &= \int (\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})(\mathbf{h}(\mathbf{x}) - \hat{\mathbf{y}}_{k|k-1})^T \\ &\quad \phi(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}, \end{aligned} \quad (14)$$

$$\begin{aligned} \mathbf{P}_k^{yy} &= \mathbb{E}_{\mathbf{x}_k|\mathbf{y}^{k-1}}\left\{(\mathbf{h}(\mathbf{x}) - \hat{\mathbf{y}}_{k|k-1})(\mathbf{h}(\mathbf{x}) - \hat{\mathbf{y}}_{k|k-1})^T\right\} \\ &= \int (\mathbf{h}(\mathbf{x}) - \hat{\mathbf{y}}_{k|k-1})(\mathbf{h}(\mathbf{x}) - \hat{\mathbf{y}}_{k|k-1})^T \\ &\quad \phi(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}. \end{aligned} \quad (15)$$

3.2 Point-based Gaussian approximation filters

The integrals in (8), (9), (12), (14) and (15) are Gaussian type that can be efficiently approximated by various quadrature methods. Specifically, if a set of weighted points $\{(\mathbf{y}_i, w_i), i = 1, \dots, N\}$ can be used to approximate the integral

$$\int \mathbf{h}(\mathbf{x})\phi(\mathbf{x}; \mathbf{0}, \mathbf{I})d\mathbf{x} \approx \sum_{i=1}^N w_i \mathbf{h}(\mathbf{y}_i), \quad (16)$$

then the general Gaussian-type integral can be approximated by

$$\int \mathbf{h}(\mathbf{x})\phi(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) \, d\mathbf{x} \approx \sum_{i=1}^N w_i \mathbf{h}(\mathbf{S}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}), \quad (17)$$

where $\mathbf{P} = \mathbf{S}\mathbf{S}^T$ and \mathbf{S} can be obtained by Cholesky decomposition or singular value decomposition (SVD).

Using (17), we can then approximate (8) and (9) as follows:

$$\hat{\mathbf{x}}_{k|k-1} \approx \sum_{i=1}^N w_i \mathbf{f}(\boldsymbol{\xi}_{k-1,i}) \quad (18)$$

and

$$\begin{aligned} \mathbf{P}_{k|k-1} &\approx \sum_{i=1}^N w_i \mathbf{f}(\boldsymbol{\xi}_{k-1,i} - \hat{\mathbf{x}}_{k|k-1}) \\ &\quad \times (\boldsymbol{\xi}_{k-1,i} - \hat{\mathbf{x}}_{k|k-1})^T + \mathbf{Q}_{k-1}, \end{aligned} \quad (19)$$

where $\boldsymbol{\xi}_{k-1,i}$ is the transformed quadrature point obtained from the covariance decomposition, i.e.,

$$\mathbf{P}_{k-1|k-1} = \mathbf{S}_{k-1}\mathbf{S}_{k-1}^T, \quad (20)$$

$$\boldsymbol{\xi}_{k-1,i} = \mathbf{S}_{k-1}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}_{k-1|k-1}. \quad (21)$$

Similarly, we can approximate (12), (14) and (15) as follows:

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=1}^N w_i \mathbf{h}(\tilde{\boldsymbol{\xi}}_{k,i}), \quad (22)$$

$$\mathbf{P}_k^{xy} = \sum_{i=1}^N w_i (\tilde{\boldsymbol{\xi}}_{k,i} - \hat{\mathbf{x}}_{k|k-1}) (\mathbf{h}(\tilde{\boldsymbol{\xi}}_{k,i}) - \hat{\mathbf{y}}_{k|k-1})^T, \quad (23)$$

$$\mathbf{P}_k^{yy} = \sum_{i=1}^N w_i (\mathbf{h}(\tilde{\boldsymbol{\xi}}_{k,i}) - \hat{\mathbf{y}}_{k|k-1}) (\mathbf{h}(\tilde{\boldsymbol{\xi}}_{k,i}) - \hat{\mathbf{y}}_{k|k-1})^T, \quad (24)$$

where $\tilde{\boldsymbol{\xi}}_{k,i}$ is the transformed point obtained from the decomposition of the predicted covariance, i.e.,

$$\mathbf{P}_{k|k-1} = \tilde{\mathbf{S}}_k \tilde{\mathbf{S}}_k^T, \quad (25)$$

$$\tilde{\boldsymbol{\xi}}_{k,i} = \tilde{\mathbf{S}}_k \boldsymbol{\gamma}_i + \hat{\mathbf{x}}_{k|k-1}. \quad (26)$$

Various numerical rules can be used to form the approximation in (16), which lead to different Gaussian approximation filters. In particular, the unscented transformation, the Gauss-Hermite quadrature rule, and the sparse-grid quadrature rules are used in the unscented Kalman filter (UKF), the Gauss-Hermite quadrature Kalman filter (GHQF), and the sparse-grid quadrature filter (SGQF), respectively.

Recently, the fifth-degree quadrature filter has been proposed and shown to be more accurate than the third-degree quadrature filters, such as the UKF and the third-degree cubature Kalman filter (CKF₃), when the system is highly nonlinear or contains large uncertainty [16]. In this paper, we consider the UKF, CKF₃, and the fifth-degree

cubature Kalman filter (CKF₅). Other filters such as the central difference filter [14] and divided difference filter [28] can also be used. The CKF₅ is based on Mysovskikh's method which uses fewer points than the fifth-degree quadrature filter in [16]. In the following, different numerical rules used in (16) are briefly summarized.

3.2.1 Unscented transform

In the unscented Kalman filter (UKF), we have $N = 2n + 1$ where n is the dimension of \mathbf{x} . The quadrature points and the corresponding weights are given respectively by

$$\boldsymbol{\gamma}_i = \begin{cases} \mathbf{0}, & i = 1, \\ \sqrt{(n + \kappa)}\mathbf{e}_{i-1}, & i = 2, \dots, n + 1, \\ -\sqrt{(n + \kappa)}\mathbf{e}_{i-n-1}, & i = n + 2, \dots, 2n + 1, \end{cases} \quad (27)$$

and

$$w_i = \begin{cases} \frac{\kappa}{n + \kappa}, & i = 1, \\ \frac{1}{2(n + \kappa)}, & i = 2, \dots, 2n + 1, \end{cases} \quad (28)$$

where κ is a tunable parameter, and \mathbf{e}_i is the i -th n -dimensional unit vector in which the i -th element is 1 and other elements are 0.

3.2.2 Cubature rules

The left-hand side of (16) can be rewritten as

$$\int \mathbf{h}(\mathbf{x}) \phi(\mathbf{x}; \mathbf{0}, \mathbf{I}) \, d\mathbf{x} = \frac{1}{\pi^{n/2}} \int \mathbf{h}(\sqrt{2}\mathbf{x}) \exp(-\mathbf{x}^T \mathbf{x}) \, d\mathbf{x}. \quad (29)$$

Consider the integral

$$I(\mathbf{h}) = \int \mathbf{h}(\mathbf{x}) \exp(-\mathbf{x}^T \mathbf{x}) \, d\mathbf{x}. \quad (30)$$

By letting $\mathbf{x} = r\mathbf{s}$ with $\mathbf{s}^T \mathbf{s} = 1$ and $r = \sqrt{\mathbf{x}^T \mathbf{x}}$, $I(\mathbf{h})$ can be rewritten in the spherical-radial coordinate system as

$$I(\mathbf{h}) = \int_0^\infty \int_{U_n} \mathbf{h}(r\mathbf{s}) r^{n-1} \exp(-r^2) \, d\sigma(\mathbf{s}) \, dr, \quad (31)$$

where $U_n = \{\mathbf{s} \in \mathbf{R}^n : \|\mathbf{s}\| = 1\}$, and $\sigma(\cdot)$ is the spherical surface measure or the area element on U_n .

Note that (31) contains two types of integrals: the radial integral $\int_0^\infty h_r(r) r^{n-1} \exp(-r^2) \, dr$ and the spherical integral $\int_{U_n} \mathbf{h}_s(\mathbf{s}) \, d\sigma(\mathbf{s})$.

If the radial rule can be approximated by

$$\int_0^\infty h_r(r) r^{n-1} \exp(-r^2) \, dr \approx \sum_{i=1}^{N_r} w_{r,i} h_r(r_i), \quad (32)$$

and the spherical integral can be approximated by

$$\int_{U_n} \mathbf{h}_s(\mathbf{s}) \, d\sigma(\mathbf{s}) \approx \sum_{j=1}^{N_s} w_{s,j} \mathbf{h}_s(\mathbf{s}_j), \quad (33)$$

then (31) can be approximated by

$$I(\mathbf{h}) \approx \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} w_{r,i} w_{s,j} \mathbf{h}(r_i; s_j). \quad (34)$$

A third-degree cubature rule to approximate (29) is obtained by using the third-degree spherical rule and radial rule [15]:

$$\int \mathbf{h}(\mathbf{x}) \phi(\mathbf{x}; \mathbf{0}, \mathbf{I}) d\mathbf{x} \approx \frac{1}{2n} \sum_{i=1}^n [\mathbf{h}(\sqrt{n}\mathbf{e}_i) + \mathbf{h}(-\sqrt{n}\mathbf{e}_i)]. \quad (35)$$

Remark: The third-degree cubature rule is identical to the unscented transformation with $\kappa = 0$.

To construct the fifth-degree cubature rule, the Mysovskikh's method [29] and the moment matching method [16] are used to provide the fifth-degree spherical rule and radial rule, respectively. The final fifth-degree cubature rule is given by

$$\begin{aligned} \int \mathbf{h}(\mathbf{x}) \phi(\mathbf{x}; \mathbf{0}, \mathbf{I}) d\mathbf{x} \approx & \frac{2}{n+2} \mathbf{h}(\mathbf{0}) + \\ & + \frac{n^2(7-n)}{2(n+1)^2(n+2)^2} \sum_{i=1}^{n+1} [\mathbf{h}(\sqrt{n+2}\mathbf{s}_1^{(i)}) \\ & + \mathbf{h}(-\sqrt{n+2}\mathbf{s}_1^{(i)})] \\ & + \frac{2(n-1)^2}{(n+1)^2(n+2)^2} \sum_{i=1}^{n(n+1)/2} [\mathbf{h}(\sqrt{n+2}\mathbf{s}_2^{(i)}) \\ & + \mathbf{h}(-\sqrt{n+2}\mathbf{s}_2^{(i)})], \end{aligned} \quad (36)$$

where the point $\mathbf{s}_1^{(i)}$ is given by

$$\mathbf{s}_1^{(i)} = [p_1^{(i)}, p_2^{(i)}, \dots, p_n^{(i)}], \quad i = 1, 2, \dots, n+1, \quad (37)$$

with

$$p_j^{(i)} = \begin{cases} -\sqrt{\frac{n+1}{n(n-j+2)(n-j+1)}}, & j < i, \\ \sqrt{\frac{(n+1)(n-i+1)}{n(n-i+2)}}, & j = i, \\ 0, & j > i. \end{cases} \quad (38)$$

Moreover, the set of points $\{\mathbf{s}_2^{(i)}\}$ is given by

$$\begin{aligned} \{\mathbf{s}_2^{(i)}\} = & \left\{ \sqrt{\frac{n}{2(n-1)}} (\mathbf{s}_1^{(k)} + \mathbf{s}_1^{(l)}) : k < l, k, l = 1, \right. \\ & \left. 2, \dots, n+1 \right\}. \end{aligned} \quad (39)$$

3.3 Augmented state-space model for network inference

In the state-space model for gene regulatory networks described in Section 3.2, the underlying network structure is characterized by the $n \times n$ regulatory coefficient matrix \mathbf{A} in (2) and the parameters $\boldsymbol{\mu} = [\mu_1, \dots, \mu_n]$ in (4). The problem of network inference then becomes to estimate \mathbf{A} and $\boldsymbol{\mu}$. To do that, we incorporate the unknown parameters \mathbf{A} and $\boldsymbol{\mu}$ into the state vector to obtain an augmented state-space model, and then apply the point-based Gaussian approximation filters to estimate the space vector and thereby obtaining the estimates of \mathbf{A} and $\boldsymbol{\mu}$.

Specifically, we denote $\boldsymbol{\theta} = [a_{11}, a_{12}, \dots, a_{1n}, \dots, a_{nm}, \mu_1, \dots, \mu_n]^T$ and the augmented state vector $\bar{\mathbf{x}}_k = [\mathbf{x}_k^T, \boldsymbol{\theta}^T]^T$. Then, the augmented state equation can be written as

$$\bar{\mathbf{x}}_k = \bar{\mathbf{f}}(\bar{\mathbf{x}}_{k-1}) + \bar{\mathbf{v}}_k = \begin{bmatrix} \mathbf{A}_{k-1} \mathbf{g}_{k-1}(\mathbf{x}_{k-1}) \\ \boldsymbol{\theta}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{k-1} \\ \mathbf{0} \end{bmatrix}. \quad (40)$$

Note that \mathbf{A}_{k-1} and \mathbf{g}_{k-1} can be obtained from $\boldsymbol{\theta}_{k-1}$, and $\bar{\mathbf{v}}_k \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{Q}}_k)$ with $\bar{\mathbf{Q}}_k = \text{diag}([\mathbf{Q}_k \quad \mathbf{O}_{n^2+n}])$, where \mathbf{O}_m denotes an $m \times m$ all-zero matrix.

In the remainder of this paper, we assume that the noisy gene expression levels are observed. Therefore, the augmented measurement equation becomes

$$\mathbf{y}_k = \mathbf{h}(\bar{\mathbf{x}}_k) + \mathbf{n}_k = \mathbf{B}\bar{\mathbf{x}}_k + \mathbf{n}_k, \quad (41)$$

where $\mathbf{B} = [\mathbf{I}_n, \mathbf{O}_{n \times (n^2+n)}]$, $\mathbf{O}_{n \times (n^2+n)}$ denotes an $n \times (n^2+n)$ all zeros matrix.

The point-based Gaussian approximation filters can then be used to obtain the estimate of the augmented state, $\hat{\bar{\mathbf{x}}}_k$, from which the estimates of the unknown network parameters, i.e., $\hat{\mathbf{A}}$ and $\hat{\boldsymbol{\mu}}$ can then be obtained.

Note that since the measurement Equation (41) is linear, the filtering Equations (10, 11) become

$$\hat{\bar{\mathbf{x}}}_{k|k} = \hat{\bar{\mathbf{x}}}_{k|k-1} + \mathbf{L}_k(\mathbf{y}_k - \mathbf{B}\hat{\bar{\mathbf{x}}}_{k|k-1}), \quad (42)$$

$$\text{and } \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{L}_k \mathbf{B} \mathbf{P}_{k|k-1}, \quad (43)$$

$$\text{with } \mathbf{L}_k = \mathbf{P}_{k|k-1} \mathbf{B}^T (\mathbf{R}_k + \mathbf{B} \mathbf{P}_{k|k-1} \mathbf{B}^T)^{-1}, \quad (44)$$

which are the same as the filtering updates for Kalman filters.

4 Incorporating prior information

In practice, some prior knowledge on the underlying GRN is typically available. In this section, we outline approaches to incorporating such prior knowledge into the point-based Gaussian approximation filters for network inference. In particular, we consider two types of prior information, namely, sparsity constraints and range constraints on the network. For networks with sparsity constraints, we incorporate an iterative thresholding procedure into the Gaussian approximation filters.

And to accommodate range constraints, we employ PDF-truncated Gaussian approximation filters.

4.1 Optimization-based approach for sparsity constraints

4.1.1 The optimization formulations

Note that under the Gaussian assumption, the state estimation $\hat{\mathbf{x}}_{k|k}$ of the Kalman filter is equivalently given by the solution to the following optimization problem [30,31]

$$\hat{\mathbf{x}}_{k|k} = \arg \min_{\bar{\mathbf{x}}} J(\bar{\mathbf{x}}), \quad (45)$$

$$\begin{aligned} \text{with } J(\bar{\mathbf{x}}) \triangleq & (\mathbf{y}_k - \mathbf{h}(\bar{\mathbf{x}}))^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{h}(\bar{\mathbf{x}})) \\ & + (\bar{\mathbf{x}} - \hat{\mathbf{x}}_{k|k-1})^T \mathbf{P}_{k|k-1}^{-1} (\bar{\mathbf{x}} - \hat{\mathbf{x}}_{k|k-1}). \end{aligned} \quad (46)$$

To incorporate the prior information of the GRN, (46) is modified as

$$\tilde{J}(\bar{\mathbf{x}}) = J(\bar{\mathbf{x}}) + \lambda J_p(\bar{\mathbf{x}}), \quad (47)$$

where $J_p(\bar{\mathbf{x}})$ is a penalty function associated with the prior information and λ is a tunable parameter that regulates the tightness of the penalty.

For example, in gene regulatory networks, each gene only interacts with a few genes [20]. To capture such a sparsity constraint, a Laplace prior distribution can be used for the connection coefficient matrix \mathbf{A} , i.e.,

$$p(\mathbf{A}) = (\lambda/2)^{n^2} \exp \left(-\lambda \sum_{i=1}^n \sum_{j=1}^n |a_{ij}| \right). \quad (48)$$

Therefore, in this case, $J_p(\bar{\mathbf{x}}) = -\log p(\mathbf{A}) = c_1 \|\mathbf{A}\|_1 + c_2$ where c_1 and c_2 are constants. And, (47) can be rewritten as

$$\tilde{J}(\bar{\mathbf{x}}) = J(\bar{\mathbf{x}}) + \lambda \|\mathbf{A}\|_1. \quad (49)$$

Note that (49) can also be interpreted as the result of applying the least squares shrinkage selection operator (LASSO) to (47). The LASSO adds an L_1 -norm constraint to the GRN so that the regulatory coefficient matrix \mathbf{A} tends to be sparse with many zero elements.

As another example, if some known regulatory relationship exists, then it should be taken into account to improve the estimation accuracy. Specifically, define an $n \times n$ indicator matrix $\mathbf{E} = [e_{ij}]$ where $e_{ij} = 1$ indicates that there is a lack of regulation from gene j to gene i . Then, similar to the use of LASSO, a penalty on a_{ij} should incur if $e_{ij} = 1$. Thus, (47) can be rewritten as

$$\tilde{J}(\bar{\mathbf{x}}) = J(\bar{\mathbf{x}}) + \lambda \|\mathbf{E} \circ \mathbf{A}\|_1. \quad (50)$$

Note that as in [20], here we do not force $a_{ij} = 0$ corresponding to $e_{ij} = 1$ but rather use an L_1 -norm penalty. The advantage of such an approach is that it allows the algorithm to pick different structures but more likely to

pick the edges without penalties. ‘ \circ ’ denotes the entry-wise product operation of two matrices.

4.1.2 Iterative thresholding algorithm

Solving the optimization problems in (49) and (50) is not straightforward since $|a|$ is non-differentiable at $a = 0$. In the following, an efficient solver called the iterative thresholding algorithm is introduced.

For convenience, we consider a general optimization problem of the form

$$\arg \min_{\bar{\mathbf{x}}} J(\bar{\mathbf{x}}) = L(\bar{\mathbf{x}}) + \|\boldsymbol{\lambda} \circ \bar{\mathbf{x}}\|_1, \quad (51)$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_{n^2+2n}]^T$ and $L(\bar{\mathbf{x}})$ is a smooth function. Note that if $\boldsymbol{\lambda} = [\mathbf{0}_{1 \times n}, \lambda \times \mathbf{1}_{1 \times n^2}, \mathbf{0}_{1 \times n}]^T$, then (51) becomes (49); and if $\boldsymbol{\lambda} = [\mathbf{0}_{1 \times n}, \lambda \times \hat{\boldsymbol{\theta}}, \mathbf{0}_{1 \times n}]^T$, then (51) becomes (50). Note that $\hat{\boldsymbol{\theta}} = [e_{11}, e_{12}, \dots, e_{1n}, \dots, e_{nn}]^T$.

The solution to (51) can be iteratively obtained by solving a sequence of optimization problems. As in Newton’s method, the Taylor series expansion of $L(\bar{\mathbf{x}})$ around the solution $\bar{\mathbf{x}}^t$ at the t -th iteration is given by

$$L(\bar{\mathbf{x}}^t + \Delta \bar{\mathbf{x}}) \cong L(\bar{\mathbf{x}}^t) + \Delta \bar{\mathbf{x}}^T \nabla L(\bar{\mathbf{x}}^t) + \frac{\alpha_t}{2} \|\Delta \bar{\mathbf{x}}\|_2^2, \quad (52)$$

where ∇L is the gradient of L and α_t is such that $\alpha_t \mathbf{I}$ mimics the Hessian $\nabla^2 L$. Then, $\bar{\mathbf{x}}^{t+1}$ is given by [32]

$$\bar{\mathbf{x}}^{t+1} = \arg \min_{\mathbf{z}} (\mathbf{z} - \bar{\mathbf{x}}^t)^T \nabla L(\bar{\mathbf{x}}^t) + \frac{\alpha_t}{2} \|\mathbf{z} - \bar{\mathbf{x}}^t\|_2^2 + \|\boldsymbol{\lambda} \circ \mathbf{z}\|_1. \quad (53)$$

The equivalent form of (53) is given by [32]

$$\bar{\mathbf{x}}^{t+1} = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{z} - \mathbf{u}^t\|_2^2 + \frac{1}{\alpha_t} \|\boldsymbol{\lambda} \circ \mathbf{z}\|_1, \quad (54)$$

$$\text{with } \mathbf{u}^t = \bar{\mathbf{x}}^t - \frac{1}{\alpha_t} \nabla L(\bar{\mathbf{x}}^t), \quad (55)$$

$$\alpha_t \approx \frac{(\mathbf{s}^t)^T \mathbf{r}^t}{\|\mathbf{s}^t\|^2}, \quad (56)$$

$$\mathbf{s}^t = \bar{\mathbf{x}}^t - \bar{\mathbf{x}}^{t-1}, \quad (57)$$

$$\mathbf{r}^t = \nabla L(\bar{\mathbf{x}}^t) - \nabla L(\bar{\mathbf{x}}^{t-1}). \quad (58)$$

The solution to (54) is given by [32] $\bar{\mathbf{x}}^{t+1} = \eta^S(\mathbf{u}^t, \frac{\boldsymbol{\lambda}}{\alpha_t})$, where

$$\eta^S(\mathbf{u}, \mathbf{a}) = \text{sign}(\mathbf{u}) \max\{|\mathbf{u}| - \mathbf{a}, \mathbf{0}\} \quad (59)$$

is the soft thresholding function with $\text{sign}(\mathbf{u})$ and $\max\{|\mathbf{u}| - \mathbf{a}, \mathbf{0}\}$ being component-wise operators.

Finally, the iterative procedure for solving (51) is given by

$$\bar{\mathbf{x}}^{t+1} = \text{sign} \left(\bar{\mathbf{x}}^t - \frac{1}{\alpha_t} \nabla L(\bar{\mathbf{x}}^t) \right) \max \left\{ \left| \bar{\mathbf{x}}^t - \frac{1}{\alpha_t} \nabla L(\bar{\mathbf{x}}^t) \right| - \frac{\boldsymbol{\lambda}}{\alpha_t}, \mathbf{0} \right\}. \quad (60)$$

And the iteration stops when the following condition is met

$$\frac{|J(\bar{\mathbf{x}}^t) - J(\bar{\mathbf{x}}^{t-1})|}{|J(\bar{\mathbf{x}}^{t-1})|} \leq \epsilon, \quad (61)$$

where ϵ is a given small number.

4.2 PDF truncation method for range constraints

If the range constraints on the regulatory coefficients are available, the inference accuracy can be improved by enforcing the constraints in the Gaussian approximation filters.

In particular, assume that we impose the following range constraints on the state vector $\bar{\mathbf{x}}$

$$\mathbf{c} \leq \bar{\mathbf{x}} \leq \mathbf{d}. \quad (62)$$

The PDF truncation method [31] can be employed to incorporate the above range constraint into the Gaussian approximation filters, by converting the updated mean $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ to the pseudo mean $\hat{\mathbf{x}}_{k|k}^t$ and covariance $\mathbf{P}_{k|k}^t$ which are then used in the next prediction and filtering steps.

We next briefly outline the PDF truncation procedure. We use $\hat{\mathbf{x}}_{k|k,i}^t$ and $\mathbf{P}_{k|k,i}^t$ to denote the mean and covariance after the first i constraints have been enforced. Initially, we set $\hat{\mathbf{x}}_{k|k,0}^t = \hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k,0}^t = \mathbf{P}_{k|k}$. Consider the following transformation

$$\mathbf{z}_{k,i} = \mathcal{G}_i \mathcal{D}_i^{-1/2} \mathcal{S}_i^T (\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_{k|k,i}^t) \quad (63)$$

where \mathcal{S}_i and \mathcal{D}_i are obtained from the Jordan canonical decomposition $\mathcal{S}_i \mathcal{D}_i \mathcal{S}_i^T = \mathbf{P}_{k|k,i}^t$ and \mathcal{G}_i is obtained by using the Gram-Schmidt orthogonalization and it satisfies [33]

$$\mathcal{G}_i \mathcal{D}_i^{1/2} \mathcal{S}_i^T \mathbf{e}_i = \left[\left(\mathbf{e}_i^T \mathbf{P}_{k|k,i}^t \mathbf{e}_i \right)^{1/2}, 0, \dots, 0 \right]. \quad (64)$$

Then, the upper bound $\mathbf{e}_i^T \bar{\mathbf{x}} \leq d_i$ is transformed to [33]

$$[1, 0, \dots, 0] \mathbf{z}_{k,i} \leq \frac{d_i - \mathbf{e}_i^T \hat{\mathbf{x}}_{k|k,i}^t}{\left(\mathbf{e}_i^T \mathbf{P}_{k|k,i}^t \mathbf{e}_i \right)^{1/2}} \triangleq \tilde{d}_i. \quad (65)$$

Similarly, the lower bound $\mathbf{e}_i^T \bar{\mathbf{x}} \geq c_i$ is transformed to

$$[1, 0, \dots, 0] \mathbf{z}_{k,i} \geq \frac{c_i - \mathbf{e}_i^T \hat{\mathbf{x}}_{k|k,i}^t}{\left(\mathbf{e}_i^T \mathbf{P}_{k|k,i}^t \mathbf{e}_i \right)^{1/2}} \triangleq \tilde{c}_i. \quad (66)$$

The constraint requires that the first element of $\mathbf{z}_{k,i}$ lies between \tilde{c}_i and \tilde{d}_i . Hence, only the truncated PDF of the first element of $\mathbf{z}_{k,i}$ is considered and it is given by [33]

$$f(z) = \alpha_i \exp(-z^2/2), \quad (67)$$

$$\text{with } \alpha_i = \frac{\sqrt{2}}{\sqrt{\pi} [\text{erf}(\tilde{d}_i/\sqrt{2}) - \text{erf}(\tilde{c}_i/\sqrt{2})]}. \quad (68)$$

Then, the mean and variance of the first element of $\mathbf{z}_{k,i}$ after imposing the i -th constraint are given respectively by

$$\mu_i = \int_{\tilde{c}_i}^{\tilde{d}_i} z f(z) dz = \alpha_i \left[\exp(-\tilde{c}_i^2/2) - \exp(-\tilde{d}_i^2/2) \right], \quad (69)$$

$$\begin{aligned} \sigma_i^2 &= \int_{\tilde{c}_i}^{\tilde{d}_i} (z - \mu_i)^2 f(z) dz \\ &= \alpha_i \left[\exp(-\tilde{c}_i^2/2) (\tilde{c}_i - 2\mu_i) \right. \\ &\quad \left. - \exp(-\tilde{d}_i^2/2) (\tilde{d}_i - 2\mu_i) \right] + \mu_i^2 + 1. \end{aligned} \quad (70)$$

Thus, the mean and covariance of the transformed state vector after imposing the i -th constraint are given respectively by

$$\bar{\mathbf{z}}_{k,i} = [\mu_i, 0, \dots, 0]^T, \quad (71)$$

$$\mathbf{Q}_{k,i} = \text{diag}([\sigma_i^2, 1, \dots, 1]). \quad (72)$$

By taking the inverse transform of (63), we then get

$$\hat{\mathbf{x}}_{k|k,i+1}^t = \mathcal{S}_i \mathcal{D}_i^{1/2} \mathcal{G}_i^T \bar{\mathbf{z}}_{k,i} + \hat{\mathbf{x}}_{k|k,i}^t \quad (73)$$

$$\mathbf{P}_{k|k,i+1}^t = \mathcal{S}_i \mathcal{D}_i^{1/2} \mathcal{G}_i^T \mathbf{Q}_{k,i} \mathcal{G}_i \mathcal{D}_i^{1/2} \mathcal{S}_i^T. \quad (74)$$

After imposing all n constraints, the final constrained state estimate and covariance at time k are given respectively by $\hat{\mathbf{x}}_{k|k}^t \triangleq \hat{\mathbf{x}}_{k|k,n}^t$ and $\mathbf{P}_{k|k}^t \triangleq \mathbf{P}_{k|k,n}^t$.

5 Numerical results

5.1 Synthetic network

In this section, a synthetic network that contains eight genes is used to test the performance of the EKF, the

Table 1 Comparison of UKF with different κ

Filters	True positive rate			False positive rate			Positive predictive rate		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
UKF($\kappa = -5$)	0.7576	0.9355	0.8472	0.5000	0.7647	0.5955	0.5094	0.6279	0.5824
UKF($\kappa = -2$)	0.7576	0.9355	0.8406	0.5161	0.7647	0.5933	0.5094	0.6279	0.5825
UKF($\kappa = 0$)	0.7576	0.9375	0.8426	0.5161	0.7647	0.5918	0.5094	0.6364	0.5840
UKF($\kappa = 2$)	0.7576	0.9375	0.8407	0.5152	0.7353	0.5895	0.5098	0.6279	0.5841
UKF($\kappa = 5$)	0.7576	0.9063	0.8394	0.5161	0.7353	0.5933	0.5192	0.6279	0.5821

Table 2 Comparison of different filters

Filters	True positives #			False positives #			True negatives #			False negatives #		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
EKF	2	17	10.60	23	44	36.4	2	15	7.08	2	24	9.92
UKF	25	29	26.80	16	26	19.28	8	16	13.06	2	8	4.86
CKF ₃	25	30	26.74	16	26	19.10	8	15	13.14	2	8	5.02
CKF ₅	25	29	26.64	16	26	19.24	8	16	13.08	1	8	5.04

Filters	True positive rate			False positive rate			Positive predictive rate		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
EKF	0.0769	0.8667	0.5224	0.6053	0.9545	0.8358	0.0800	0.3208	0.2231
UKF	0.7576	0.9355	0.8472	0.5	0.7576	0.5955	0.5094	0.6279	0.5824
CKF ₃	0.7576	0.9375	0.8426	0.5161	0.7647	0.5918	0.5094	0.6364	0.5840
CKF ₅	0.7576	0.9667	0.8417	0.5000	0.7647	0.5946	0.5094	0.6279	0.5814

Table 3 Inferred results of the conventional filter and filters incorporating the prior information

Filters	True positives #			False positives #			True negatives #			False negatives #		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
UKF	25	29	26.80	16	26	19.28	8	16	13.06	2	8	4.86
UKF _{p1}	25	29	27.34	14	19	16.52	13	18	15.72	2	8	4.42
UKF _{p2}	23	26	24.16	13	16	13.86	16	18	17.20	7	10	8.78
UKF _{p3}	25	29	26.70	12	24	17.50	9	19	14.50	3	8	5.30
Filters	True positive rate			False positive rate			Positive predictive rate					
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg			
UKF	0.7576	0.9355	0.8472	0.5	0.7647	0.5955	0.5094	0.6279	0.5824			
UKF _{p1}	0.7576	0.9355	0.8614	0.4375	0.5935	0.5121	0.5778	0.6744	0.6239			
UKF _{p2}	0.6970	0.7879	0.7335	0.4194	0.5000	0.4462	0.5897	0.6667	0.6355			
UKF _{p3}	0.7576	0.9063	0.8348	0.3871	0.7273	0.5463	0.5294	0.6923	0.6049			

Table 4 Comparison of UKF_{p1} using different λ

Filters	True positive rate			False positive rate			Positive predictive rate		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
UKF _{p1} (λ = 0.1)	0.7576	0.9355	0.8484	0.5000	0.7647	0.5900	0.5094	0.6279	0.5850
UKF _{p1} (λ = 0.5)	0.7576	0.9677	0.8535	0.4688	0.7647	0.5696	0.5094	0.6512	0.5948
UKF _{p1} (λ = 1)	0.7576	0.9355	0.8614	0.4375	0.5935	0.5121	0.5778	0.6744	0.6239
UKF _{p1} (λ = 5)	0.7500	0.9355	0.8439	0.3548	0.5455	0.4672	0.5814	0.7105	0.6456
UKF _{p1} (λ = 10)	0.7273	0.9063	0.8217	0.3226	0.4848	0.4156	0.6190	0.7368	0.6695

UKF, the CKF₃, the CKF₅, and their corresponding filters incorporating the prior information. Forty data points are collected to infer the structure of the network. The system noise and measurement noise are assumed to be Gaussian distributed with means **0** and covariances $\bar{Q}_k = \text{diag}([0.01I_8 \quad O_{72}])$ and $R_k = 0.01I_8$, respectively. The connection coefficient matrix is given by

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2.4 & 3.2 \\ 0 & 0 & 0 & 4.1 & 0 & -2.4 & 0 & 4.1 \\ -5.0 & 2.1 & -1.5 & 0 & 4.5 & 0 & 2.1 & 0 \\ 0 & 1.3 & 2.5 & -3.7 & 1.8 & 0 & 0 & -3.1 \\ 0 & 0 & 0 & -2.6 & -3.2 & 0 & -1 & 4 \\ -1.5 & -1.8 & 0 & 3.4 & 1.4 & 1.1 & 0 & 1.7 \\ -1.8 & 0 & 0 & -3 & 1.1 & 2.4 & 0 & 0 \\ -1.3 & 0 & -1 & 0 & 2.1 & 0 & 0 & 2.2 \end{pmatrix} \quad (75)$$

and $\mu_i = 2, i = 1, \dots, 8$. For the filter, each coefficient in \hat{A} is initialized from a Gaussian distribution with mean 0 and variance 0.2. Moreover, the coefficient μ_i is initialized from a Gaussian distribution with mean 1.5 and variance 0.2. The system state is initialized using the first measurement.

The metric used to evaluate the inferred GRN is the true positive rate (TPR), the false positive rate (FPR), and the positive predictive value (PPV). They are given by [34]

$$\text{TPR} = \frac{\text{TP\#}}{\text{TP\#} + \text{FN\#}}, \quad (76)$$

$$\text{FPR} = \frac{\text{FP\#}}{\text{FP\#} + \text{TN\#}}, \quad (77)$$

$$\text{PPV} = \frac{\text{TP\#}}{\text{TP\#} + \text{FP\#}}, \quad (78)$$

where the number of true positives (TP#) denotes the number of links correctly predicted by the inference algorithm; the number of false positives (FP#) denotes the number of incorrectly predicted links; the number of true negatives (TN#) denotes the number of correctly predicted nonlinks; and the number of false negatives (FN#) denotes the number of missed links by the inference algorithm [8].

5.1.1 Comparison of the EKF with point-based Gaussian approximation filters

The UKF with different parameter κ is tested. The simulation results based on 50 Monte Carlo runs are shown in Table 1. It can be seen that UKFs with $\kappa = 0, 2, 5$ have slight better performance than UKFs with $\kappa = -5, -2$. One possible reason is that the weights of all sigma points used in the UKF are all positive when $\kappa \geq 0$. In general, all positive weights will guarantee better stability of the filtering algorithm. However, it should be emphasized that, in this specific example, there is no big difference between UKFs with different κ . In addition, the objective of this paper was to investigate the proposed filter incorporating the prior information. Hence, the UKF is used to denote UKF with $\kappa = 3 - n$ and compare with the filters incorporating the prior information.

The inference results of the EKF, the UKE, the CKF₃, and the CKF₅ are summarized in Table 2, all results are based on 50 Monte Carlo runs. It can be seen that all point-based Gaussian approximation filters have better performance than the EKF since the average(avg) FPR is lower and the average TPR and precision are higher than that of the EKF. Although the CKFs exhibit slightly better

Table 5 Effect of strength of the links using different λ

Filters	True positive rate			False positive rate			Positive predictive rate		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
UKF _{p1} (λ = 0.1)	0.7576	0.9677	0.8484	0.4688	0.7647	0.5713	0.5094	0.6512	0.5929
UKF _{p1} (λ = 0.5)	0.7576	0.9333	0.8468	0.4516	0.7059	0.5422	0.5385	0.6512	0.6057
UKF _{p1} (λ = 1)	0.7500	0.9032	0.8221	0.3750	0.5758	0.4953	0.5814	0.6842	0.6257
UKF _{p1} (λ = 5)	0.7273	0.8750	0.8220	0.3548	0.5000	0.4169	0.6098	0.7179	0.6684
UKF _{p1} (λ = 10)	0.7500	0.8750	0.8214	0.3226	0.5000	0.4143	0.6098	0.7368	0.6696

Table 6 Effect of false prior information using different λ

Filters	True positive rate			False positive rate			Positive predictive rate		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
UKF _{p1} ($\lambda = 0.1$)	0.7576	0.9667	0.8491	0.5000	0.7647	0.5933	0.5094	0.6279	0.5835
UKF _{p1} ($\lambda = 0.5$)	0.7576	0.9355	0.8535	0.4839	0.7647	0.5962	0.5094	0.6279	0.5836
UKF _{p1} ($\lambda = 1$)	0.7576	0.9333	0.8572	0.4839	0.7059	0.6001	0.5200	0.6279	0.5830
UKF _{p1} ($\lambda = 5$)	0.6970	0.8125	0.7546	0.4194	0.5938	0.5000	0.5682	0.6486	0.6062
UKF _{p1} ($\lambda = 10$)	0.5758	0.7576	0.6810	0.3226	0.5000	0.4066	0.5676	0.7059	0.6369

filtering performance than the UKF in some runs, they are comparable in terms of TPR, FPR, and PPV.

Based on the above tests, in the rest of the paper, only the UKF is used.

5.1.2 Comparison of the UKF and the UKF incorporating the prior information

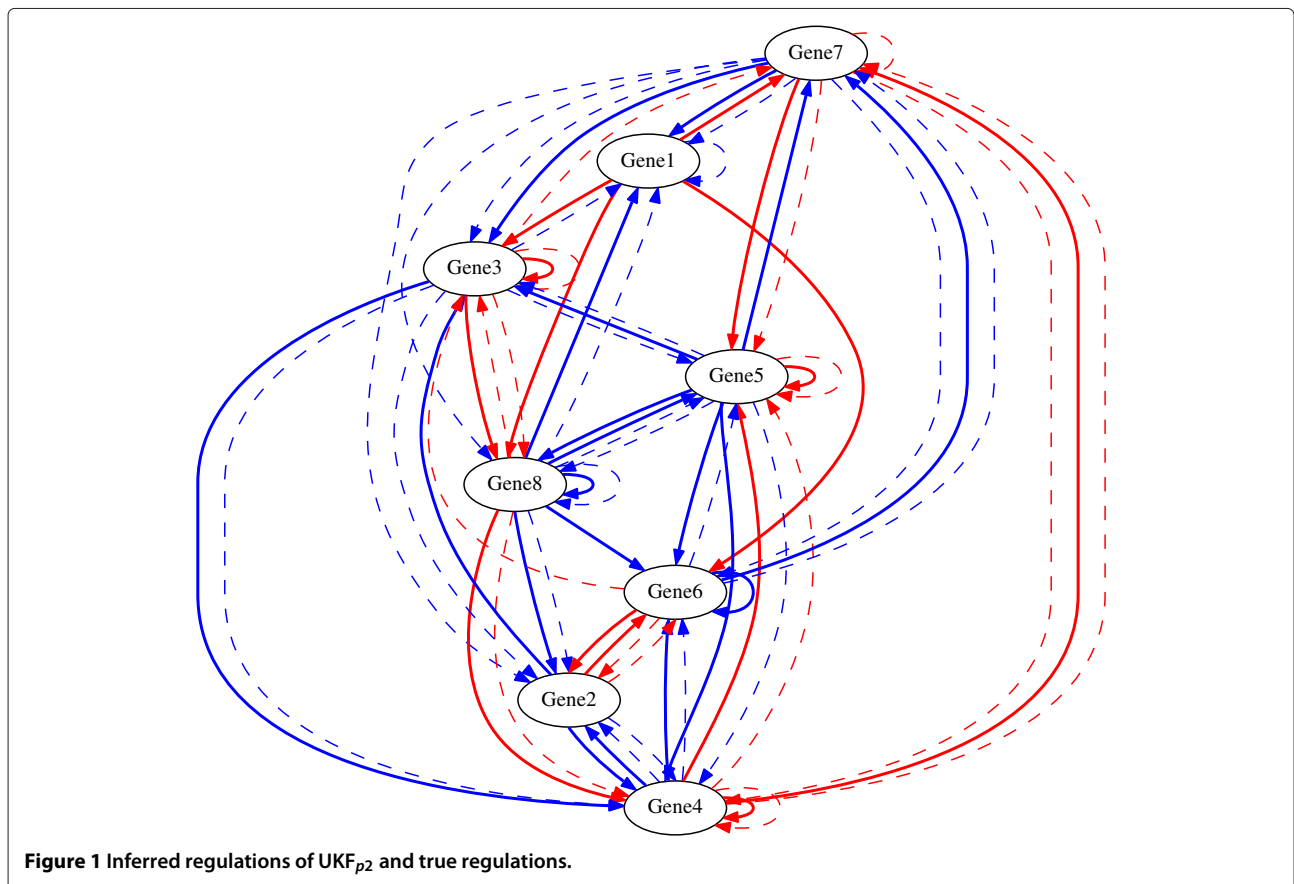
As mentioned above, the UKF is used as a typical filter to compare the performance with and without the prior information.

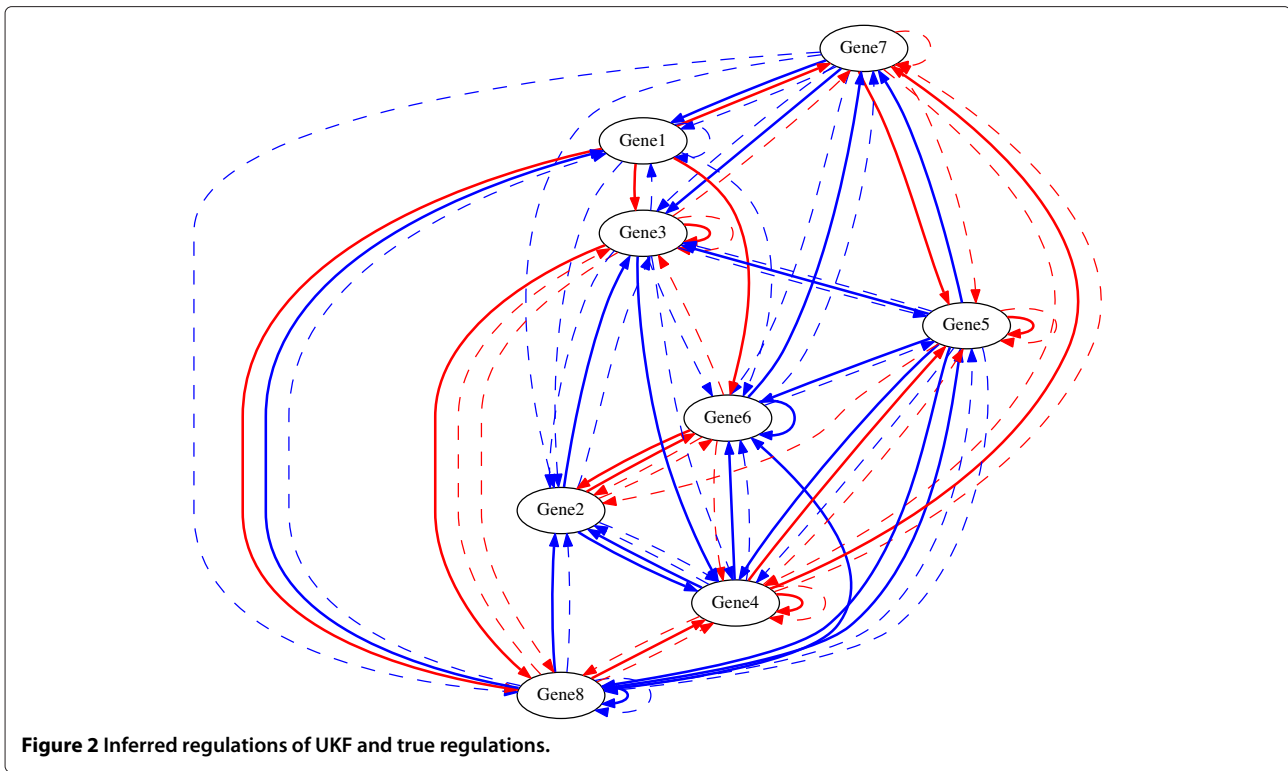
Incorporating existing network information The following prior existing network information is assumed to be known: 1) gene1, gene5, and gene7 have little possibility to regulate gene2; 2) gene2, gene3, gene8 have little

possibility to regulate gene7. Hence, the indicator matrix in (50) is given by

$$E = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{79}$$

The comparison of the UKF and the UKF incorporating the existing network information (denoted by UKF_{p1}) with $\lambda = 2$ is shown in Table 3. It can be seen that the





average TP# and TN# of the UKF_{p1} are both higher than those of the UKF. In addition, the average FP# and FN# of the UKF_{p1} are lower than those of the UKF. Hence, the UKF_{p1} predicts more correct links and nonlinks than the UKF. Moreover, the UKF_{p1} produces less incorrect links and missed links than the UKF. The average TPR and the precision of the UKF_{p1} are higher than those of the UKF. In addition, the average FPR of the UKF_{p1} is lower than that of the UKF. Hence, by using the existing network information, the inference accuracy can be improved.

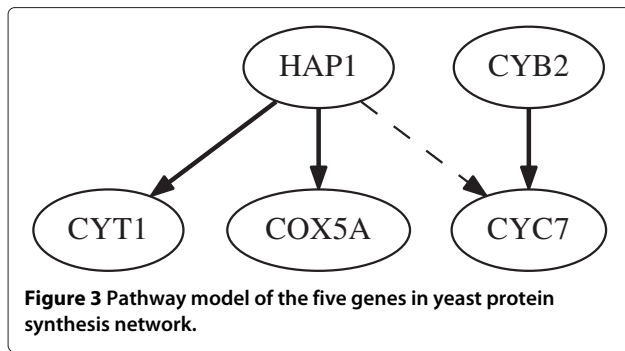
The performance of UKF_{p1} with different λ is shown in Table 4. It is seen that the performance of UKF_{p1} and UKF is close when λ is small since only small regulation is imposed on the solution. When λ is large, the difference between the UKF_{p1} and UKF is large. In particular, the UKF_{p1} provides sparser solution than the UKF when λ is large. It can be seen from Table 4, the average FPR

of UKF_{p1} decreases with the increasing of λ . The average TPR of UKF_{p1}, however, does not increase monotonically with the increasing of λ . The average PPR of UKF_{p1} increases with the increasing of λ . Hence, roughly speaking, the UKF_{p1} is better than the UKF when large λ is used.

To consider the strength of the links, rather than setting it to 1, e_{ij} (in the indicator matrix E) is set to different values. Large e_{ij} is used if the strength of the link from gene j to gene i is strong. For convenience, the UKF considering the strength of links is denoted as UKF_{p̂1}. To compare the performance of UKF_{p̂1} with UKF_{p1}, for UKF_{p̂1}, the values of the second row in Equation (79) is multiplied by 5. The performance of UKF_{p̂1} using different λ is given in Table 5. It can be seen from Tables 4 and 5 that the performance of UKF_{p̂1} and UKF_{p1} is close when λ is small, e.g., $\lambda = 0.1$. In addition, the average TPR and FPR of UKF_{p̂1} is smaller

Table 7 Comparison of UKF_{p2} using different λ

Filters	True positive rate			False positive rate			Positive predictive rate		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
UKF _{p2} ($\lambda = 0.1$)	0.7576	0.9355	0.8304	0.4839	0.6970	0.5699	0.5306	0.6512	0.5914
UKF _{p2} ($\lambda = 0.5$)	0.6970	0.8710	0.7750	0.4194	0.5758	0.4902	0.5682	0.6585	0.6198
UKF _{p2} ($\lambda = 1$)	0.6970	0.7879	0.7335	0.4194	0.5000	0.4462	0.5897	0.6667	0.6355
UKF _{p2} ($\lambda = 5$)	0.4545	0.6667	0.5501	0.3226	0.4516	0.3791	0.5714	0.6471	0.6064
UKF _{p2} ($\lambda = 10$)	0.4545	0.5455	0.4800	0.2903	0.3871	0.3523	0.5556	0.6538	0.5920

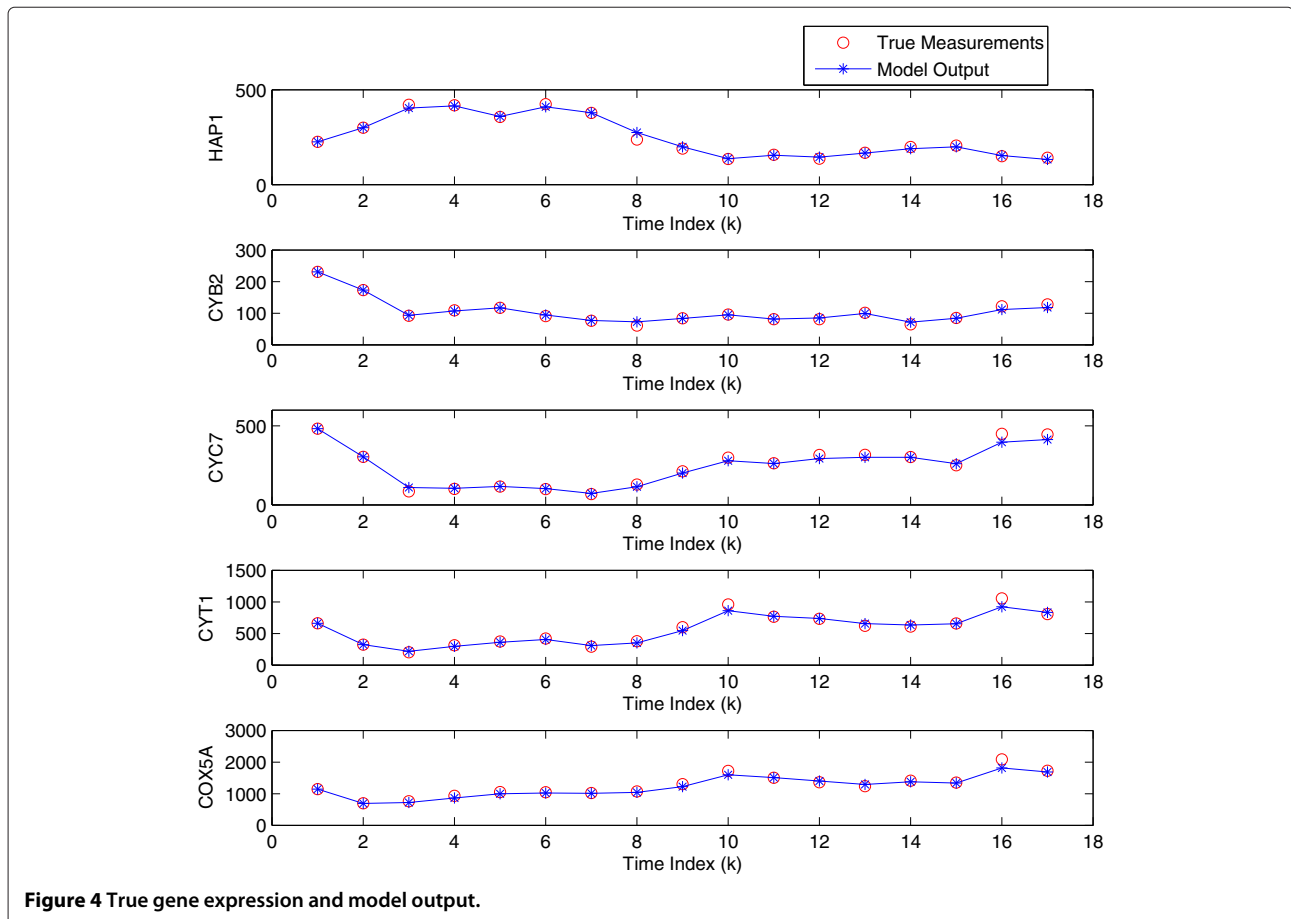


than that of UKF_{p1} for all tested λ except for $\lambda = 0.1$. Hence, PPR is used to evaluate the performance of $UKF_{\hat{p}1}$ and UKF_{p1} . Although the average PPR of $UKF_{\hat{p}1}$ and UKF_{p1} is close when the λ is large, e.g., $\lambda = 10$, the average PPR of $UKF_{\hat{p}1}$ is consistently higher than that of UKF_{p1} . The results indicate that the inference accuracy of $UKF_{\hat{p}1}$ and UKF_{p1} are close when λ is very small or very large. The inference accuracy of $UKF_{\hat{p}1}$ outperforms UKF_{p1} when the appropriate strength of the link and parameter λ are used.

To consider the effect of false prior knowledge, the second row of the indicator matrix in Equation (79) is changed to $[0, 1, 1, 1, 0, 1, 0, 1]$, which conflicts with the truth. For convenience, we use $UKF_{\bar{p}1}$ to denote the UKF incorporating this false prior knowledge. In Table 6, the performance of $UKF_{\bar{p}1}$ with different λ is shown. It can be seen from Tables 4 and 6 that the average TPR of $UKF_{\bar{p}1}$ is smaller than that of UKF_{p1} when λ is small, e.g., $\lambda = 0.1, 0.5$. In addition, the average FPR of $UKF_{\bar{p}1}$ is larger than that of UKF_{p1} when λ is large, e.g., $\lambda = 5, 10$. Moreover, although the average PPR of $UKF_{\bar{p}1}$ is close to that of UKF_{p1} when λ is small, the average PPR of $UKF_{\bar{p}1}$ is consistently lower than that of UKF_{p1} . Hence, as expected, the results indicate that the false prior knowledge will lead to worse inference result.

Incorporating LASSO The problem setup is the same as before except that the LASSO rather than the existing network information is used. The UKF incorporating LASSO is denoted as UKF_{p2} .

As shown in Table 3, the average TP# and FP# of UKF_{p2} are lower than those of UKF and the average TN# and FN# of UKF_{p2} are higher than those of UKF. Hence, UKF_{p2} produces less links, including correct and incorrect ones.



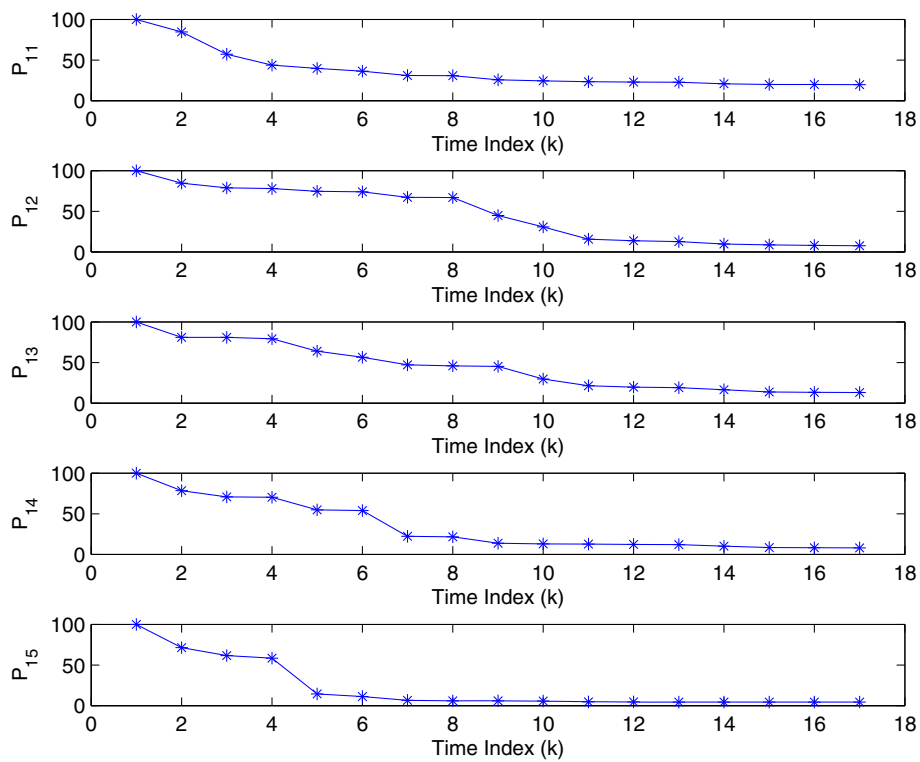


Figure 5 Variance of regulatory coefficients.

In addition, UKF_{p2} produces more nonlinks and missed links. It is consistent with the fact that the LASSO tends to provide a sparse solution. It can be seen from Table 3 that the average FPR of UKF_{p2} is lower than that of UKF and the average precision of UKF_{p2} is higher than that of UKF. Hence, by incorporating LASSO, the inference accuracy is improved.

A representative inference result of UKF_{p2} and the true regulations are shown in Figure 1. For comparison, the inference result of UKF and the true regulations are shown in Figure 2. By comparing Figure 2 and Figure 1, it can be seen that UKF falsely predicts the nonlinks from gene1 to gene2, from gene3 to gene6, from gene4 to gene8, from gene5 to gene2, and from gene6 to gene4 while UKF_{p2} does not.

The performance UKF_{p2} with different λ is shown in Table 7. It is seen that the performance of UKF_{p2} and

UKF is close when λ is small since only small regulation is imposed on the solution. When λ is large, the difference between UKF_{p2} and UKF is large. The average TPR and FPR of UKF_{p2} decrease with the increasing of the λ . The average PPR does not increase monotonically with the increasing of λ . Generally speaking, for different λ , UKF_{p2} is more sensitive than that of UKF_{p1}. Although the performance of UKF_{p2} depends on λ , the average PPR of UKF_{p2} is consistently higher than that of UKF. Hence, roughly speaking, UKF_{p2} has better performance than UKF.

Incorporating the range constraint The existing network information can be used to provide the rough range constraint of \bar{x} . A tight constraint is forced on the regulation coefficient a_{ij} when there is a small regulation possibility from gene_j to gene_i and a loose constraint is forced on the regulation coefficient with no prior information.

Table 8 Inferred results of the UKF and UKF_{p2}

Filters	True positives #	False positives #	True negatives #	False negatives #
UKF	1	7	14	3
UKF _{p2}	2	3	18	2
Filters	TPR	FPR	Precision	
UKF	0.25	0.3333	0.1250	
UKF _{p2}	0.5000	0.1429	0.4000	

In the simulation, for the coefficients corresponding to the zero elements in (79), the lower bound and the upper bound are set as -10 and 10 , respectively. For the coefficients corresponding to the nonzero elements in (79), the lower bound and the upper bound are set as -0.1 and 0.1 , respectively. The UKF incorporating the range constraint is denoted as UKF_{p3} . As shown in Table 3, the average FPR of UKF_{p3} is lower than that of UKF and the average precision of UKF_{p3} is higher than that of UKF.

5.2 Yeast protein synthesis network

In this section, time-series gene expression data of the yeast protein synthesis network is used. Five genes (HAP1, CYB2, CYC7, CYT1, and COX5A) of the yeast protein synthesis network are considered and 17 data points which can be found in [35] are collected. The regulation relationship between them has been revealed by the biological experiment and shown in Figure 3. The dashed arrow in Figure 3 denotes 'repression' and the solid arrow denotes 'activation.'

The GRN is inferred by the UKF and UKF_{p2} . The predicted gene expressions using parameters estimated by UKF_{p2} and the true measured gene expressions are shown in Figure 4. It can be seen that the model output fits the measured data well. The variances of the regulatory coefficients of HAP1 (P_{1i} ($1 \leq i \leq 5$)) are shown in Figure 5. It can be seen that the filter converges since the variance P_{1i} approaches zero. The results for other regulatory coefficients are similar and not shown here. The evaluation of the inferred GRN by UKF and UKF_{p2} is shown in Table 8.

By incorporating the sparsity constraint, UKF_{p2} provides much better inference results than UKF. As shown in Table 8, the TP# and TN# of UKF_{p2} are higher than those of UKF and the FP# and FN# are lower than those of UKF. In addition, it can be seen from Table 8, the FPR of UKF_{p2} is lower than that of UKF and the TPR and the precision of UKF_{p2} is higher than that of UKF.

6 Conclusions

In this paper, we have proposed a framework of employing the point-based Gaussian approximation filters which incorporates the prior knowledge to infer the gene regulatory network (GRN) based on the gene expression data. The performance of the proposed framework is tested by a synthetic network and the yeast protein synthesis network. Numerical results show that the inference accuracy of the GRN by the proposed point-based Gaussian approximation filter incorporating the prior information is higher than using the traditional filters without incorporating prior knowledge. The proposed method works for small- and medium-size GRNs due to the computational complexity considerations. It remains a future

research topic how to adapt the proposed inference framework to handle large GRNs at reasonable computational complexity.

Competing interests

All authors declare that they have no competing interests.

Author details

¹Intelligent Fusion Technology, Germantown, MD 20876, USA. ²Department of Electrical Engineering, Columbia University, New York, NY 10027, USA.

Received: 25 July 2013 Accepted: 11 November 2013

Published: 17 December 2013

References

1. M Zou, SD Conzen, A new dynamic Bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics* **21**(1), 71–79 (2005)
2. X Zhou, X Wang, R Pal, I Ivanov, M Bittner, ER Dougherty, A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics* **20**(17), 2918–2927 (2004)
3. M Quach, N Brunel, F d'Alché Buc, Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics* **23**(23), 3209–3216 (2007)
4. Z Wang, X Liu, Y Liu, J Liang, V Vinciotti, An extended Kalman filtering approach to modeling nonlinear dynamic gene regulatory networks via short gene expression time series. *Comput. Biol. Bioinformatics, IEEE/ACM Trans.* **6**(3), 410–419 (2009)
5. X Wu, P Li, N Wang, P Gong, EJ Perkins, Y Deng, C Zhang, State space model with hidden variables for reconstruction of gene regulatory networks. *BMC Syst Biol.* **5**(Suppl 3), S3 (2011)
6. AV Werhli, D Husmeier, Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat. Appl. Genet. Mol. Biol.* **6**, Article 15 (2007)
7. J Mazur, D Ritter, G Reinelt, L Kaderali, Reconstructing nonlinear dynamic models of gene regulation using stochastic sampling. *BMC Bioinformatics* **10**, 448 (2009)
8. A Noor, E Serpedin, M Nounou, H Nounou, Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity. *Comput. Biol. Bioinformatics, IEEE/ACM Trans.* **9**(4), 1203–1211 (2012)
9. M Hecker, S Lambeck, S Toepfer, E van Someren, R Guthke, Gene regulatory network inference: Data integration in dynamic models - a review. *Biosystems* **96**(1), 86–103 (2009)
10. F Markowetz, R Spang, Inferring cellular networks - a review. *BMC Bioinformatics* **8**(Suppl 6), S5 (2007)
11. Y Huang, I Tienda-Luna, Y Wang, Reverse engineering gene regulatory networks. *Signal Process. Mag., IEEE* **26**(1), 76–97 (2009)
12. H de Jong, Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.* **9**, 67–103 (2002)
13. SJ Julier, JK Uhlmann, Unscented filtering and nonlinear estimation. *Proc. IEEE* **92**(3), 401–422 (2004)
14. K Ito, K Xiong, Gaussian filters for nonlinear filtering problems. *Automatic Control, IEEE Trans.* **45**(5), 910–927 (2000)
15. I Arasaratnam, S Haykin, Cubature kalman filters. *Automatic Control, IEEE Trans.* **54**(6), 1254–1269 (2009)
16. B Jia, M Xin, Y Cheng, Sparse-grid quadrature nonlinear filtering. *Automatica* **48**(2), 327–341 (2012)
17. M Arulampalam, S Maskell, N Gordon, T Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Process., IEEE Trans.* **50**(2), 174–188 (2002)
18. X Shen, H Vikalo, Inferring parameters of gene regulatory networks via particle filtering. *EURASIP J. Adv. Signal Process.* **2010**, 204612 (2010)
19. E Steele, A Tucker, PA 't Hoen, M Schuemie, Literature-based priors for gene regulatory networks. *Bioinformatics* **25**(14), 1768–1774 (2009)
20. S Christley, Q Nie, X Xie, Incorporating existing network information into gene network inference. *PLoS ONE* **4**(8), e6799 (2009)
21. Y Tamada, S Kim, H Bannai, S Imoto, K Tashiro, S Kuhara, S Miyano, Estimating gene networks from gene expression data by combining

- Bayesian network model with promoter element detection. *Bioinformatics* **19**(suppl 2), 227–236 (2003)
22. H Li, M Zhan, Unraveling transcriptional regulatory programs by integrative analysis of microarray and transcription factor binding data. *Bioinformatics* **24**(17), 1874–1880 (2008)
 23. N Bouaynaya, R Shterenberg, D Schonfeld, Methods for optimal intervention in gene regulatory networks [applications corner]. *Signal Process. Mag., IEEE* **29**(1), 158–163 (2012)
 24. L Chen, K Aihara, Chaos and asymptotical stability in discrete-time neural networks. *Physica D: Nonlinear Phenomena* **104**(3), 286–325 (1997)
 25. L Qian, H Wang, ER Dougherty, Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering. *Signal Process., IEEE Trans.* **56**(7), 3327–3339 (2008)
 26. J Vohradsky, Neural model of the genetic network. *J. Biol. Chem.* **276**(39), 36168–36173 (2001)
 27. E Mjolsness, T Mann, R Castano, B Wold, From coexpression to coregulation: an approach to inferring transcriptional regulation among gene classes from large-scale expression data. in *Advances in Neural Information Processing Systems* **12**, 928–934 (1999)
 28. M Nørgaard, NK Poulsen, O Ravn, New developments in state estimation for nonlinear systems. *Automatica* **36**(11), 1627–1638 (2000)
 29. IP Mysovskikh, *The Approximation of Multiple Integrals by Using Interpolatory Cubature Formulae in Quantitative Approximation*, ed. by R DeVore, K Scherer (Academic Press, New York, 1980)
 30. AH Jazwinski, *Stochastic Processes and Filtering Theory* (Academic Press Inc., Waltham, MA, 2007)
 31. BO Teixeira, LA Tôres, LA Aguirre, DS Bernstein, On unscented Kalman filtering with state interval constraints. *J. Process Control* **20**(1), 45–57 (2010)
 32. S Wright, R Nowak, M Figueiredo, Sparse reconstruction by separable approximation. *Signal Process., IEEE Trans.* **57**(7), 2479–2493 (2009)
 33. D Simon, DL Simon, Constrained Kalman filtering via density function truncation for turbofan engine health estimation. *Int. J. Syst. Sci.* **41**(2), 159–171 (2010)
 34. F Emmert-Strib, M Dehmer, *Analysis of Microarray Data* (Wiley-Blackwell, Hoboken, NJ, 2008)
 35. H Wang, L Qian, E Dougherty, Inference of gene regulatory networks using s-system: a unified approach. *Syst. Biol., IET* **4**(2), 145–156 (2010)

doi:10.1186/1687-4153-2013-16

Cite this article as: Jia and Wang: Gene regulatory network inference by point-based Gaussian approximation filters incorporating the prior information. *EURASIP Journal on Bioinformatics and Systems Biology* 2013 **2013**:16.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
