BIOINFORMATICS ORIGINAL PAPER Vol. 27 no. 2 2011, pages 238–244
doi:10.1093/bioinformatics/btq631

Systems biology Advance Access publication November 10, 2010

Fast and efficient dynamic nested effects models

Holger Fröhlich[1,*], Paurush Praveen[1] and Achim Tresch[2]

[1]Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn-Aachen International Center for IT, Dahlmannstrasse 2, 53113 Bonn and [2]Department of Chemistry and Biochemistry, Ludwig-Maximilians-Universität München, Gene Center Munich and Center for integrated Protein Science CiPSM, Feodor-Lynen-Strasse 25, 81377 Munich, Germany

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Targeted interventions in combination with the measurement of secondary effects can be used to computationally reverse engineer features of upstream non-transcriptional signaling cascades. *Nested effect models* (NEMs) have been introduced as a statistical approach to estimate the upstream signal flow from downstream nested subset structure of perturbation effects. The method was substantially extended later on by several authors and successfully applied to various datasets. The connection of NEMs to Bayesian Networks and factor graph models has been highlighted.

Results: Here, we introduce a computationally attractive extension of NEMs that enables the analysis of perturbation time series data, hence allowing to discriminate between direct and indirect signaling and to resolve feedback loops.

Availability: The implementation (R and C) is part of the Supplement to this article.

Contact: frohlich@bit.uni-bonn.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on August 23, 2010; revised on October 18, 2010; accepted on November 5, 2010

1 INTRODUCTION

Reverse engineering of biological networks is a key for the understanding of biological systems. The exact knowledge of interdependencies between proteins in the living cell is crucial for the identification of drug targets for various diseases. However, due to the complexity of the system a complete picture with detailed knowledge of the behavior of individual proteins is still out of reach. Nonetheless, the advent of gene perturbation techniques like RNA interference (RNAi) (Fire *et al*., 1998), opened new perspectives for network reconstruction by boosting the ability to subject organisms to well-defined interventions.

In general, biological networks are represented mathematically by a graph in which the nodes represent biological entities and the edges encode functional relations (of various kinds) between these entities. A number of approaches have been proposed in the literature for estimating networks from perturbation effects. Many of these aim at reconstructing a network from directly observable effects. For example, Rung *et al*. (2002) build a directed disruption graph by drawing an edge (i,j), if gene i results in a significant expression change at gene j. Wagner (2001) uses such disruption networks as a starting point for a further graph-theoretic method, which removes indirect effects (Aho *et al*., 1972), hence making the network more parsimonious. Tresch *et al*. (2007) and Klamt *et al*. (2010) enhance this approach by additionally making use of edge probabilities and signs to make the network consistent with the observed biological effects.

Moreover, Bayesian Networks have been used to model the statistical dependency between perturbation experiments (Maathuis *et al*., 2009, 2010; Pe'er *et al*., 2001; Sachs *et al*., 2005). For this purpose Pearl (2000) introduces his *do* calculus, an idealized model of interventions. Other probabilistic graphical models that have been proposed to model interventional data comprise factor graphs (Gat-Viks *et al*., 2006) and dependency networks (Rogers and Girolami, 2005). In genetics, epistasis analysis is often been applied for learning from indirect downstream effects. For example, Driessche *et al*. (2005) use expression profiles from single and double knockdowns to partly reconstruct a developmental pathway in *Dictyostelium discoideum* via a simple cluster analysis. Kanabar *et al*. (2009) embedds epistasis analysis into a statistical framework. Battle *et al*. (2010) propose a special kind of Bayesian Network to infer pathway structures from genetic interaction data. All of these approaches deal with static data. However, with the increasing availability of time-resolved data, it is desirable and helpful to model the temporal aspects of signaling. To this end, differential equation systems have been suggested e.g. by Nelander *et al*. (2008).

Our approach is an expansion of *Nested effects models* (NEMs) (Anchang *et al*., 2009; Fröhlich *et al*., 2007, 2008, 2009; Markowetz *et al*., 2005, 2007; Tresch and Markowetz, 2008; Vaske *et al*., 2009; Zeller *et al*., 2009). NEMs are specifically designed to learn the signaling flow—on a transcriptional as well as non-transcriptional level—between perturbed genes from indirect, high-dimensional effects, typically monitored by DNA microarrays. NEMs use a graphical modeling framework to compare a given network hypothesis with the observed nested structure of downstream effects. In conclusion, NEMs estimate an interaction network between hidden entities from a set of observable downstream effects. NEMs have been applied successfully to a number of datasets, e.g. the immune response in *Drosophila melanogaster* (Markowetz *et al*., 2005), a transcription factor network in *Saccharomices cerevisiae* (Markowetz *et al*., 2007), the ER-α pathway in human breast cancer cells (Fröhlich *et al*., 2007, 2008), a synthetic lethality interactions network in *S.cerevisiae* (Zeller *et al*., 2009), murine stem cell development in mice (Anchang *et al*., 2009) and the Rosetta dataset for *S.cerevisiae* (Vaske *et al*., 2009).

*To whom correspondence should be addressed.

This article complements the attempt of Anchang *et al*. (2009) to extend static NEMs to the modeling of perturbation time series measurements. Most importantly, this allows for the resolution of feedback loops in the signaling cascade, as well as for the discrimination of direct and indirect signalling. In contrast to Anchang *et al*. the key idea in our model is to unroll the signal flow over time. This allows for a computation showing some similarity to Dynamic Bayesian Networks and naturally extends the classical NEM formulation introduced by Markowetz *et al*. Our model circumvents the need for time consuming Gibbs sampling, which makes it also computationally attractive.

2 METHODS

2.1 NEMs

For self-containedness, we start with a brief review of the original statistical inference framework by Markowetz *et al*. (2005). In this framework, one distinguishes between silenced/perturbed entities (genes or proteins, which cannot be observed directly), called S-genes ($\mathcal{S}$) and other entities (genes or proteins) showing a measurable downstream effect (E-genes, $\mathcal{E}$). The idea of NEMs is to separate the upstream signaling pathway (the graph connecting the S-genes) from the downstream signaling (the graph connecting S-genes to E-genes). Consequently, the edge set of a NEM is partitioned into a directed edge set connecting S-genes among themselves, and another one describing the connections between the S-genes and the E-genes. It is assumed that each E-gene is attached to at most one S-gene only. Knocking down a specific S-gene S_k interrupts the signal flow in the downstream pathway, and hence an effect on the E-genes attached to S_k or one of the S-genes depending on S_k is expected. Perturbing each S-gene once will thus result in a nested subset structure of effects, which can be used to reverse engineer the upstream signal flow graph (Fig. 1).

The linking of E-genes to S-genes is formally represented by a binary $|\mathcal{S}|\times|\mathcal{E}|$ matrix $\boldsymbol{\Theta}=(\Theta_{ij})$ with $\Theta_{ij}=1$, if effect j is linked to S-gene i and $\Theta_{ij}=0$ otherwise. The cross-talk between S-genes (our network hypothesis) is given by a binary $|\mathcal{S}|\times|\mathcal{S}|$ matrix $\boldsymbol{\Phi}=(\Phi_{ij})$, with $\Phi_{ij}=1$ whenever S-gene i is upstream of S-gene j, and the convention $\Phi_{ii}=1$ for all $i\in\mathcal{S}$. A (static) NEM assumes that perturbing S-gene $s\in\mathcal{S}$ leads to an observable downstream effect for E-gene $e\in\mathcal{E}$, if there is a path from s to e, i.e. there exists an $s'\in\mathcal{S}$, such that $\Phi_{ss'}=1$ and $\Theta_{s'e}=1$. Hence, matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ together determine whether a perturbation of $s\in\mathcal{S}$ has an effect on a E-gene $e\in\mathcal{E}$. Similarly, if several S-genes $S\subseteq\mathcal{S}$ have been perturbed simultaneously, an E-gene is affected if there is a path from at least one of the S-genes in S to the respective E-gene. Suppose we have conducted a set $\mathcal{K}$ of perturbation experiments. For each experiment $k\in\mathcal{K}$, let $S_k\subseteq\mathcal{S}$ denote the set of S-genes that has been actively perturbed. Note that at this point we generalize the original formulation of NEMs (Markowetz *et al*., 2005) in a way that allows for the integration of combinatorial perturbations.

Let $\boldsymbol{D}=(D_{kj})$ be a $|\mathcal{K}|\times|\mathcal{E}|$ matrix of experimentally observed effects. The entry of D_{kj} is a collection of all observations of E-gene j under the perturbation(s) of S_k. It may consist of counts of how often a specific gene showed a knockdown effect among ℓ experiment repetitions, as originally proposed by Markowetz *et al*. (2005). Alternatively, Fröhlich *et al*. (2007, 2008) and Tresch and Markowetz (2008) suggested using log P-value densities and log odds ratios, respectively, which are closely related to each other (Fröhlich *et al*., 2009).

In general, a NEM model can be scored according to a likelihood function of the form:

$$p(\boldsymbol{D}\,|\,\boldsymbol{\Phi},\boldsymbol{\Theta})=\prod_{i\in\mathcal{E}}\prod_{k\in\mathcal{K}}p(D_{ik}\,|\,\Phi,\Theta) \qquad (1)$$

The local likelihoods $p(D_{ik}\,|\,\Phi,\Theta)$ are the building blocks of the NEM. A meaningful way to define these local likelihoods and calculate them efficiently is described in the Supplementary Material.

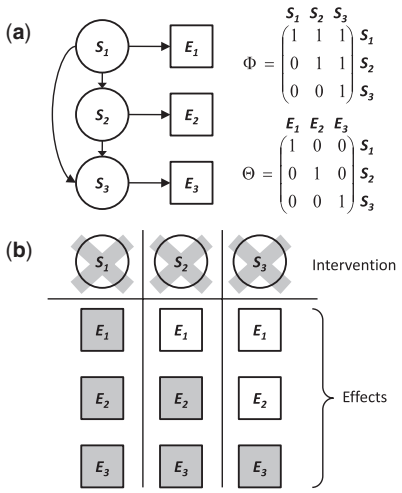

Fig. 1. Main idea of the inference framework by Markowetz *et al*. (**a**) A static NEM is parametrized by a directed graph between S-genes encoded by $\boldsymbol{\Phi}$, together with a directed graph attaching each E-gene to an S-gene given by $\boldsymbol{\Theta}$. (**b**) Relation of perturbations and observable effects. A perturbation of S-gene S_2 affects the downstream signaling pathway (S_3), and hence an effect on the E-genes attached to S_2 and to S_3, namely E_2 respectively E_3, is predicted (grey shading).

2.2 Dynamic NEMs (dynoNEMs)

The above model monitors static perturbation effects. Specifically, a perturbation signal is supposed to propagate deterministically through the whole S-gene network $\boldsymbol{\Phi}$. Cycles in $\boldsymbol{\Phi}$ imply that perturbation effects are indistinguishable within this model. As a matter of principle, it is impossible to detect feedback loops in $\boldsymbol{\Phi}$. Thus, it is highly desirable to have time series measurements of perturbation effects, which help resolve biological feedback loops and distinguish direct from indirect effects.

From now on we suppose our data $\boldsymbol{D}$ to consist of a time series, i.e. $\boldsymbol{D}=\{D_{ij}(t)\,|\,t=1,...,T\}$. These measurements could be P-values, counts or any other kind of statistics quantifying the effect of a knockdown for E-gene i under perturbation of S-gene j at time t (see Supplementary Material). The time variable t here denotes the *index* of a time point in a discrete time series, but not the time point itself.

As an example, suppose the biological 'truth' is given by the signaling pathway shown in Figure 1a. We unroll the signal flow in this network over time (Fig. 2) in the following way: the node set $\mathcal{E}(t)=\{E(t),E\in\mathcal{E}\}$, $\mathcal{S}(t)=\{S(t),\ S\in\mathcal{S}\}$ of the dynamic network consists of a copy of the static network nodes, one for each timepoint $t=1,...,T$. An E-gene $E(t)$ is linked to $S(t)$ whenever E is linked to S in the static situation, i.e. it is determined by the same matrix $\boldsymbol{\Theta}$ as in the static case. The actual unrolling takes place in the wiring of the S-genes. Informally, the static adjacency matrix $\boldsymbol{\Phi}$ is converted to a $|\mathcal{S}|\times|\mathcal{S}|$ weighted adjacency matrix $\boldsymbol{\Psi}=(\Psi_{ij})$, where 0 means no edge and a value $\Psi_{ij}>0$ implies an influence of node i on E-genes downstream of node j delayed by Ψ_{ij} time steps. Specifically, we have $T\geq\Psi_{ij}\geq\Phi_{ij}$ for $i,j\in\mathcal{S}$. A non-zero entry $\Psi_{ij}\neq0$ implies that there are edges $S_i(t)\rightarrow S_j(t+\Psi_{ij})$, $t=1,\ldots,T-\Psi_{ij}$. Furthermore, we make the convention $\Psi_{ii}=1$. Please note again that we neither aim to model physical signaling nor downstream transcription times. Instead, a positive time lag between nodes i and j in our model describes the number of time steps, after which a knockdown of node i results in an *observed* effect downstream of node j. This specifically implies that we do not need any assumptions about the physical time it takes a signal at node j to produce a downstream effect at an E-gene.

In contrast to classical Dynamic Bayesian Networks (Ghahramani, 1997), an edge in our model may not connect consecutive time layers, but it may

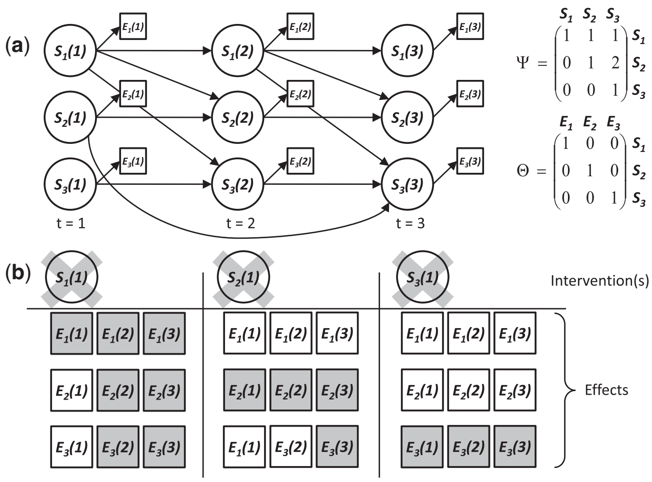

Fig. 2. Unrolling of the signal flow in the network from Figure 1 along time (t = time index). (**a**) Network topology and parametrization of the dynamic NEM. (**b**) Predicted effects (grey shading) resulting from perturbations.

skip a certain amount of time steps (as it is the case for the entry $\Psi_{S_2S_3}=2$ in Figure 2, which implies the edge $S_2(1)\rightarrow S_3(3)$. In other words, we do not rely on a first order Markov assumption in our model. In this way, we model the unknown and variable time delays in perturbation responses due to the upstream signaling. In the following, we refer to this model as *dynoNEM*. The likelihood formula for dynoNEMs is given as (c.f. Supplementary Material):

$$p(\boldsymbol{D}\,|\,\Psi)=\prod_{i\in\mathcal{E}}\sum_{s\in\mathcal{K}}\prod_{k\in\mathcal{K}}\prod_{t=1}^{T}p(D_{ik}(t)\,|\,\boldsymbol{\Psi},\Theta_{is}=1)\Pr(\Theta_{is}=1) \tag{2}$$

To compute $p(D_{ik}(t)\,|\,\boldsymbol{\Psi},\Theta_{is}=1)$ according to the proposed unrolling of the signal flow, we introduce a time-dependent Boolean perturbation state for each S-gene s, which we encode with 0 (perturbed, S-gene is underexpressed or overexpressed) and 1 (unperturbed, S-gene is naturally expressed), respectively. A knockdown of s corresponds to a switch $1\rightarrow 0$. Since the perturbation state of s at a particular time step t is not observable, we identify it with the value $[s(t)]$ of a random variable $s(t)$. Let $pa(s)(t)$ denote the set of parents nodes of s at time t (i.e. the set $\{p\,|\,0<\Psi_{ps}<t\}$ —which can be empty, if appropriate). Then, according to the unrolling of the signal flow over time, we write:

$$\begin{aligned} p(D_{ik}(t)\;|\;\boldsymbol{\Psi},\Theta_{is}=1)= \\ \sum_{[s(t)]\in\{0,1\}}\big(p(D_{ik}(t)\,|\,s(t)=[s(t)],\Theta_{is}=1)\cdot \\ \Pr(s(t)=[s(t)]\,|\,pa(s)(t))\big) \end{aligned} \tag{3}$$

In the absence of more precise information, we define:

$$\begin{aligned} \Pr(s(t)=0\,|\,pa(s)(t)=[\boldsymbol{r}]) &= \begin{cases} 1 & \exists p\in pa(s)(t):[p]=1 \\ 0 & \text{otherwise} \end{cases} \\ \Pr(s(t)=1|pa(s)(t)=[\boldsymbol{r}]) &= 1-\Pr(s(t)=0\,|\,pa(s)(t)=[\boldsymbol{r}]) \end{aligned} \tag{4}$$

The definition means that s is perturbed at time t, if any of its parents (including s itself) are perturbed.

We are now left with the definition of the local likelihoods $p(D_{ik}(t)\,|\,s(t)=[s(t)],\Theta_{is}=1)$, which can be calculated efficiently in a similiar fashion as for static NEMs. Details are described in the Supplementary Material.

2.3 A prior for network structures and time delays

In the above paragraph, we introduced the weighted adjacency matrix $\boldsymbol{\Psi}$ as a compact representation of the raw upstream network structure and time delays between perturbation events and downstream responses. Learning an upstream signaling cascade now amounts to learning matrix $\boldsymbol{\Psi}$ based on the likelihood in Equation (2). While scoring candidate network structures $\boldsymbol{\Psi}$ we assume higher edge weights (i.e. observing an effect after longer time lag) to be less likely than small edge weights. Morever, we have to ensure that redundant edges, i.e. edges which are principally not observable do not appear. In other words, addition of these edges to $\boldsymbol{\Psi}$ does not change the likelihood of our model. In general, edges $x\rightarrow y$ are redundant, if their time lag is larger or equal than the length of some other path from x to y, where the path length is defined here as the sum of the path's time lags.

We specify our demands in form of a prior $p(\boldsymbol{\Psi})$. Additionally to punishing higher edge weights, $p(\boldsymbol{\Psi})$ should enable us to include prior knowledge into the network scoring, i.e. to bias the scoring toward known interactions. For this purpose, we assume to have a given matrix $\hat{\boldsymbol{\Psi}}$ of prior probabilities for each edge. Fröhlich *et al.* (2007) proposed the following prior for $\boldsymbol{\Psi}$:

$$p(\boldsymbol{\Psi}|\nu)=\prod_{i,j}\frac{1}{2\nu}\exp\left(\frac{-|\Psi_{ij}-\hat{\Psi}_{ij}|}{\nu}\right) \tag{5}$$

where $\nu>0$ is an adjustable scaling parameter. The parameter ν can be chosen according to the BIC criterion:

$$\text{BIC}=-2\log p(\boldsymbol{D}\,|\,\boldsymbol{\Phi})+\log(|\mathcal{E}|)\sum_{i,j}\mathbf{1}\left\{|\Psi_{ij}-\hat{\Psi}_{ij}|>0\right\}$$

where $\sum_{i,j}\mathbf{1}\left\{|\Psi_{ij}-\hat{\Psi}_{ij}|>0\right\}$ is an estimate of the number of parameters in the model.

In the default case (which is employed in our subsequent studies), we suppose $\hat{\boldsymbol{\Psi}}$ to be an empty graph. This way sparse network structures are favored.

2.4 Network learning for dynoNEMs

Network learning in our case amounts to find an optimal weighted adjacency matrix $\boldsymbol{\Psi}$, where the entries Ψ_{ij} can take on discrete values $0,\ldots,T$. We employ a greedy hill climbing strategy here. For this purpose, we define three search operators: edge weight increase ($\Psi_{ij}\mapsto\Psi_{ij}+1$, if $\Psi_{ij}<T$), edge weight decrease ($\Psi_{ij}\mapsto\Psi_{ij}-1$, if $\Psi_{ij}>0$), edge reversal (exchange of Ψ_{ij} and Ψ_{ji}) . At each search step, we are applying all possible operators and accept the solution increasing the posterior likelihood most. This requires $O(|\mathcal{S}|^2)$ likelihood evaluations per search step, where each likelihood computation according to Equation (2) has a time complexity of $O(T|\mathcal{E}||\mathcal{S}|^2)$ by itself. Hence, each search step requires $O(T|\mathcal{E}||\mathcal{S}|^4)$ time.

To further assess the confidence of the inferred network hypothesis on real experimental data, we employ non-parametric bootstrapping (1000 times). That means, from the whole set $\mathcal{E}$ of available downstream effects we randomly draw bootstrap samples $\mathcal{E}'\subseteq\mathcal{E}$ of size $|\mathcal{E}|$ with replacement. On each bootstrap sample, we estimate a network hypothesis using greedy hill climbing. This allows us to estimate confidence intervals for each Ψ_{ij}.

2.5 Related approaches

2.5.1 DNEMs A first approach for learning from perturbation time series was proposed recently by Anchang *et al.* (2009). Our method differs from theirs according to the following aspects:

(1) We do not aim to infer the true upstream signaling time from downstream effects. Instead we only estimate the time lag between a perturbation and an observed downstream effect. This allows us to unroll the signal flow in the upstream signaling cascade over time.

(2) Thanks to this trick we do not have to use time consuming Gibbs sampling. Instead, we employ a simple greedy hill climbing strategy in combination with a non-parametric bootstrap to assess confidences of inferred edges. This makes our approach much more practical and efficient.

2.5.2 Static NEMs for time series Supposed that measurements of all time points were statistically independent one can in principle directly apply static NEMs to time series data. In case of statistical independency of time points, we have

$$p(D_{ik}(t)\,|\,\mathbf{\Psi}, s(t)=[s(t)], \Theta_{is}=1)=p(D_{ik}(t)\,|\,\mathbf{\Psi}, \Theta_{is}=1) \quad (6)$$

Denote

$$p(D_{ik}\,|\,\mathbf{\Psi}, \Theta_{is}=1)\equiv\prod_{t=1}^{T} p(D_{ik}(t)\,|\,\mathbf{\Psi}, \Theta_{is}=1)$$

then we retrieve back the classical likelihood formula for NEMs (see Supplementary Material). The difference of dynoNEMs to this idea is the propagation of perturbation effects over time using the time lags (Equation 3).

A principle problem when applying the static NEM approach in the context of time series data is that structures with the same transitive closure are likelihood equivalent, i.e. cannot be distinguished. For that reason, we use the maximum a-posteori (MAP) approach introduced in Tresch and Markowetz here instead (Tresch and Markowetz, 2008). In this approach, the assignment of E-genes to S-genes encoded in the parameter $\mathbf{\Theta}$ is not integrated out, but estimated explicitely. This in principle allows to distinguish direct from indirect perturbation effects, and thus breaks the likelihood equivalence. A comparision between the MAP approach and the classical NEM approach can be found in Fröhlich *et al*. (2009).

We use greedy hill climbing to learn the network structure for time series data via a static NEM. Search operations are edge insertion, edge deletion and edge reversal, and the greedy hill climber is initialized with an empty network. The same prior for network structures as described above is employed to enforce sparsity of the solution. In the following, we will use the static NEM approach as a 'ground truth' for method comparison.

3 RESULTS

3.1 Simulations

3.1.1 Network sampling To assess the reconstruction performance of our approach, we ran simulations on 10 networks with n S-genes (n was varied during our simulations, see below), which were generated in a random fashion from KEGG signaling pathways as follows: a random KEGG signaling pathway was picked as base graph. Only gene–gene interactions were considered within that graph. Within the chosen pathway, a core node was selected randomly. Starting from that core node a random walk (visiting nodes with equal probability) was started and continued until the total number of visited nodes was n. The subgraph of visited nodes was regarded as an S-gene network, and a set of m E-genes (m is varied during our simulations) was attached uniform randomly to all these S-genes. Finally, time lags for edges between connected S-genes were sampled from the set $\{1,2,3,4\}$ via a geometric distribution with parameter p. Doing so, we ensured that an indirect edge $x\rightarrow y$ had always a time lag being smaller than than the sum of time lags on all other paths from x to y, i.e. we did not get redundant edges.

We further ensured that at least one of the sampled networks was cyclic. All sampled networks can be found in the Supplementary Material.

3.1.2 Data sampling To generate data for the sampled 10 networks, we simulated knockdowns of each of the network S-genes according to Equation (4). That means, if S-gene k was 'knocked out', then a downstream S-gene s was supposed to be perturbed as soon as the perturbation signal reached s (here the time lags came into play). If s was perturbed at time t, then we sampled 'P-values' for all E-genes attached to s from the f_1 distribution (see Supplementary Material), otherwise from a uniform distribution (i.e. we supposed to have only one control available, see Supplementary Material). The parameters for the f_1 distribution were itself drawn randomly in the same manner as described in Fröhlich *et al*. (2009). Afterwards the local effect likelihoods $p(D_{ik}(t)\,|\,s(t)=[s(t)], \Theta_{is}=1)$ were estimated using again the f_1 distribution, but this time with slightly different parameters in order to simulate the estimation error of distribution parameters occuring in practice (see Supplementary Material for details).

The whole data generation process was repeated 100 times for each network and each parameter configuration being investigated in the following.

3.1.3 Dependency on time lag distribution In a first simulation, we investigated the dependency of our dynoNEM and static NEM models on the time lag distribution for networks with $n=5$ S-genes, $m=100$ E-genes and time series of length $T=10$, i.e. we varied the parameter p of the geometric distribution. The dependency was measured in terms of sensitivity and specificity (1 − false positive rate) of network reconstruction. Moreover, for dynoNEMs we recored the mean squared error (MSE) of estimated time lags. It is worth mentioning that the hill climber for the dynoNEM model was always initialized with the network estimated by a static NEM. We also tested an initialization with an empty graph, but did not find much difference to the results presented here. A direct comparison to the DEM approach by Anchang *et al*. was impossible due to the prohibitively high computation time for their method.

Since in this simulation T was (intentially) much larger than the maximum time lag of 4, the result showing an almost constant behavior of sensitivity, specificity and MSE over the whole parameter range was somehwat expected (Figures in Supplementary Material). It demonstrated that provided we have long enough time series the reconstruction quality is independent of the shape of the time lag distribution. Besides that, it is worth mentioning that our dynoNEMs achieved a sensitivity of >90% with a specifictiy of almost 100%, whereas static NEMs had much lower average sensitivity of ~80% and specificity of ~90%.

3.1.4 Dependency on length of time series In a next simulation, we looked at the influence of the lenght T of the time series for networks with $n=5$ S-genes, $m=100$ E-genes and P fixed to 0.5 (Figures 3 and 4, rest in Supplementary Material). As expected, an increase of the number of time points lead to a strong improved sensitivity for both, dynoNEMs and static NEMs. The average difference in sensitivity between both methods remained between 15 and 20%. Like before, dynoNEMs achieved a very high specificity of almost 100%, even with a low number of time points, whereas static NEMs showed a slight decrease of specificity with an increasing number of time points. The MSE of estimated time lags dropped to almost 0 with increasing T.

3.1.5 Dependency on number of S- and E-genes In a third simulation, we investigated the performance for a varying number of E-genes with $n=5, 10$ and 15 S-genes and time series of length $T=10$. Time lags were sampled from a geometric distribution with $P=0.5$ (Figures 5 and 6. The dependence on the choice of p, i.e. the influence of the signaling times prior has been evaluated in Section 3 of the Supplementary Material). In all cases, the dynoNEM

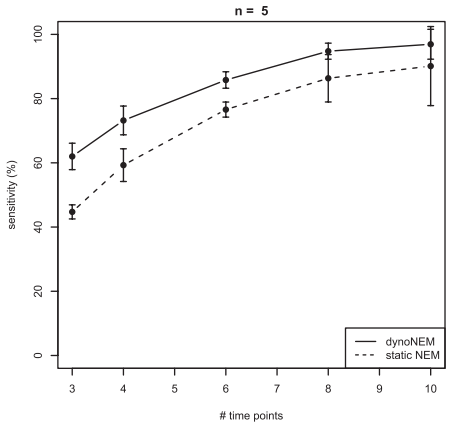

Fig. 3. Dependency of sensitivity of network reconstruction on length of time series (parameter T). Error bars indicate the SD of the mean.

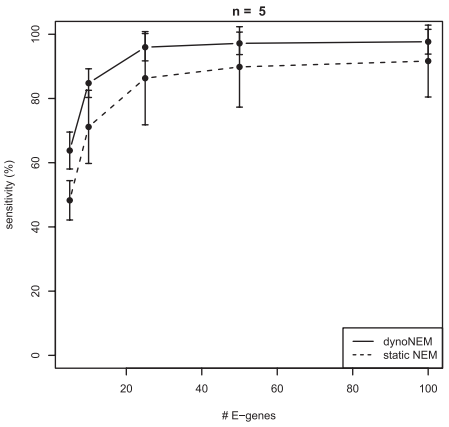

Fig. 5. Dependency of mean sensitivity of network reconstruction on number of E-genes ($n=5$ S-genes, $T=10$).

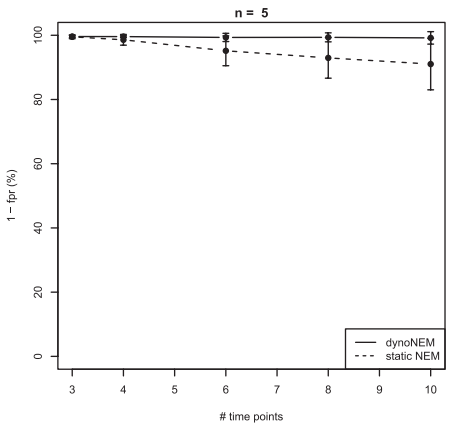

Fig. 4. Dependency of specificity of network reconstruction on length of time series (parameter T). Error bars indicate the SD of the mean.

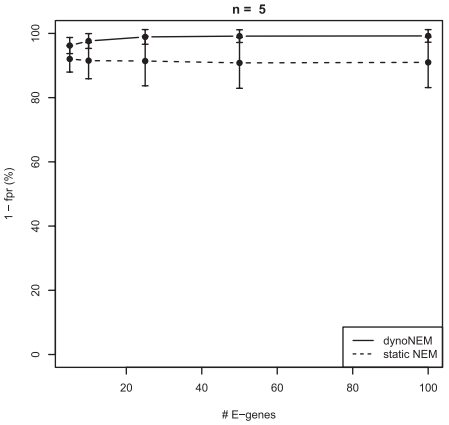

Fig. 6. Dependency of mean specificity of network reconstruction on number of E-genes ($n=5$ S-genes, $T=10$).

model led to a consistently better sensitivity and specificity than the static NEM model, with improvements in sensitivity and specifictiy of ~15–20%. At the same time, the simulation demonstrated that already with a rather low number of E-genes, dynoNEMs reach a rather good reconstruction accuracy. For instance, for $n=5$ it seems that just 25 E-genes are sufficient to get a sensitivity of >90%. Like before, the specificity was constantly >95%. With increasing network size, i.e. larger n, the sensitivity drops slightly to 80% ($n=$ 10, 100 E-genes) and 70% ($n=15$, 300 E-genes), respectively, but the specificity remains constantly >95% (Figures in Supplementary Material).

3.1.6 Dependency on network architecture We finally investigated, whether there were systematic differences in the reconstruction accuracy for each of our sampled networks. We compared the sensitivities and specificities for each network with $n=5$ S-genes, $m=100$ E-genes, time lags sampled with $P=0.5$ and varying parameter T (see Supplementary Material). This simulation showed for longer time series an influence of the network architecture, whereas for smaller ones the length of the time series was the dominating factor. Specifically, the cyclic network #10 appeared to be harder to learn in terms of sensitivity and specificity than the other ones, and also the indirect edge in network #2 caused sometimes difficulties. Nonetheless, a detailed investigation of the frequency with which individual edges were inferred clearly revealed that dynoNEMs were in general able to identify feed-forward and feed-back loops correctly with high accuracy, even from short time series (see Supplementary Material).

3.2 Application to murine stem cell development

We applied dynoNEMs to a dataset investigating molecular mechanisms of self-renewal in murine embryonic stem cells (Ivanova *et al.*, 2006). The dataset consists of time series microarray measurements for RNAi knockdowns of each of the six genes namely, *Nanog*, *Oct4*, *Sox2*, *Esrrb*, *Tbx3* and *Tcl1*. Mouse embryonic stem cells (ESCs) were grown in the presence of leukemia inhibitory factor (LIF), thus retaining their undifferentiated self-renewing state (positive controls). Differentiation-associated changes in gene expression were measured by replacing LIF with retinoic acid (RA), thus inducing differentiation of stem cells (negative controls). RNAi silencing of the six afore mentioned genes was done in (LIF+, RA−) cell culturs to investigate their potential for induced cell

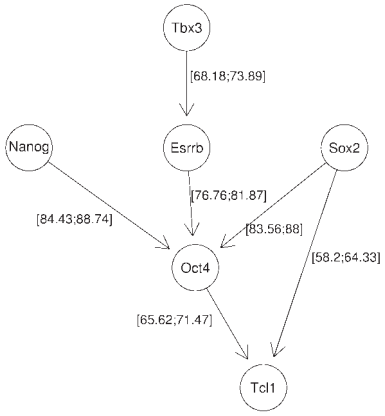

Fig. 7. Inferred network for murine stem cell development; 95% confidence intervals (%) are shown at each edge.

differentiation. Microarray expression measurements at 6–7 time points in 1-day intervals were taken for the two controls (positive and negative) and the six RNAi assays.

The dataset by Ivanova *et al*. (2006) was measured once on Affymetrix MOE430A and once on MOE430B chips. The authors normalized both datasets via the MAS 5.0 software (Hubbell *et al*., 2002). In consistency with Anchang *et al*. (2009) for the following analysis, we only concentrated on the Affymetrix MOE430A chip series. In our framework, the six genes *Nanog*, *Oct4*, *Sox2*, *Esrrb*, *Tbx3* and *Tcl1* correspond to S-genes, and all genes showing a significant fold change in response to LIF depletion are E-genes. We here used the same set of 122 E-genes and the same data discretization procedure as described in Anchang *et al*. (2009).

We applied our dynoNEM model within a non-parametric bootstrap procedure and recorded, how often each edge appeared in 1000 inferred networks (one network per bootstrap sample). We then computed exact binomial distribution confidence intervals (95%) for the appearence probability of each edge via R-package *binom* (Dorai-Raj, 2009). The validity of bootstrap probabilities for edges was further checked in a simulation (see Supplementary Material). Only edges with lower confidence bound >50% were regarded as reliable and shown in Figure 7. The median time lags for all edges were 1, i.e. had the same speed.

There are several similarities to the inferred network shown in Anchang *et al*. (2009), which was obtained via the DNEM method, namely the cascades $Tbx3 \rightarrow Esrrb \rightarrow Oct4 \rightarrow Tcl1$, $Nanog \rightarrow Oct4 \rightarrow Tcl1$ and $Sox2 \rightarrow Oct4 \rightarrow Tcl1$. A further striking similarity is that the transcription factor *Oct*4 regulating *Tcl*1 is itself jointly regulated by the three transcription factors *Nanong*, *Sox*2 and *Esrrb*. In contrast to Anchang *et al*., in our network *Nanog* is not placed upstream of *Sox*2 and does not have any indirect outgoing edges. Indeed, the only shortcut in our network is $Sox2 \rightarrow Tcl1$. Our network is thus more sparse than the one shown by Anchang *et al*. This can be partially attributed to our structure prior (Section 2.3), which favors sparse networks. A comparison of the pure graph structures (neglecting edge probabilities) of both, our network and the network by Anchang *et al*., via our likelihood model favored our network with a Bayes factor of 2.14×10^{18}. Thus, we believe that the reason for the (relatively small) differences between our network and Anchang *et al*. are a mixture of a different likelihood model combined with a sparsity prior.

A further comparison of our dynoNEM network with a reconstruction based on static NEMs can be found in the Supplementary Material.

For validation purposes, we next performed a literature-based reconstruction of the six gene network via the MetaCore^{TM} software (GeneGo Inc.) showing eight known transcriptional and three known non-transcriptional regulations (see Supplementary Material). The sub-networks $Sox2 \rightarrow Oct4 \rightarrow Tcl1$ and $Esrrb \rightarrow Oct4 \rightarrow Tcl1$ appearing in our data-based reconstruction can be found in this literature network as well as *Nanog* and *Tbx3* being placed upstream of *Oct4*. The edges $Sox2 \rightarrow Tcl1$ and $Nanog \rightarrow Oct4$, which both appear in our network and the one shown in Anchang *et al*. (2009) can be explained by the fact that *Sox2* binds to *Oct4*, which itself binds to *Essrb*, to which also *Nanog* binds (Bilodeau *et al*., 2009). Hence, $Sox2 \rightarrow Tcl1$ and $Nanog \rightarrow Oct4$ are most likely mediated through *Oct*4 and *Essrb*, respectively. Likewise, $Tbx3 \rightarrow Esrrb$ can be interpreted as being mediated through *Oct*4. A different way of interpretation was given by Anchang *et al*., who noted that feed-forward loops, such as $Sox2 \rightarrow Tcl1$, $Nanog \rightarrow Oct4$ and $Tbx \rightarrow Esrrb$, are very frequent motifs in transcriptional networks and could serve to stabilize the differentiated state of cells relative to the self-renewal state, hence making cell differentiation a mainly unidirectional process (c.f. Alon, 2006).

Summing up, we mainly see known transcriptional regulations, and our network is consistent with the known literature.

4 CONCLUSION

We developed a novel approach to infer signaling cascades from high-dimensional perturbation time series measurements via dynoNEMs, hence allowing to discriminate direct from indirect perturbation effects and to resolve feedback loops. Our approach directly extends the NEMs framework introduced by Markowetz *et al*. from the static to the dynamic case by unrolling the network structure over time. This idea allows for a fast and efficient computation of the likelihood function without any time consuming Gibbs sampling. Our simulations revealed a high accuracy of network reconstruction for sufficiently small networks (up to 15 nodes). Thanks to the incorporated network structure prior, we achieved a consistantly high specificity, i.e. low false positive rate. Applying our dynoNEMs to estimate the network between six proteins (five transcription factors) playing a key role in murine stem cell development, we found a good agreement with results published by Achang *et al*. and with the biological literature. We therefore believe that our approach can serve as a useful tool to generate data-driven hypotheses about signaling and/or transcriptional networks based on high-dimensional perturbation effects.

Funding: P.P. is supported by the state of NRW through a B-IT research school grant.

Conflict of Interest: none declared.

REFERENCES

Aho,A. *et al*. (1972) The transitive reduction of a directed graph. *SIAM J. Comput.*, **1**, 131–137.

Alon,U. (2006) *Introduction into Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall/CRC, Boca Raton, Florida, USA.

Anchang,B. *et al*. (2009) Modeling the temporal interplay of molecular signaling and gene expression by using dynamic nested effects models. *Proc. Natl Acad. Sci. USA*, **106**, 6447–6452.

Battle,A. *et al*. (2010) Automated identification of pathways from quantitative genetic interaction data. *Mol. Syst. Biol.*, **6**, 1–13.

Bilodeau,S. *et al*. (2009) Setdb1 contributes to repression of genes encoding developmental regulators and maintenance of es cell state esrrb activates oct4 transcription and sustains self-renewal and pluripotency in embryonic stem cells. *Genes Dev.*, **23**, 2484–2482.

Dorai-Raj,S. (2009) *binom: Binomial Confidence Intervals For Several Parameterizations*. R package version 1.0-5.

Driessche,N.V. *et al*. (2005) Epistasis analysis with global transcriptional phenotypes. *Nat. Genet.*, **37**, 471–477.

Fire,A. *et al*. (1998) Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans. *Nature*, **391**, 806–811.

Fröhlich,H. *et al*. (2007) Large scale statistical inference of signaling pathways from RNAi and microarray data. *BMC Bioinformatics*, **8**, 1–15.

Fröhlich,H. *et al*. (2008) Estimating large scale signaling networks through nested effect models with intervention effects from microarray data. *Bioinformatics*, **24**, 2650–2656.

Fröhlich,H. *et al*. (2009) Nested effects models for learning signaling networks from perturbation data. *Biometrical J.*, **2**, 304–323.

Gat-Viks,I. *et al*. (2006) A probabilistic methodology for integrating knowledge and experiments on biological networks. *J. Comput. Biol.*, **13**, 165–181.

Ghahramani,Z. (1997) Learning dynamic Bayesian networks. *Lect. Notes Comput. Sci.*, **1387**, 168–197.

Hubbell,E. *et al*. (2002) Robust estimators for expression analysis. *Bioinformatics*, **18**, 1585–1592.

Ivanova,N. *et al*. (2006) Dissecting self-renewal in stem cells with RNA interference. *Nature*, **442**, 533–538.

Kanabar,P.N. *et al*. (2009) Inferring disease-related pathways using a probabilistic epistasis model. *Pac. Symp. Biocomput.*, **491**, 480–491.

Klamt,S. *et al*. (2010) TRANSWESD: inferring cellular networks with transitive reduction. *Bioinformatics*, **26**, 2160–2168.

Maathuis,M.H. *et al*. (2010) Predicting causal effects in large-scale systems from observational data. *Nat. Methods*, **7**, 247–248.

Maathuis,M. *et al*. (2009) Estimating high-dimensional intervention effects from observational data. *Ann. Stat.*, **37**, 3133–3164.

Markowetz,F. *et al*. (2005) Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics*, **21**, 4026–4032.

Markowetz,F. *et al*. (2007) Nested effects models for high-dimensional phenotyping screens. *Bioinformatics*, **23**, i305–i312.

Nelander,S. *et al*. (2008) Models from experiments: combinatorial drug perturbations of cancer cells. *Mol. Syst. Biol.*, **4**, 216.

Pearl,J. (2000) *Causality: Models, Reasoning and Inference*. Cambridge University Press, Cambridge.

Pe'er,D. *et al*. (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17** (Suppl. 1), S215–S224.

Rogers,S. and Girolami,M. (2005) A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, **21**, 3131–3137.

Rung,J. *et al*. (2002) Building and analysing genome-wide gene disruption networks. *Bioinformatics*, **18** (Suppl. 2), S202–S210.

Sachs,K. *et al*. (2005) Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, **308**, 523–529.

Tresch,A. *et al*. (2007) Discrimination of direct and indirect interactions in a network of regulatory effects. *J. Comput. Biol.*, **14**, 1217–1228.

Tresch,A. and Markowetz,F. (2008) Structure learning in nested effects models. *Stat. Appl. Genet. Mol. Biol.*, **7**.

Vaske,C.J. *et al*. (2009) A factor graph nested effects model to identify networks from genetic perturbations. *PLoS Comput. Biol.*, **5**, e1000274.

Wagner,A. (2001) How to reconstruct a large genetic network from n gene perturbations in fewer than n 2 easy steps. *Bioinformatics*, **17**, 1183–1197.

Zeller,C. *et al*. (2009) A bayesian network view on nested effects models. *EURASIP J. Bioinformatics Syst. Biol.*, **2009**, 8.