JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 21, Number 11, 2014
© Mary Ann Liebert, Inc.
Pp. 855–865
DOI: 10.1089/cmb.2014.0117

Bayesian Blind Source Separation for Data with Network Structure

KATRIN ILLNER, CHRISTIANE FUCHS, and FABIAN J. THEIS

ABSTRACT

In biology, more and more information about the interactions in regulatory systems becomes accessible, and this often leads to prior knowledge for recent data interpretations. In this work we focus on multivariate signaling data, where the structure of the data is induced by a known regulatory network. To extract signals of interest we assume a blind source separation (BSS) model, and we capture the structure of the source signals in terms of a Bayesian network. To keep the parameter space small, we consider stationary signals, and we introduce the new algorithm `emGrade`**, where model parameters and source signals are estimated using expectation maximization. For network data, we find an improved estimation performance compared to other BSS algorithms, and the flexible Bayesian modeling enables us to deal with repeated and missing observation values. The main advantage of our method is the statistically interpretable likelihood, and we can use model selection criteria to determine the (in general unknown) number of source signals or decide between different given networks. In simulations we demonstrate the recovery of the source signals dependent on the graph structure and the dimensionality of the data.**

Key words: Bayesian network, expectation maximization, linear mixing model, model selection, stationary signals.

1. INTRODUCTION

The separation of informative signals from multivariate data is a widespread task in biological applications. The data might, for example, be from gene regulatory networks or metabolomic pathways, where one is interested in the underlying processes. The aspect of separation is known as blind source separation (BSS); that is, the signals of interest and the actual experimental data are linked by a linear mixing model. In this work, we focus on signaling data, where the signals are associated with some known network structure, and we use this structure to archive a more appropriate separation of the data.

A widely used BSS approach for stationary time-series data is to diagonalize the time-delayed covariance (autocovariance) of the multivariate process. Recently in our group, Kowarsch et al. (2010) generalized this concept to signaling data. Based on stationarity assumptions for networks they introduced the graph-delayed covariance and again, diagonalization yields an estimate for the separated signals. We continue the approach of Kowarsch et al. and provide a probabilistic source separation method for network data. Parts of

Institute of Computational Biology, Helmholtz Zentrum München, German Research Center for Environmental Health, Neuherberg, Germany; and Institute for Mathematical Sciences. Technische Universität München, Munich, Germany.

this method have been introduced in our earlier papers (Illner et al., 2012; 2014) but with a focus on the application to gene expression data.

The main idea is to describe the unknown signals in terms of a Bayesian network. This concept is well known for the modeling of regulatory systems. Friedman et al. (2000) and Imoto et al. (2002) used Bayesian networks to describe the dependence and interaction of genes involved in the cell cycle. In this work, we assume Gaussian random variables and a linear dependence between the variables. The strength of dependence is parameterized by the graph-delayed covariance, and we learn the mixing parameters in a Bayesian framework. The resulting algorithm is called `emGrade` (expectation-maximization graph-decorrelation algorithm), and in simulations with network data it leads to improved estimation results compared to other BSS algorithms. A drawback of our approach is relatively specific assumptions on the stochastic properties of the signals; on the other hand, we benefit from Bayesian modeling. Using model selection criteria we can determine the correct number of unknown source signals and identify the most appropriate network structure of these signals. Throughout the article we use bold symbols to denote random variables and solid symbols to denote parameters and realizations of random variables.

2. STATISTICAL MODEL FOR DATA WITH NETWORK STRUCTURE

The signals we are interested in are associated with some known network structure. More precisely, the information of a signal propagates along the edges of the network. In the following, we assume a directed acyclic graph and define the distribution of the signals in terms of a Bayesian network.

2.1. A stationary Gaussian model

Let $G=(V,E)$ be a directed acyclic graph, with $V=\{v_i \mid i=1,\ldots,N\}$ the set of nodes and $E\subset V\times V$ the set of edges. The nodes are ordered in a way such that for each node v_i all parent nodes have indices lower than i. Let $pa_i=(j_1<\ldots<j_{n_i})$ index all parent nodes of v_i, and we assume that $v_1,\ldots,v_{n_0-1}$ are the root nodes of the graph, that is, $pa_i=\emptyset$ for $i<n_0$.

The associated Bayesian network (Lauritzen, 1996) is then given by a set of random variables $\boldsymbol{S}=(\boldsymbol{s}(i))_{i=1}^N$ such that the joint distribution decomposes as

$$\mathrm{p}(\boldsymbol{S})=\prod_{i=n_0}^{N}\mathrm{p}(\boldsymbol{s}(i)\mid\mathbf{Pa}(i))\prod_{i=1}^{n_0-1}\mathrm{p}(\boldsymbol{s}(i)). \tag{1}$$

Here, $\mathbf{Pa}(i)=(\boldsymbol{s}(j_1)',\ldots,\boldsymbol{s}(j_{n_i})')'$ with $j_1<\ldots<j_{n_i}$ denotes the vector of all random variables associated with the parent nodes of v_i.

From now we assume Gaussian random variables $(\boldsymbol{s}(i))_{i=1}^N$ with state space $\mathbb{R}^q$. We further assign edge weights $\lambda_{ij}\in\mathbb{R}$ to all edges $e_{ij}\in E$, and we denote the resulting weighted graph by $G_\Lambda=(V,E,\Lambda)$. Let $\boldsymbol{s}(i)$ and $\boldsymbol{s}(j)$ be random variables associated with adjacent nodes of the graph. We make the following stationarity (and scaling) assumptions:

$$\begin{aligned}&\text{(A1) } \mathbb{E}[\boldsymbol{s}(i)]=0_q,\\&\text{(A2) } \mathrm{Cov}(\boldsymbol{s}(i),\boldsymbol{s}(i))=I_q,\\&\text{(A3) } \mathrm{Cov}(\boldsymbol{s}(i),\boldsymbol{s}(j))=\lambda_{ij}D.\end{aligned}$$

The parameter D is constant over the network, and we call it the *graph-delayed covariance* of the stationary Gaussian model. According to our actual purpose of source separation, we assume that $D=\mathrm{diag}(d_1,\ldots,d_q)$ is a diagonal matrix. The larger an entry d_i the larger is the (absolute) value of the covariance between two adjacent random variables in the ith component.

With (A1)–(A3) and the decomposition of $\mathrm{p}(\boldsymbol{S})$ in Equation (1), all conditional distributions are uniquely defined, and we get

$$\boldsymbol{s}(i)\mid\mathbf{Pa}(i)\sim\begin{cases}\mathcal{N}(0_q,I_q) & \textit{if } \mathbf{Pa}(i)=\emptyset,\\ \mathcal{N}(\omega_D(i)\mathbf{Pa}(i),\Sigma_D(i)) & \textit{otherwise},\end{cases} \tag{2}$$

where $\omega_D(i)\in\mathbb{R}^{q\times q\,n_i}$ and $\Sigma_D(i)\in\mathbb{R}^{q\times q}$ depend only on the graph-delayed covariance D. If D is diagonal we have that $\Sigma_D\,(i)$ is also diagonal and $\omega_\mathrm{D}\,(i)$ consists of blocks of diagonal matrices. Dependent on the weighted graph G_Λ one can determine an interval $I^G \subseteq \mathbb{R}$ such that all covariance matrices $\Sigma_D\ (i)$ are positive definite for diagonal D with components in I^G. We refer to the Gaussian distribution p($\boldsymbol{S}$) defined (1) in Equations and (2) as *source model* $\mathcal{M}(G_\Lambda, q)$, where q is the dimension of the random variables. The distribution is parameterized by the graph-delayed covariance D, and if samples from $\mathcal{M}(G_\Lambda, q)$ are given, the maximum likelihood approach naturally yields an estimate $\hat{D}^{\text{ML}}$.

The concept of a graph-delayed covariance was originally introduced by Kowarsch et al. (2010). We now shortly review their definition. For a weighted directed graph $G_\mathcal{K} = (V, E, \mathcal{K})$ with weights $\kappa_{ij} \in \mathbb{R}$ and associated random variables $\boldsymbol{S} = (\boldsymbol{s}(i))_{i=1}^N$, they considered the covariance between a node and the weighted sum of its parent nodes as

$$D^{\text{Pa}} = \text{Cov}\left(\sum_{i \in pa_j} \kappa_{ij}\boldsymbol{s}(i), \boldsymbol{s}(j)\right), \tag{3}$$

and they assumed that it is independent of the index j. For $\kappa_{ij} = (|\mathbf{Pa}(j)|\lambda_{ij})^{-1}$ with λ_{ij} from above we have $D^{\text{Pa}} = D$, where D is the graph-delayed covariance in the stationary Gaussian model. To estimate D^{Pa} from samples $s(1), \ldots, s(N)$ they introduced

$$\hat{D}^{\text{Pa}} = \frac{1}{N - n_0 - 1}\sum_{j=n_0}^{N}\sum_{i \in pa_j} \kappa_{ij} s(i)\ s(j)', \tag{4}$$

where $n_0, \ldots, N$ are all indices of nonroot nodes. According to assumption (A3), we actually have more detailed information about the covariance between adjacent variables. We therefore additionally consider the following refined edge-based estimate:

$$\hat{D}^{\text{E}} = \frac{1}{|E| - 1}\sum_{(i,j)\in E} \frac{1}{\lambda_{ij}} s(i)\ s(j)'. \tag{5}$$

Both estimates $\hat{D}^{\text{Pa}}$ and $\hat{D}^{\text{E}}$ yield nonprobabilistic algorithms to separate network data—`Grade(Pa)` and `Grade(E)`. The former is the original method from Kowarsch et al. (2010), and in section 4.2 we shortly introduce both algorithms for estimation comparison.

2.2. *Graph models for simulations*

To illustrate the covariance structure of the stationary Gaussian model we introduce three graph models. All graphs are related to biological networks, or to time-series for comparison. In section 4 we again consider these graph models in our simulations about source separation.

(CC) Cell-cycle: The estimated network for the cell-cycle pathway based on gene expression data (Imoto et al., 2002). The network consists of 81 nodes and 84 edges.

(TF) Transcription factors: Three hub nodes and each directly signals on a subset of nodes.

(LL) Line signals: Similar to time-series, we define a network that consists of two line signals sharing the middle part, and one separated line signal.

Figure 1 illustrates these networks together with the associated covariance structure, and we randomly assigned weights ±1 to the edges. Note that one can theoretically consider any weights $\lambda_{ij} \in \mathbb{R}$.

3. *A Blind Source Separation Model for Mixed Network Data*

In blind source separation (BSS) we assume that we observe a linear mixture of the actual signals of interest. The aim is to estimate the mixing as well as the underlying signals. In the following we derive a new blind source separation method for network data and decribe the unobserved (latent) signals in terms of the source model from the last section.

3.1. *Linear Mixing Model*

We consider the following mixing model: $\boldsymbol{X} = (\boldsymbol{x}(i))_{i=1}^N$ are observed Gaussian variables with state space $\mathbb{R}^m$, and we assume latent Gaussian variables $\boldsymbol{S} = (\boldsymbol{s}(i))_{i=1}^N$ with state space $\mathbb{R}^q (q \leq m)$, such that each variable $\boldsymbol{x}(i)$ is a linear mixture of the components of the latent variable $\boldsymbol{s}(i)$:

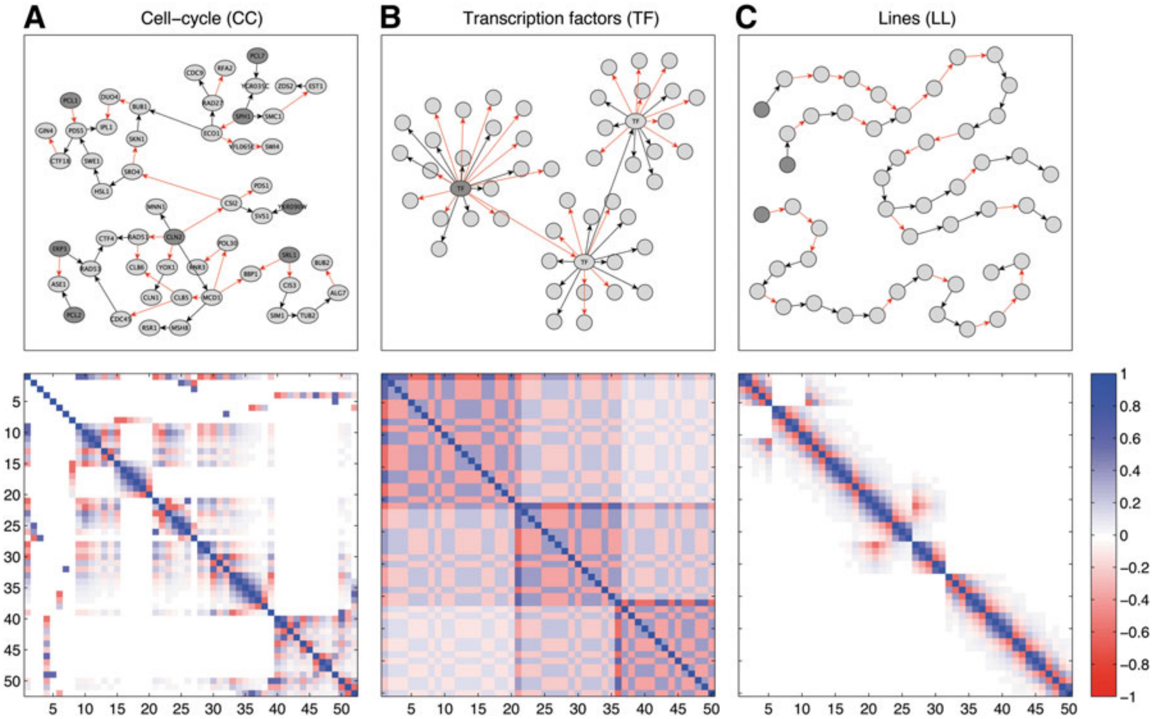

FIG. 1. Graph models and covariance structure. The upper graphics illustrate a connected subnetwork of cell-cycle (CC), transcription factors (TF), and lines (LL). Darker nodes indicate root nodes, and we randomly assigned edge weights with values +1 (black) and −1 (red). The lower graphics show the covariance structure associated with each graph model for one-dimensional random variables. The graph-delayed covariance was set to D = 0.6.

$$\boldsymbol{x}(i) = A\ \boldsymbol{s}(i) + \mu + \varepsilon(i), \qquad i = 1, \ldots, N. \tag{6}$$

Here, $\varepsilon(i)$ is additive noise distributed as $\varepsilon(i) \sim \mathcal{N}(0, \sigma^2 I_m)$ and independent of the latent variables. $A \in \mathbb{R}^{m \times q}$ denotes the mixing matrix and $\mu \in \mathbb{R}^m$ is a constant mean vector for all $\boldsymbol{x}(i)$. We refer to the components of the latent variables as *sources*, that is, for $k = 1, \ldots, q$ we have a source $\boldsymbol{s}_k = (\boldsymbol{s}_k(i))_{i=1}^N$.

We now extend the Bayesian network from section 2.1. Let $\boldsymbol{S}$ be *latent* variables, and the dependence is given by a weighted graph $G_\Lambda = (V, E, \Lambda)$ as before. We additionally introduce *observed* variables $\boldsymbol{X}$, where $\boldsymbol{x}(i) = A\boldsymbol{s}(i) + \mu + \varepsilon(i)$ for all i. The joint distribution of $\boldsymbol{X}$ and $\boldsymbol{S}$ then decomposes as

$$\mathrm{p}(\boldsymbol{X}, \boldsymbol{S}) = \prod_{i=1}^{N} \mathrm{p}(\boldsymbol{x}(i) \mid \boldsymbol{s}(i)) \prod_{i=n_0}^{N} \mathrm{p}(\boldsymbol{s}(i) \mid \mathbf{Pa}(i)) \prod_{i=1}^{n_0 - 1} \mathrm{p}(\boldsymbol{s}(i)), \tag{7}$$

where $\boldsymbol{x}(i) \mid \boldsymbol{s}(i) \sim \mathcal{N}(A\boldsymbol{s}(i) + \mu, \sigma^2 I_m)$ directly follows from the linear mixing. A graphical representation of this latent variable model is given in Figure 2a.

3.2. Parameter inference using expectation maximization

The unknown components of our model are the parameters $\theta = (A, \mu, \sigma, D)$ and the latent variables $\boldsymbol{S}$, and we are interested in both. A widely used approach for latent variable models is expectation maximization (McLachlan and Krishnan, 2007); that is, parameters and latent variables are updated alternately and each update improves the data log-likelihood $\ell(\theta;\ \boldsymbol{X}) = \ln \mathrm{p}(\boldsymbol{X}|\theta)$.

For expectation maximization we need to consider the complete data log-likelihood $\ell_c(\theta;\ \boldsymbol{X}, \boldsymbol{S}) = \ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S}|\theta)$. Let $\mathbb{E}_{S|X,\theta}[.]$ denote the expectation with respect to the posterior distribution $\boldsymbol{S} \mid \boldsymbol{X}$, θ of the latent variables $\boldsymbol{S}$ given the observable variables $\boldsymbol{X}$ and parameters θ. The expectation of the complete data log-likelihood is then given by

$$\mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S}|\theta)] = \mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{X}|\boldsymbol{S}, A, \mu, \sigma^2)] + \mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{S}|D)]. \tag{8}$$

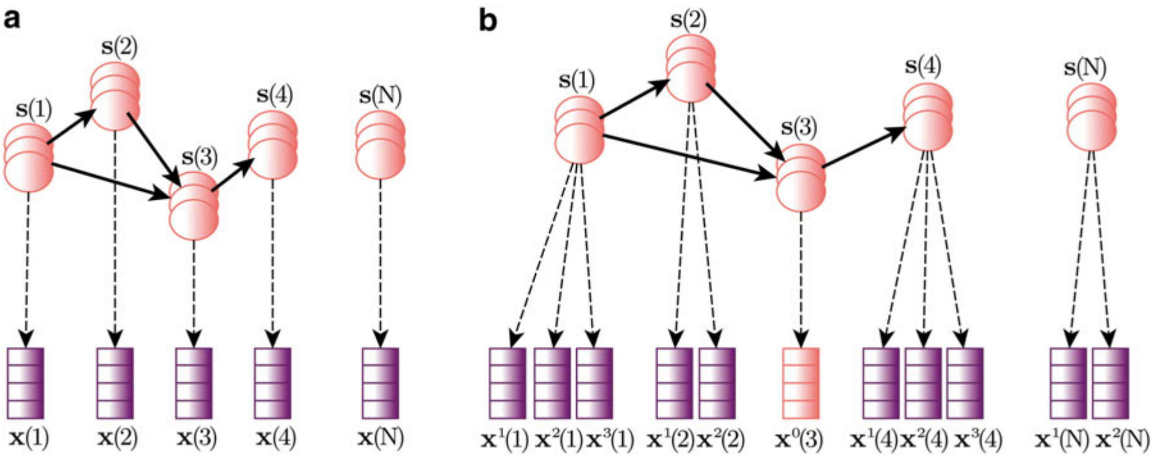

FIG. 2. Graphical representation of `emGrade`. Panel **(a)** shows the basic model with one observed variable $\boldsymbol{x}(i) = A\boldsymbol{s}(i) + \mu + \varepsilon(i)$ for all indices i. In Panel **(b)** we take into account multiple and/or missing observations. For all indices i we either have multiple observed variables $\boldsymbol{x}^r(i) = A\boldsymbol{s}(i) + \mu + \varepsilon^r(i)$ with $r = 1, \dots, r_i$, or a latent variable $\boldsymbol{x}^0(i) = A\boldsymbol{s}(i) + \mu + \varepsilon^0(i)$ when the observation for index i is missing. In both figures the observed part is shown in purple and the latent part in red.

For better readability we use the notations $A_\mu = (A, \mu)$ and $\boldsymbol{s}_*(i) = (\boldsymbol{s}(i)', 1)'$—both enlarged by a constant component. We then have:

$$\begin{aligned}\mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{X}|\boldsymbol{S}, A, \mu, \sigma^2)] = &-\frac{Nm}{2}\ln(2\pi) - \frac{Nm}{2}\ln(\sigma^2) \\ &- \frac{1}{2\sigma^2}\sum_{i=1}^{N}[\mathrm{Tr}(\mathbb{E}_{S|X,\theta}[\boldsymbol{x}(i)\boldsymbol{x}(i)']) \\ &- 2\mathrm{Tr}(\mathbb{E}_{S|X,\theta}[\boldsymbol{s}_*(i)\boldsymbol{x}(i)']A_\mu) \\ &+ \mathrm{Tr}(\mathbb{E}_{S|X,\theta}[\boldsymbol{s}_*(i)\boldsymbol{s}_*(i)']A'_\mu A_\mu)]\end{aligned} \tag{9}$$

$$\begin{aligned}\mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{S}|D) = &-\frac{Nq}{2}\ln(2\pi) - \sum_{i=n_0}^{N}\ln(\det(\Sigma_D(i))) \\ &- \frac{1}{2}\sum_{i=n_0}^{N}[\mathrm{Tr}(\mathbb{E}_{S|X,\theta}[\boldsymbol{s}(i)\boldsymbol{s}(i)']\Sigma_D(i)^{-1}) \\ &- 2\mathrm{Tr}(\mathbb{E}_{S|X,\theta}[\boldsymbol{s}(i)\mathbf{Pa}(i)']\Sigma_D(i)^{-1}\omega_D(i)) \\ &+ \mathrm{Tr}(\mathbb{E}_{S|X,\theta}[\mathbf{Pa}(i)\mathbf{Pa}(i)']\omega_D(i)'\Sigma_D(i)^{-1}\omega_D(i))] \\ &- \frac{1}{2}\sum_{i=1}^{n_0-1}\mathbb{E}_{S|X,\theta}[\boldsymbol{s}(i)\boldsymbol{s}(i)']\end{aligned} \tag{10}$$

The EM-algorithm consists of two steps that are repeated alternately until convergence. In the *E-step*, we determine the posterior distribution of the latent variables which yields the expectations in (9) and (10). Here, we use the well-known property $\mathbb{E}[z_1 z_2'] = \mathrm{Cov}(z_1, z_2) + \mathbb{E}[z_1]\mathbb{E}[z_2]'$ for random variables z_1 and z_2. If z_2 or both variables are observed, the l.h.s. equals $\mathbb{E}[z_1]z_2'$ and $z_1 z_2'$, respectively. To get these posterior estimates we use the junction tree algorithm implemented in the Bayes net toolbox for Matlab (Murphy et al., 2001). In the *M-step*, we maximize $\mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S})]$ with respect to the parameters. Due to our specific stationarity assumptions the toolbox is not applicable for parameter maximization. Let $\mathrm{Esx} = \sum_{i=1}^{N} \mathbb{E}_{S|X,\theta}[\boldsymbol{s}_*(i)\boldsymbol{x}(i)']$ and we define Ess and Exx accordingly. We have the following updates

$$A_\mu = (\mathrm{Esx})'(\mathrm{Esx})^{-1} \tag{11}$$

$$\sigma^2 = \frac{1}{Nm}[\mathrm{Tr}(\mathrm{Exx}) - 2\mathrm{Tr}(\mathrm{Esx}\ A_\mu) + \mathrm{Tr}(\mathrm{Ess}\ A'_\mu A_\mu)] \tag{12}$$

$$D = \text{numerical maximization}$$

The parameter updates for A, μ (in form of A_μ), and σ^2 can be derived directly from (9). The parameter D occurs as different rational terms in all $\omega_D(i)$ and $\Sigma_D(i)$. For all source models $\mathcal{M}(G)$, with G a weighted graph, one can theoretically derive formulas for the update of D. Since we consider many different graph models in our simulations (regarding structure, number of nodes, and egde weights), we use numerical maximization and do not provide explicit update formulas for D. The search space is given by the interval I^G, and since D, as well as $\omega_D(i)$ and $\Sigma_D(i)$, are (block-)diagonal we can maximize $\mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{S})]$ with respect to each component of D separately.

The proposed expectation-maximization scheme for the linear mixing model from section 3.1 provides a method to separate network data. Similarly to the separation assumptions of `Grade` (graph-decorrelation algorithm) we assume a diagonal matrix D, and we therefore call the new algorithm `emGrade` (expectation-maximization graph-decorrelation algorithm).

3.3. Repeated and missing observations

In a Bayesian network a random variable is either latent or observed. To take into account repeated and/or missing observations we redefine the graphical structure in Figure 2a. In addition to the latent variables $\boldsymbol{S}$ we introduce a latent variable $\boldsymbol{x}^0(i) = A\boldsymbol{s}(i) + \mu + \varepsilon^0(i)$ if the observation at index i is missing, and we introduce observed variables $\boldsymbol{x}^r(i) = A\boldsymbol{s}(i) + \mu + \varepsilon^r(i)$ for $r = 1, \ldots, r_i$ if we have r_i observations for index i (Fig. 2b):

$$\boldsymbol{X}^0 = \{\boldsymbol{x}^0(i) \mid \text{no observations at index } i\}, \tag{13}$$

$$\boldsymbol{X} = \{\boldsymbol{x}^1(i), \ldots, \boldsymbol{x}^{r_i}(i) \mid r_i \text{ observations at index } i\}. \tag{14}$$

In the E-step we infer $\boldsymbol{S}$ and $\boldsymbol{X}^0$ from the posterior distribution $\boldsymbol{S}, \boldsymbol{X}^0 \mid \boldsymbol{X}, \theta$. The expectation of the complete data log-likelihood decomposes into

$$\mathbb{E}_{S,X^0|X,\theta}[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{X}^0, \boldsymbol{S})] = \mathbb{E}_{S,X^0|X,\theta}[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{X}^0|\boldsymbol{S})] + \mathbb{E}_{S,X^0|X,\theta}[\ln \mathrm{p}(\boldsymbol{S})]. \tag{15}$$

In the following we investigate the impact of repeated and/or missing observations on the predictive power of the latent variable model. We assign the true parameters $\theta = (A, \mu, \sigma, D)$ to the model and infer source signals (E-step) from the posterior expectation $\mathbb{E}_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S}|\theta)]$. We then compare estimated and true source signals dependent on the number of repeated and/or missing observation values and dependent on the variance of the observation noise. As a distance measure we use

$$\mathrm{dist}(\hat{S}, S) = \frac{1}{\sqrt{qN}} \left\|\hat{S} - S\right\|_F, \tag{16}$$

where $||.||_F$ denotes the Frobenius norm of a matrix. In Figure 3 we generated data from model (CC) with weights +1 and random parameters $\theta = (A, \mu, \sigma^2 D)$. Expectedly, we find a better source recovery if we have many repeated and no missing observations as well as a low noise level. The performance also increases if the dimension of the latent variables is smaller than the dimension of the observed variables (A-B), and disregarding the graph structure of the observations yields a worse performance (C). For the other graph models, (TF) and (LL), the results are similar.

3.4. Separation of subnetworks

Until now we assumed one q-dimensional source model $\mathcal{M}(G, q)$ to jointly model all source signals $\boldsymbol{s}_k = (\boldsymbol{s}_k(i))_{i=1}^N$ for $k = 1, \ldots, q$. If we have more detailed information about single components (pathways) of a larger network, and we want to separate the data according to these subnetworks, we can model the distribution of each source signal separately. Let therefore $P_1, \ldots, P_q$ be weighted graphs on the same set of nodes. For each source we consider the one-dimensional source model $\mathcal{M}(P_i, 1)$ with graph-delayed covariance $d_i \in \mathbb{R}$. We then assume that the joint distribution of S decomposes as

$$\mathrm{p}(\boldsymbol{S}) = \prod_{k=1}^{q} \mathrm{p}(\boldsymbol{s}_k) = \prod_{k=1}^{q} \prod_{i=1}^{N} \mathrm{p}(\boldsymbol{s}_k(i) \mid \mathbf{Pa}_k(i)). \tag{17}$$

We denote this source model based on q different pathways as $\mathcal{M}(P_1, \ldots, P_q)$. If all pathways are identical (i.e., $P_i = G$ for all i), the above definition yields the original source model $\mathcal{M}(G, q)$ with diagonal graph-delayed covariance $D \in \mathbb{R}^{q \times q}$. The new definition only effects the expectation step; in the graphical

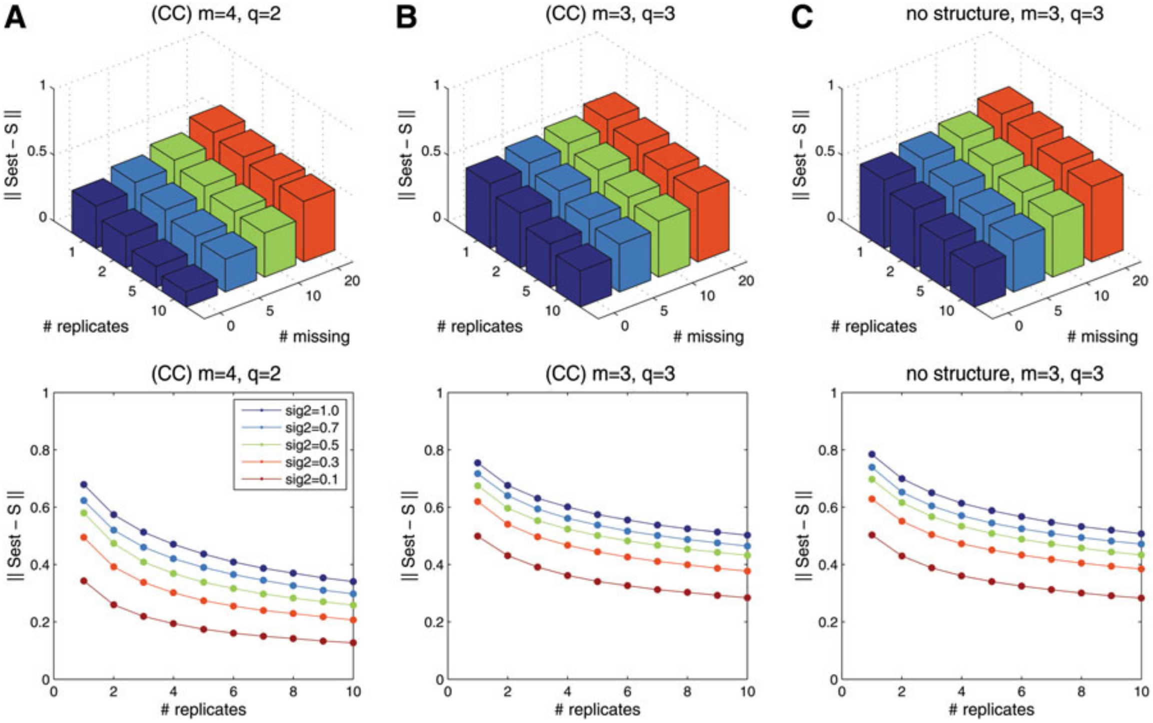

FIG. 3. Source recovery from repeated observations with missing values and observation noise. We generate repeated data from model (CC). In the upper plots we then ignore up to 20 observed variables (missing values), and in the bottom plots we add noise from $\mathcal{N}(0, \sigma^2)$ for $\sigma^2 = 0.1, 0.3, 0.5, \ldots, 1.0$ to all entries. We infer the source signals from the posterior distribution, where the data and the true parameters are given. The plots show the mean difference over 100 runs between the original source signals and the estimates. In **(A)** we fix the dimensions at $m = 4$ (observed variables) and $q = 2$ (latent variables), in **(B)** we have $m = q = 3$. In **(C)** for comparison, we disregard the structure of the data and consider a network without edges for source estimation.

representation we split the node for $\boldsymbol{s}(i)$ into q nodes representing the one-dimensional random variables $\boldsymbol{s}_1(i), \ldots, \boldsymbol{s}_q(i)$ for $i = 1, \ldots, N$. Again, the junction tree algorithm provides posterior estimates of the latent variables $\boldsymbol{S}$. In the last section we use this proper source modeling to identify pathways that are most likely present in the data.

4. PERFORMANCE AND FEATURES OF EMGRADE

In this part we evaluate the performance of emGrade and demonstrate some gains from the Bayesian modeling—we introduce a family of information criteria to determine number and structure of the unknown source signals. For all simulations we consider the graph models from section 2.2 and fix the egde weights at +1.

4.1. *Convergence of emGrade*

We define convergence in terms of changes in the single parameter estimates rather than changes in the log-likelihood function. This approach was suggested in Abbi et al. (2008), since convergence of the single parameters usually requires more EM-iterations (when the same threshold is considered). We say that emGrade has converged if

$$\forall i \ |\theta_i - \theta_i^{(new)}| < 10^{-8}, \tag{18}$$

with parameters $\theta = (A, \mu, \sigma^2, D)$. The current parameter values are then the estimation result, and we call a run *nonconvergent* if there is no convergence after 10,000 EM-iterations. To get an impression of the convergence behavior of emGrade we generated data from model (CC), and we considered different combinations of $m, q = 1, \ldots, 5$ (dimension of latent/observed variables). Figure 4 shows that for $q < m$ the

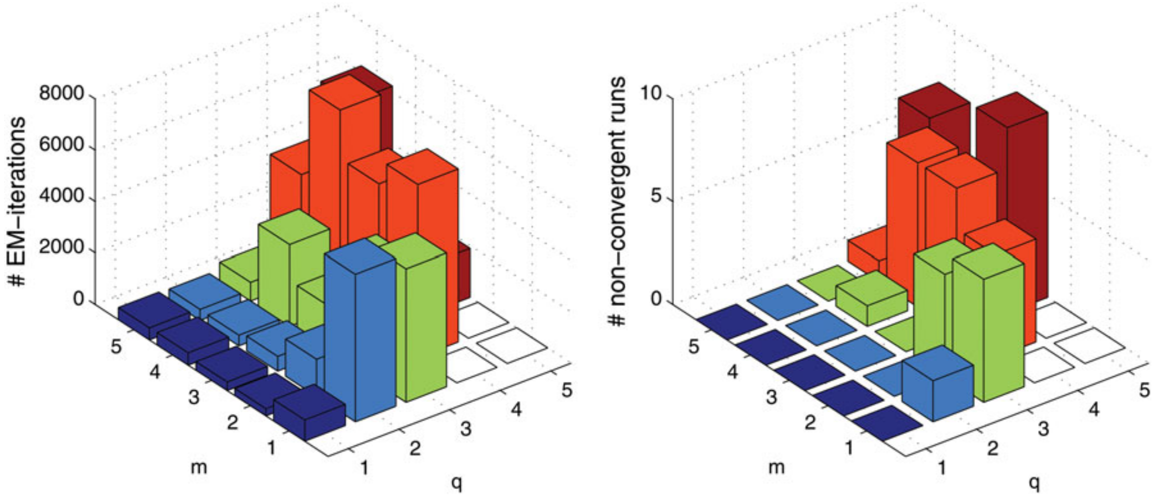

FIG. 4. Number of EM-iterations. We generate data from model (CC) and consider different combinations of $m, q = 1, \ldots, 5$ (dimension of observed/latent variables). The plots show the mean number of EM-iterations among all convergent runs (left) and the number of nonconvergent runs (right). For all *m/q*-combinations with $q \geq m+2$ we performed 10 runs of `emGrade`.

mean number of EM-steps is small, and for $q > m$ or large dimensions $q = m$ the number of EM-steps explodes and many runs do not converge at all. In blind source separation applications we usually assume $q \leq m$, and we limit ourselves to such cases in the following.

4.2. *Algorithm comparison*

We now compare `emGrade` to other blind source separation algorithms. The most similar algorithm regarding model assumptions is `Grade`, which diagonalizes the sample graph-delayed covariance using singular value decomposition. We distinguish between two versions—`Grade(Pa)` and `Grade(E)`, where the graph-delayed covariance is estimated as $\hat{D}^{\mathrm{Pa}}$ and $\hat{D}^{\mathrm{E}}$ (section 2.1). We further consider algorithms for time-series data and simply resume the order of the random variables. `AMUSE` (Tong et al., 1990) and `SOBI` (Belouchrani et al., 1997) both assume stationarity (of the time-series), i.e. the autocovariance Cov $(\boldsymbol{x}(i), \boldsymbol{x}(i+\tau))$ is independent of the index i at any lag $\tau \in \mathbb{Z}$. The algorithms then diagonalize sample autocovariances at one or multiple lags, respectively. Finally, we compare our results to `PCA` and `fastICA` (Hyvärinen and Oja, 2000), both act independently of the structure of the random variables.

All algorithms provide estimates of the mixing matrix and the source signals—but unlike `emGrade` they do not estimate the full parameter vector. $\theta = (A, \mu, \sigma^2, D)$. Instead of a likelihood-based evaluation of the performance we use the distance of estimated and true mixing matrix as performance measure:

$$\mathrm{dist}(\hat{A}, A) = \min_{p \in \mathcal{P}} \frac{1}{\sqrt{mq}} \left\| \hat{A}P - A \right\|_F. \tag{19}$$

Let $\mathcal{P} \subseteq M\ at(q, q)$ be the set of all $q \times q$ matrices with one nonzero entry per row and column and this entry has value ± 1. With this we correct for possible permutation and sign-changing of the mixing columns. If m-dimensional data is given all algorithms—except `emGrade`—estimate a mixing matrix in $Mat(m,m)$. In case of $q < m$ we only use the first q columns of the mixing estimates. In Figure 5 we compare the estimation performance of all algorithms, and we fix the dimensions at $m = 3$ and $q = 2$. For all proposed graph models `emGrade` outperforms the other algorithms in terms of correctness of the estimates, the drawback is a much higher run-time. In the case of $m = q$ the improvement of the estimates is less apparent.

4.3. *Pathway identification and number of source signals*

We now take full advantage of the probabilistic modeling and use model selection criteria to determine the correct number of source signals and to identify active pathways in the network. Let $\mathcal{M}(G, q)$ denote the source model with q source signals, and the joint distribution is based on a weighted graph G. We then consider the following information criterion:

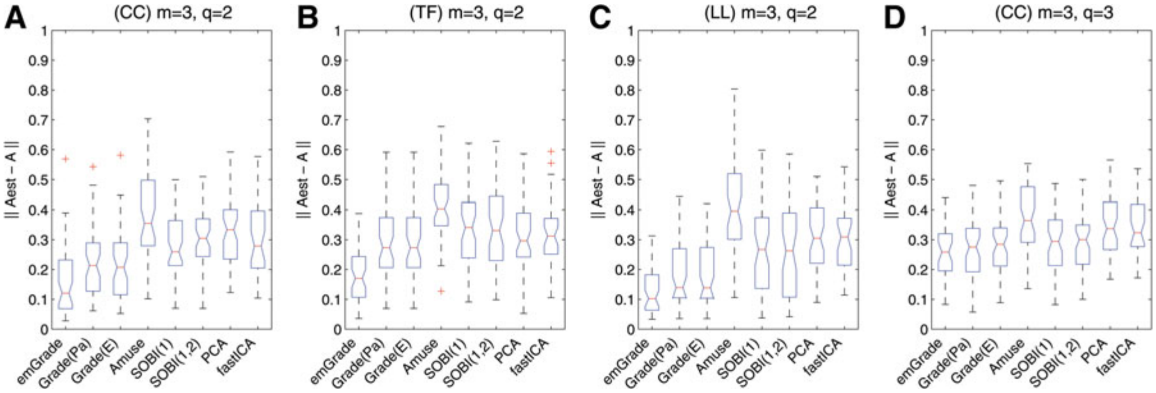

FIG. 5. Algorithm comparison. The plots show the mean performance over 50 runs of the algorithms `emGrade`, `Grade(G)`, `Grade(E)`, `AMUSE`, `SOBI` (at lag 1 and at lags 1,2), `PCA`, and `fastICA`. In **(A)**–**(C)** we generated data from models (CC), (TF), and (LL) with $m=3$ and $q=2$. In this case $(m>q)$ `emGrade` yields the best estimation performance for all graph models. In **(C)** where $m=q=3$ the improvement compared to the other algorithms is smaller.

$$\mathrm{IC}(\mathcal{M}(G,q)) = -2\ell(\theta; X, \mathcal{M}(G,q)) + kc, \tag{20}$$

where k denotes the number of model parameters, and c is some constant. For $c=2$ the above equation yields the Akaike information criterion (AIC) (Akaike, 1974) and for $c=\ln(N)$ the equation yields the Bayesian information criterion (BIC) (Schwarz, 1978) with N the number of observed variables. To determine the true number of source signals we fix the graph G and search for the lowest IC value among different source models. Since the comparison is based on the log-likelihood value of the `emGrade` estimates we use $|\ell(\theta;\ \boldsymbol{X}) - \ell(\theta^{(new)};\ \boldsymbol{X})| < 10^{-6}$ as convergence criterion. In Figure 6 we generated data from model (CC) with $q=3$ the true number of source signals (dimension of the latent variables) and $m=3$, 4, 5 observations (dimension of the observed variables). We then compare the IC values of $M(\hat{q}) = \mathcal{M}((CC), \hat{q})$ for $\hat{q}=1, \ldots, 5$, where we consider different constant values c. In case of $m > q$ we find a nearly perfect estimation of the true number of source signals for $c=2$ (AIC) and $c=\ln(N)$ (BIC).

For pathway identification we divide each network from section 2.2 into three pathways (subnetworks) P_1, P_2, and P_3. For (CC) we define the pathways as the three connected components of the complete cell-cycle network, for (TF) we consider the single hub-nodes together with their target nodes as pathways, and for (LL) we consider the two overlapping lines and the additional line as pathways. We then generate data from the pathway source model $\mathcal{M}(P_i, P_j)$ introduced in section 3.4, and we determine the lowest IC value among all source models $M(i,j) = \mathcal{M}(P_i, P_j)$ for $i,j=1,2,3$. If the edges of the pathways are nonoverlapping [models (CC) and (TF)] we observe a good pathway identification; for model (LL) often only one pathway is identified correctly (Fig. 7).

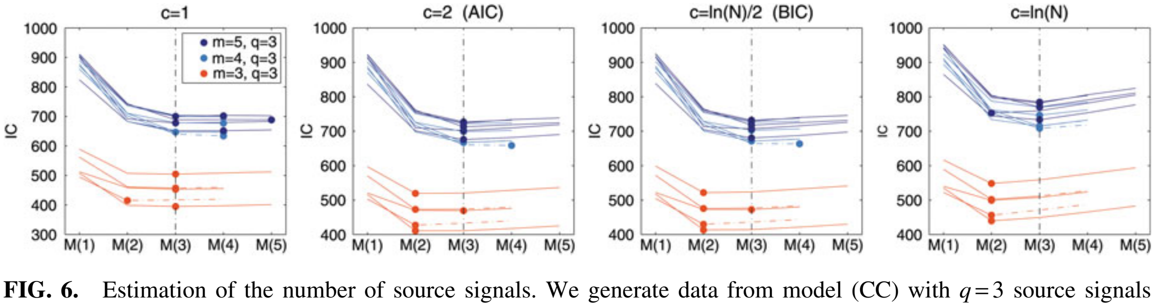

FIG. 6. Estimation of the number of source signals. We generate data from model (CC) with $q=3$ source signals (dimension of the latent variables) and m-dimensional observations for $m=5,4,3$ in different colors. For the graph-delayed covariance we consider $D=[0.7, -0.49, 0.3]$. The plots show the IC values of all source models $M(\hat{q}) = \mathcal{M}((\mathrm{CC}), \hat{q})$ for increasing $\hat{q}=1, \ldots, 5$. In each plot we consider a different constant c. The black vertical lines show the true number of source signals and the dots indicate the selected source model in each comparison, that is, the model with the lowest IC value. Dashed lines lead to IC values of nonconvergent runs (using the log-likelihood value after 10,000 iterations), and some IC values are $+\infty$. IC, information criterion.

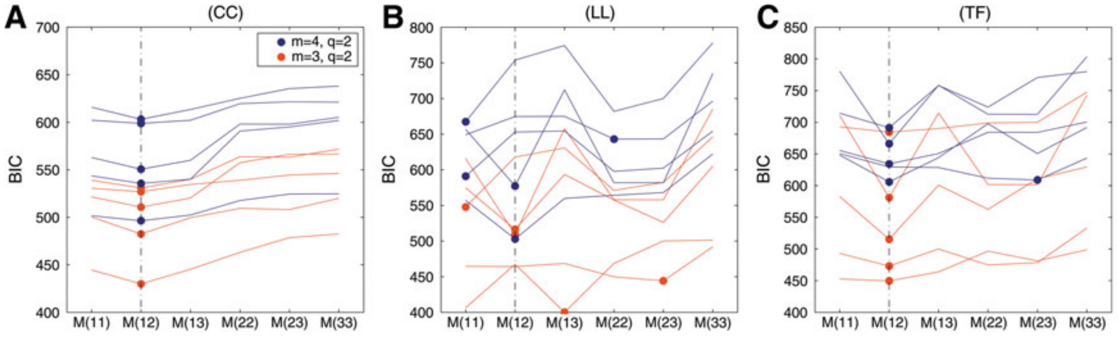

FIG. 7. Pathway identification. We generate $q = 2$ source signals from the respective pathways P_1 and P_2 of the models (CC), (LL), and (TF). From observations of dimension $m = 4,3$ (in different colors) we calculate the BIC of all pathway source models $M(i, j) = \mathcal{M}(P_i, P_j)$, where we assume source signals from pathways P_i and P_j ($i,j = 1,2,3$). The black vertical lines show the true pathway combination, and the dots indicate the selected source model in each comparison, that is, the model with the lowest BIC value. BIC, Bayesian information criterion.

5. DISCUSSION AND CONCLUSION

In this work we defined the distribution of signaling data in terms of a stationary Bayesian network with Gaussian random variables. Based on this definition, we proposed the probabilistic blind source separation algorithm `emGrade` to determine underlying signals of interest in a multivariate mixture. The iterative expectation maximization procedure for parameter and source inference achieved good convergence in small dimensions. Moreover, we were able to determine the true number of source signals and to identify the correct active pathways (subnetworks) in simulations. The separate modeling of each source signal according to different specific pathways might be seen as a key adventage of our method. In our ongoing work we consider gene expression data in which we assume that the observations consist of a mixture of biological processes. For different combinations of literature-derived pathways we compare the BIC values of our model. With this we want to determine the active processes and compare different data sets (e.g., treatment versus control) in terms of a change in the underlying processes. In addition, we want to biologically validate our method, that is, we want to show that the literature-derived network information yields an improved separation of the data. Here, we consider knock-down experiments in which a specific known pathway is not present in the knock-down data set. Using enrichment analysis to assign biological processes to the estimated source signals we can qualify the estimation performance of different BSS methods in a biological manner. With this we want to continue the findings from Kowarsch et al. (2010) and strengthen the relevance of network-based BSS methods in biological applications.

ACKNOWLEDGMENTS

This work was financially supported by the German Federal Ministry of Education and Research (BMBF) within the GerontoSys project ‘‘Stromal Aging’’ (Grant No. FKZ 0315576C), the German Research Foundation (DFG) within the project InKoMBio (Grant No. BO 3834/1-1) and the European Union within the ERC grant ‘‘LatentCauses’’.

AUTHOR DISCLOSURE STATEMENT

The authors declare that no competing financial interests exist.

REFERENCES

Abbi, R., El-Darzi, E., Vasilakis, C., et al. 2008. Analysis of stopping criteria for the EM algorithm in the context of patient grouping according to length of stay. *Proc. 4th Int. IEEE Conf. on Intell. Syst.* 1, 3–9.

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Trans. Automat. Contr.* 19, 716–723.

Belouchrani, A., Abed-Meraim, K., Cardoso, J.-F., et al. 1997. A blind source separation technique using second-order statistics. IEEE Trans. Signal Processing 45, 434–444.

Friedman, N., Linial, M., Nachman, I., et al. 2000. Using Bayesian networks to analyze expression data. *J. Comput. Biol.* 7, 601–620.

Hyvärinen, A., and Oja, E. 2000. Indepedent component analysis: algorithms and applications. *Neural Net*. 13, 411–430.

Illner, K., Fuchs, C., and Theis, F.J. 2012. Blind source separation using latent gaussian graphical models. Proc. 9th Int. Workshop on Comput. Syst. Biol., 43–46.

Illner, K., Fuchs, C., and Theis, F.J. 2014. Bayesian blind source separation applied to the lymphocyte pathway. Proc. 21st Int. Conf. on Comput. Stat., 625–632.

Imoto, S., Goto, T., and Miyano, S. 2002. Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. *Pac. Symp. Biocomput*. 7, 175–186.

Kowarsch, A., Blöchl, F., Bohl, S., et al. 2010. Knowledge-based matrix factorization temporally resolves the cellular responses to IL-6 stimulation. BMC Bioinformatics 11, 585–598.

Lauritzen, S.L. 1996. *Graphical Models*. Oxford University Press, Oxford, United Kingdom.

McLachlan, G.J., and Krishnan, T. 2007. *The EM Algorithm and Extensions*. John Wiley & Sons, Hoboken, NJ.

Murphy, K., et al. 2001. The Bayes net toolbox for Matlab. *Computing Science and Statistics* 33, 1024–1034.

Schwarz, G. 1978. Estimating the dimension of a model. *Ann. Stat.* 6, 461–464.

Tong, L., Soon, V.C., Huang, Y.F., et al. 1990. AMUSE: A new blind identification algorithm. *Proc. IEEE Int. Symp. on Circuits and Syst.*, 1784–1787.

Address correspondence to:
Fabian J. Theis
Helmholtz Research Center Munich
Institute of Computational Biology
Ingolstädter Landstrabe 1
Neuherberg 85746
Germany

E-mail: fabian.theis@helmholtz-muenchen.de