

Learning gene network structure from time laps cell imaging in RNAi Knock downs

Henrik Failmezger^{1,†}, Paurush Praveen^{2,†}, Achim Tresch¹ and Holger Fröhlich^{2,*}

¹Computational Biology and Regulatory Networks, Max-Planck Institute for Plant Breeding Research, Carl-von-Linne-Weg 10, 50829 Cologne, Germany and ²Algorithmic Bioinformatics, Bonn-Aachen International Center for IT, Rheinische Friedrich-Wilhelms-Universität Bonn, Dahlmannstr. 2, 53113 Bonn, Germany

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: As RNA interference is becoming a standard method for targeted gene perturbation, computational approaches to reverse engineer parts of biological networks based on measurable effects of RNAi become increasingly relevant. The vast majority of these methods use gene expression data, but little attention has been paid so far to other data types.

Results: Here we present a method, which can infer gene networks from high-dimensional phenotypic perturbation effects on single cells recorded by time-lapse microscopy. We use data from the Mitocheck project to extract multiple shape, intensity and texture features at each frame. Features from different cells and movies are then aligned along the cell cycle time. Subsequently we use Dynamic Nested Effects Models (dynoNEMs) to estimate parts of the network structure between perturbed genes via a Markov Chain Monte Carlo approach. Our simulation results indicate a high reconstruction quality of this method. A reconstruction based on 22 gene knock downs yielded a network, where all edges could be explained via the biological literature.

Availability: The implementation of dynoNEMs is part of the Bioconductor R-package *nem*.

Contact: frohlich@bit.uni-bonn.de

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

Received on December 7, 2012; revised on April 8, 2013; accepted on April 14, 2013

1 INTRODUCTION

The availability of large RNAi screens (Fire *et al.*, 1998) has raised the interest in computational approaches for reverse engineering parts of biological networks from measurable effects of targeted gene perturbations. Examples thereof range from graph theoretical algorithms (Klamt *et al.*, 2010; Rung *et al.*, 2002; Tresch *et al.*, 2007; Wagner, 2001) over Bayesian Networks (Maathuis *et al.*, 2009, 2010; Pe'er *et al.*, 2001; Sachs *et al.*, 2005) to factor graphs (Gat-Viks *et al.*, 2006), epistasis analysis (Driessche *et al.*, 2005; Kanabar *et al.*, 2009) and ordinary differential equation systems (Nelander *et al.*, 2008). Most of these methods make use of gene expression data. The few attempts to

reverse engineer gene networks from phenotypic data include studies by Bakal *et al.* (2007), who rely on hierarchical clustering of static images, and Kaderali *et al.* (2009), who use a probabilistic graphical model for only one binary phenotypic variable in static images. To our knowledge, there is yet no method for the inference of networks from time-lapse microscopy based on large numbers of statistical image features.

Nested Effects Models (NEMs) are a class of probabilistic graphical models that have been introduced originally by Markowetz *et al.* (2005) and extended substantially later on by Anchang *et al.* (2009); Fröhlich *et al.* (2007, 2008, 2011); Markowetz *et al.* (2007); Niederberger *et al.* (2012); Tresch and Markowetz (2008); Vaske *et al.* (2009); Zeller *et al.* (2009). In NEMs, indirect high-dimensional downstream effects of multiple single-gene knock downs are studied. NEMs allow for inferring the signaling flow between these perturbed genes on a transcriptional as well as non-transcriptional level based on the measured intervention effects. Anchang *et al.* (2009) and Fröhlich *et al.* (2011) extended the theory of NEMs to time-series data, and applied it to infer parts of a transcriptional network involved in murine stem cell development. Originally, NEMs assumed downstream effects to be measured via gene expression profiling, but here we use phenotypic image features from movies instead. Our movies were taken from the Mitocheck database (Neumann *et al.*, 2010), in which about 20 000 human genes were silenced via RNAi and subsequently screened for cell cycle defects. We use Dynamic NEMs (Fröhlich *et al.*, 2011) to estimate the network between perturbed genes based on the dynamic response of the phenotype along the cell cycle. The inference is based on a Markov Chain Monte Carlo (MCMC) algorithm, which is introduced in this article.

2 METHODS

2.1 Movie preprocessing and feature extraction

The time-lapse movies were downloaded from the Mitocheck database (Neumann *et al.*, 2010). In this database, about 20 000 human genes were knocked down and screened for cell cycle defects via time-lapse microscopy. The nuclei of HeLa cells were fluorescently labeled by tagging the core histone 2B with GFP. Images of these cells were taken in 30 min intervals for 48 h. Movie processing was based on the following main steps: (i) cell nuclei detection in the individual frames of the movies; (ii) tracking of the nuclei over time; (iii) calculation of morphological features; and (iv) calculation of cell cycle time. Please note that in each movie, many cells were affected by a common gene perturbation. These

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

cells were tracked over time, and for every cell and every time point, a feature vector was calculated.

The OpenSource software CellProfiler (Kamentsky *et al.*, 2011) was used for all of these steps. Cells nuclei were detected by Otsu thresholding (Otsu, 1979), followed by the watershed algorithm to separate clustered nuclei (Malpica *et al.*, 1997). As we realized that the watershed algorithm often oversegmented nuclei, we implemented a segmentation correction scheme in CellProfiler based on Chen *et al.* (2006). Standard CellProfiler features were used for morphological description of the nuclei. Thirteen shape features were calculated including area, perimeter and eccentricity of the cell nuclei. Sixteen features characterized the pixel intensities for the nuclei, e.g. mean intensity and standard deviation of intensities. The 14 texture features were calculated from the co-occurrence matrix of pixel intensities. Furthermore, 12 radial distribution features were added. These features measure the pixel intensities from the nucleus center to the boundaries using four bins. Finally, 30 Zernike moments were included into the feature pool (Boland *et al.*, 1998).

As the cells did not move fast between frames, the standard CellProfiler nearest neighbor tracker delivered good results. For each cell, the time point of cell division was taken from the tracking procedure. Later analysis steps require the conversion of the absolute time at which a cell was recorded into a relative time (called cell cycle time) within an idealized cell cycle. For every movie, we first estimated the mean cell cycle time T as the quotient of the length of all trajectories (in minutes), divided by the total number of division events that have been observed in the movie. For cells that were observed at time t between two division events at times $t_1 < t_2$, we define the (relative) cell cycle time r as the quotient $r = (t - t_1)/(t_2 - t_1)$. For cells that were observed before the first division event of their trajectory, the cell cycle time is defined as $r = 1 - t/\max(t_1, T)$ with t_1 being the time of the first division event. Analogously, if cells in a trajectory of length t_ω were observed after the last cell division, we define $r = (t - t_n)/\max(t_\omega - t_n, T)$ with t_n being the time of the last division event. Cells that never divided during the observation period were assigned the relative cell cycle time $r = t/t_\omega$.

2.2 Estimation of perturbation effects

For this paragraph, consider one fixed image feature. We need to quantify the extent to which a perturbation confers a phenotype (as measured by that feature) that differs in the observed cell population compared with the wild-type cell population. Such a comparison was done in awareness of the physiological phenotypic changes occurring in wild-type cells during the cell cycle. We therefore split the cell cycle time into 10 equally sized intervals and computed for each trajectory and each feature the median value per interval. Please recall at this point that there is one trajectory per individual cell. We typically obtained more than 50 median feature values (corresponding to 50 cells) per interval from one movie. Manual inspection of the data revealed that we could not assume feature values to follow any known parametric distribution. Thus treated and control conditions were compared via a Mann–Whitney U -test.

We then fitted the distribution of P -values obtained for all 85 features at a given time step via a Beta-uniform mixture (BUM) model, as described in Fröhlich *et al.* (2008). The idea is that the distribution of P -values can be decomposed into a uniform part (the null distribution) and a second part (the alternative distribution), which itself can be modeled via two Beta distributions (Fig. 1). It has to be mentioned here that at each time point, the number of compared cells is different. Therefore, the null distribution of P -values (the uniform part of the BUM model) also differs between time points. However, we are here only interested in the alternative distribution (describing the likelihood of an effect).

After having learned the BUM model, each P -value can be converted into a probability of being generated by the null respectively the alternative distribution. Thus, for each perturbation, feature and interval, we calculate the probability of being phenotypically different from the wild-type.

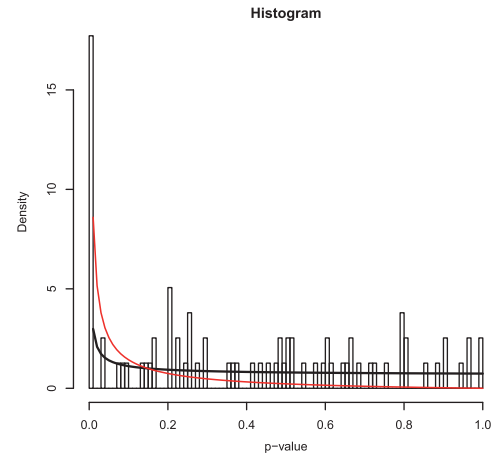


Fig. 1. Empirically observed histogram of P -values for 85 image features in one of our movies (CDK2 knock down at time step 10). Black = fitted BUM model; red = estimated alternative distribution indicating the probability of being phenotypically different from the wild-type

2.3 Dynamic NEMs

NEMs distinguish between silenced/perturbed entities (genes or proteins, which cannot be observed directly), called S-genes (\mathcal{S}), and other entities (here, phenotypic image features) describing measurable downstream effects (\mathcal{E}). The idea of NEMs is to separate the upstream signaling pathway (the graph connecting the S-genes, Φ) from the downstream effects (the graph connecting S-genes to effect features, Θ). Consequently, the edge set of an NEM is partitioned into a directed edge set connecting S-genes (Φ), and another one (Θ) describing the connections between the S-genes and effect features. It is assumed that each $e \in \mathcal{E}$ is attached to at most one $s \in \mathcal{S}$ only. Knocking down a specific S-gene $k \in \mathcal{S}$ interrupts the signal flow in the downstream pathway, and hence an effect on feature e attached to k or one of the S-genes depending on 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.

Dynamic NEMs (dynoNEMs, Fröhlich *et al.*, 2011) extend this idea further by assuming that each interaction between S-genes requires a certain amount of time, which yields to a time-delayed response of downstream effects (Fig. 2a). A positive time lag between nodes k and s describes the number of time steps, after which a knockdown of node k results in an *observed* effect for features attached to node s . We do not make any assumptions about the physical time it takes a signal at node k to produce an effect for feature e .

Formally a dynoNEM model is represented by the adjacency matrix Ψ of an edge-weighted graph between S-genes ($\Psi_{sk} \in \{0, \dots, T\}$ for all s, k) and the adjacency matrix Θ of the feature-attachment graph. That means $\Theta_{si} = 1$, if feature i ($i = 1, \dots, \ell$) is attached to S-gene s , and 0 otherwise. Furthermore, by convention, let $\Psi_{kk} = 1$. Let $D = \{D_{ij}(t) | t = 1, \dots, T, i = 1, \dots, \ell, j = 1, \dots, n\}$ denote the complete array of P -values for T time points, $\ell = 85$ features and n perturbations. The likelihood of a dynoNEM model can be computed in closed form (Fröhlich *et al.*, 2011):

$$p(D|\Psi) = \prod_{i \in \mathcal{E}} \sum_{s \in \mathcal{S}} \prod_{k \in \mathcal{S}} \prod_{t=1}^T p(D_{ik}(t) | \Psi, \Theta_{si} = 1) \Pr(\Theta_{si} = 1) \quad (1)$$

The attachment positions Θ are integrated out because they are treated as nuisance parameters. The prior $\Pr(\Theta_{si} = 1)$ is taken to be uniform over all S-genes in the absence of further information. To compute $p(D_{ij}(t) | \Psi, \Theta_{si} = 1)$, we can unroll Ψ over time, similar to a Dynamic

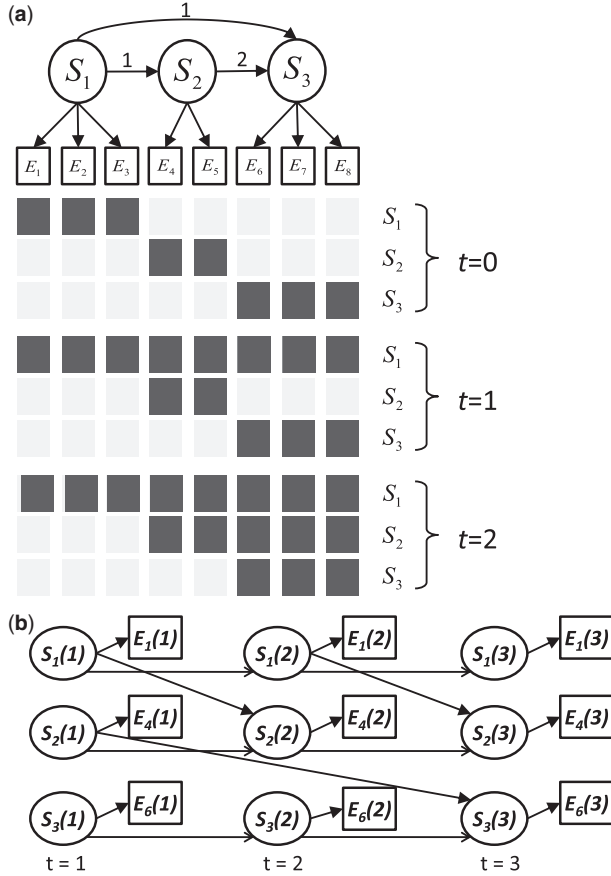


Fig. 2. (a) A dynamic NEM is parameterized by a directed edge-weighted graph between S-genes encoded (Ψ) together with a directed graph (Θ) attaching each feature to an S-gene. According to the graph structure, perturbation effects on observable features (dark) are predicted in dependency on the time point of observation. A perturbation of S-gene S_1 affects S_2 and S_3 already after one time step, but a knock down of S_2 yields an effect on S_3 after two time steps. Hence, an effect on the features attached to S_3 (E_6, E_7, E_8) is predicted after one time step, if S_1 and after two time steps, if S_2 is silenced. (b) Unrolling of the signal flow network along time (t =time index). For the sake of better visualization, only one feature per S-gene is shown

Bayesian Network (Fig. 2b). More precisely, $p(D_{ij}(t)|\Psi, \Theta_{si} = 1)$ can be decomposed as

$$p(D_{ik}(t)|\Psi, \Theta_{si} = 1) = \sum_{s(t) \in \{0, 1\}} p(D_{ik}(t)|S(t) = s(t), \Theta_{si} = 1) \Pr(S(t) = s(t)|pa(s)(t)) \quad (2)$$

where $S(t)$ denotes a random variable for the unobservable perturbation state of S-gene s at time t . Its value is given by $s(t)$. Let $pa_s(t)$ denote the set of parent nodes of s at time t . Then, according to the unrolling of the signal flow over time, we write:

$$\Pr(S(t) = 0|pa_s(t) = R) = \begin{cases} 1 & \exists r \in R : r = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The definition means that s is perturbed at time t if any of its parents (including s itself) are perturbed. Lastly, we define the local likelihoods

$p(D_{ik}(t)|S(t) = s(t), \Theta_{si} = 1)$. Using the BUM model, this quantity can be decomposed as:

$$p(D_{ik}(t)|\Theta_{si} = 1, S(t) = s(t)) = \begin{cases} f_1(D_{ik}(t)) & s(t) = 1 \\ 1 & s(t) = 0 \end{cases} \quad (4)$$

The density function f_1 quantifies the impact on feature i under the alternative hypothesis of expecting an effect and can be extracted from the BUM model. We refer the reader to Fröhlich *et al.* (2008) for details.

To account for possibly irrelevant features, we used a trick described earlier in Tresch and Markowetz (2008): a dummy S-gene ‘null’ was added to the signaling graph Ψ , which was always unconnected to all other S-genes, i.e. not predicting any downstream effects. Features assigned to ‘null’ are hence irrelevant and do not show significant effects in any of the perturbation experiments.

2.4 Markov Chain Monte Carlo

In this article, we propose to learn the network structure (Ψ) via a Metropolis–Hastings algorithm. For this purpose, we define three possible MCMC moves within the space of edge-weighted graphs:

- (1) edge weight increase
- (2) edge weight decrease
- (3) swap of the weight between edges $i \rightarrow j$ and $j \rightarrow i$.

The first move is only allowed if the edge weight is smaller than $T - 1$, the third only if $\Psi_{ij} \neq \Psi_{ji}$. Each of the allowed steps is chosen with equal probability.

To enforce sparse network structures and to prevent unlikely large time lags, we adopt the structure prior proposed in Fröhlich *et al.* (2007).

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

which is essentially a product of independent Laplacian distributions with means $\hat{\Psi}_{ij}$. Please note that setting $\hat{\Psi}_{ij} = 0$ corresponds to enforcing sparsity, while any other choice allows for incorporating prior knowledge (Fröhlich *et al.*, 2009, 2011). The scale parameter v trades the degree of sparsity against the likelihood of the model. A large v reduces the influence of the prior, whereas a smaller v increases it. In a Bayesian manner we define an exponential distribution prior over v itself,

$$v \sim \text{Exp}(\theta) \quad (6)$$

(here: $\theta = 1$) and include v into the sampling process of the MCMC. More precisely, $\log v$ is modified every 100th iteration via a Gaussian transition kernel.

In summary, an MCMC move is accepted with probability

$$a = \min\left(1, \frac{p(D|\Psi^{\text{new}})p(\Psi^{\text{new}}|v^{\text{new}})p(v^{\text{new}})N(\Psi^{\text{old}})}{p(D|\Psi^{\text{old}})p(\Psi^{\text{old}}|v^{\text{old}})p(v^{\text{old}})N(\Psi^{\text{new}})}\right) \quad (7)$$

where $N(\cdot)$ denotes the number of networks reachable within one MCMC move from the current graph.

To speed up convergence, we initialized each MCMC run with the solution of the greedy hill climber described in (Fröhlich *et al.*, 2011).

Our whole approach, which consists of image feature extraction, estimation of perturbation effects and network estimation via dynoNEMs is called MovieNEM in the following (Fig. 3).

3 RESULTS

3.1 Simulations

3.1.1 Network sampling To assess the reconstruction performance of our MovieNEM approach, we ran simulations on 10

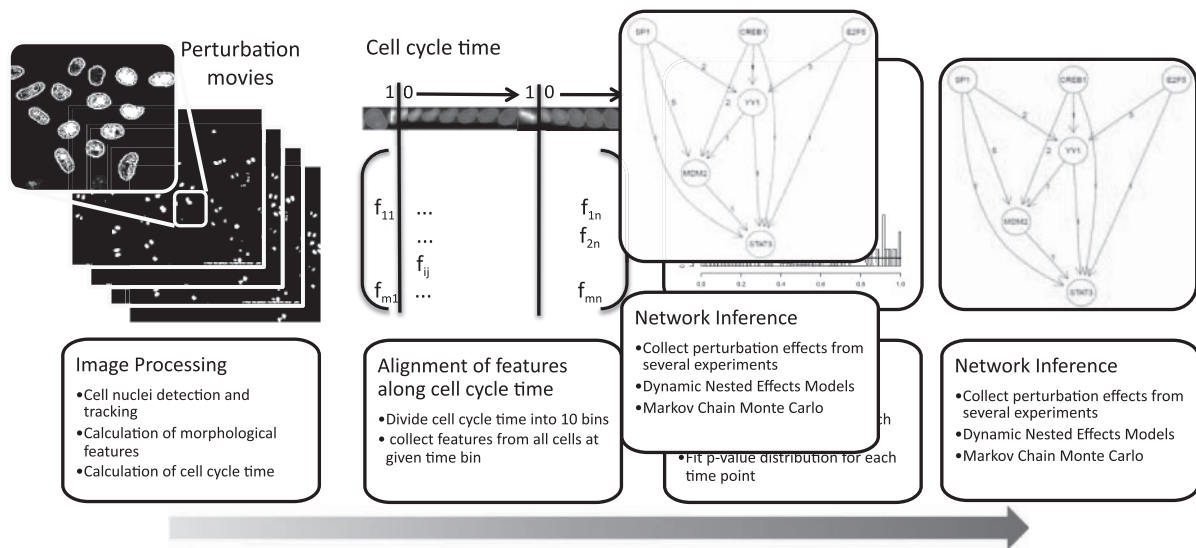


Fig. 3. Overview about MovieNEM: individual movies are first fed into an image processing pipeline consisting of four steps: (i) cell nuclei detection in the individual movie frames; (ii) tracking of the nuclei over time; (iii) calculation of morphological features; and (iv) calculation of cell cycle time. After image processing features are grouped according to the binned cell cycle time. This allows for estimating time-wise perturbation effects. Several movies, each showing one perturbation, are processed in this way and the perturbation likelihoods collected along the binned cell cycle time axis. This allows for applying Dynamic NEMs to infer the network between perturbed genes via MCMC

networks with $n \in \{5, 10, 15\}$ S-genes, 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 nodes were considered within that graph, of which 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 85 features were attached randomly to the S-genes (with uniform probability). Finally, time lags for edges between connected S-genes were sampled from the set $\{1, 2, 3, 4\}$ via a geometric distribution $P(X = k) = (1 - p)^k p$ with parameter p . We further ensured that at least one of the sampled networks was cyclic. All sampled networks can be found in the Supplementary Materials.

3.1.2 Data sampling To generate phenotypic data for the sampled 10 networks, we simulated knock downs of each of the network S-genes according to Equation (3). 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 step t , then we sampled P -values for all features attached to s from the alternative part of the BUM model, otherwise from a uniform distribution. The alternative distribution f_i was assumed to be itself a mixture of a $Beta(1, \beta)$ and a $Beta(\alpha, 1)$ distribution, and parameter α was sampled uniform randomly from $[0.1, 0.2, \dots, 0.9]$, whereas parameter β was drawn uniformly from $[5, 6, \dots, 15]$. The coefficient for the mixture of both Beta distributions was sampled uniformly from $[0.01, 0.02, \dots, 0.49]$. The length of simulated perturbation time series was in agreement with our experimental data $T=10$ time steps (see above). It should be emphasized that the sampling

process was done independently for each time step and perturbation, hence simulating different distribution characteristics (specifically different null distributions). After P -value sampling, BUM models were fitted to each time point and perturbation, allowing for computing local effect likelihoods $p(D_{ik}(t)|S(t) = s(t), \Theta_{si} = 1)$. The whole data generation process was repeated five times for each network.

For each network and sampled data set, a dynoNEM model was inferred using MCMC. For this purpose, 50 000 burn-in and 50 000 sampling iterations were executed. A thinning of 100 iterations was done to obtain an estimate of the edge-wise posterior expectation network. Edges with mean time lag not being at least 2 standard deviations above 0 were filtered out. In other words, edges with large variance were not considered.

3.1.3 Dependency on network size Initially the time-lag distribution parameter p was fixed to 0.5 and networks of size $n = 5, n = 10$ and $n = 15$ were simulated. Supplementary Figures S1–S6 revealed a high reconstruction accuracy in terms of sensitivity and specificity as well as a low mean squared error (MSE) of estimated time lags. For $n = 5$, we observed a median sensitivity of 1 together with a median specificity of 0.98 and a MSE of 0.12. For $n = 10$, the median sensitivity and specificity was about 0.9 with a MSE of about 0.9, and for $n = 15$, we still observed a median sensitivity of 0.83 with a specificity of about 0.85 and a MSE of about 1.4. Notably, we observed a certain dependency of the reconstruction performance on the network structure. For example, for $n = 10$ network #10, a dense network with lots of feedback loops, appeared to be hardest to infer, whereas network #3, a sparse acyclic graph, was estimated with high quality.

3.1.4 Dependency on time-lag distribution Next we fixed the network size to $n = 10$ and varied the parameter p to investigate the dependency on the time-lag distribution. The smaller the p the more the time-lag distribution is uniform. In comparison with our previous results with $p = 0.5$, we observed for a smaller $p = 0.2$ (distribution is more uniform) on average a small decrease with respect to the sensitivity from approximately 0.9 to 0.8, while at the same time the specificity rose from approximately 0.9 to 0.94 (Supplementary Fig. S7) and the MSE increased from approximately 0.8 to 1.3 (Supplementary Fig. S8). In contrast, an increase of p to 0.8 lead to a drop of the MSE to 0.45. The specificity remained high at approximately 0.9, while the median sensitivity was 1 (Supplementary Figs S9 and S10).

3.1.5 Dependency on uninformative features In a final simulation, we looked in how far our results were dependent on possible unspecific/noise image features. For that purpose we fixed $n = 10, p = 0.5$ and removed a varying number (5, 10, 40) of informative features from our dynoNEM model. Subsequently we added the same number (5, 10, 40) of unspecific features, for which p -values were drawn from a uniform $[0, 1]$ distribution. These noise features had thus no relation to the network to be estimated. As can be observed, our method is highly robust against noise features with respect to reconstruction of the network topology (see Supplementary Materials). Even with 40 uninformative features the average sensitivity and specificity was almost identical to the situation with 85 informative features, which underlines the effectiveness of our feature selection approach described above.

3.2 Application to movie data

3.2.1 Example: subnetwork of six genes We first exemplify the results produced by MovieNEM on a subnetwork of six genes, which we picked from a larger set of 22 genes revealing significant phenotypic perturbation effects in our movies. According to a literature-based network reconstruction, these 22 genes are relatively close to each other and are mainly involved into cell cycle, transcriptional regulation and cell differentiation. The purpose of the example is to visualize the results produced by MovieNEM, which would be hard for the whole network of 22 genes. The number of MCMC iterations in the burn-in and sample phase was both set to 100 000.

Figure 4 visualizes the edge-wise posterior expectation network together with the observed and predicted phenotypic effects and corresponding typical cell morphologies at different time points. Already at time point 1 CREB1 knock down leads to certain perturbation effects. STAT3 knock down affects fewer features than the CREB1 perturbation. Accordingly, STAT3 is placed downstream of CREB1 in the network graph. Similar observations can be made, e.g. for CREB1 and YY1, E2F5 and STAT3, YY1 and STAT3 and MDM2 and STAT3. Generally the number of phenotypic effects increases over time. At time point 2, we can see a noisy subset relation between CREB1 and MDM2. At time point 4, yet more effects and corresponding subset relations are observable, e.g. between E2F5 and YY1 as well as SP1 and MDM2.

The estimated dynoNEM model allows for making predictions of intervention effects on feature sets: according to the model features assigned to STAT3, which is placed most downstream in the network, should exhibit significant perturbation effects at all time points, and indeed this is visible in the heatmaps in Figure 4. In contrast, MDM2 features at time point 1 should react only to MDM2 and YY1 knock down, which is again in agreement with the data. However, at time point 2, MDM2-assigned features are predicted to show effects also under CREB1 and at time point 4 also under SP1 and E2F5 perturbation. Once again, this can be confirmed in our data. Similar observations can be made for other feature sets.

We conducted a permutation test to check whether our inferred network could have been expected by chance. For this purpose, we permuted the node labels in each network 1000 times and calculated for each permuted network its likelihood [Equation (1)]. All of these likelihoods were inferior to that of our inferred network.

Our example demonstrates that dynoNEMs are able to learn network structures, which are in agreement with the observed phenotypic data.

3.2.2 Complete network In the last step, we applied MovieNEM in the same way as described above to infer the complete network between the 22 genes with significant phenotype (Supplementary Figs S12–S16). As mentioned above, these 22 genes are mainly involved into cell cycle, transcriptional regulation and cell differentiation. Again our edge-wise posterior expectation network scored better than 1000 random S-gene permutations.

We investigated, in how far our edge-wise posterior expectation network contained edges, which were explainable by paths contained in the literature network (see Supplementary Materials). This way, all 122 edges could be mapped to known literature pathways (see Excel files in Supplementary Materials). Moreover, out of these pathways, ~82% had a length shorter than 3 (Supplementary Fig. S14). On the other hand and not surprisingly, the literature network contained additional interactions, which could not be observed in our estimated networks. In consequence, ~60% of all paths between the 22 genes within the literature network corresponded to paths within our estimated network. This can have two reasons: either the additional literature-known interactions exist in reality, but MovieNEM could not infer them or they do not exist in HeLa cells and are hence not inferred. An important factor for false negatives is that we can only infer interactions between genes that show a clear phenotypic knock-down effect. If, for example, proteins A and B can independently activate C, then the knock-down effect of A may be partially compensated by B. In this case, we would only see a weak or no knock-down phenotype of A and consequently the edge $A \rightarrow C$ could be missed in our MovieNEM model.

According to our inferred model, specifically CDK1, CDC25A, CDKN1A and E2F5 are central hub nodes. All these proteins play an important role into G1/S and (except for E2F5) G2/M phase transitions according to their GO annotation. CDC25A, E2F5 and CDK1 have an extraordinary high in-degree, whereas CDKN1A has a high number of out-going interactions. CDKN1A and CDK1 are both involved into Ras signaling. It is known that CDKN1A binds to and inhibits

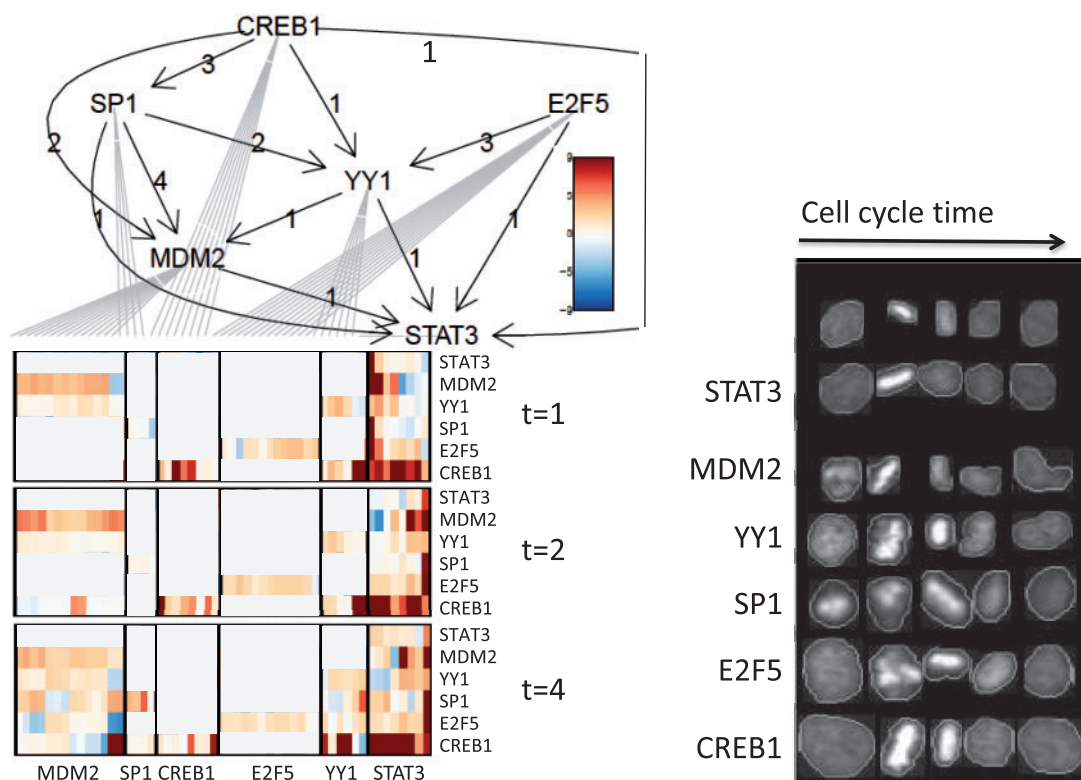


Fig. 4. Inferred MovieNEM model (edge-wise posterior expectation) for six sample genes (top). Inferred edges are not necessarily direct physical interactions, but can represent whole pathways. Heatmaps (bottom) depict estimated perturbation effects ($\log f_1(P - \text{value})$)—compare Section 2.2) at different time points. The ordering of phenotypic features in the heatmaps is due to the MovieNEM model, and gray lines indicate the maximum likelihood assignment of features to perturbed genes. For clearer demonstration purposes, features sets being unaffected by a perturbation are plotted in gray here. The original figure can be found in the Supplementary Materials. The right-hand side of the figure depicts examples of morphological phenotypes along the cell cycle for each of the six gene knock downs

CDK1 (Dulic *et al.*, 1998), which is in agreement with the inferred edge $CDKN1A \rightarrow CDK1$ in our model. E2F5 is a transcription factor playing a role into cell cycle control (GO annotation) and is inhibited by CDKN1A (Dimri *et al.*, 1996), which is reflected by the edge $CDKN1A \rightarrow E2F5$ in our inferred network.

4 CONCLUSION

We have shown that it is possible to learn pathway structures from phenotypic perturbation effects recorded in time-lapse movies. At the heart of the method lies the extraction of morphological features yielding measurable differences in cell phenotypes. We have developed a method to quantify these differences such that an extended version of the dynoNEM method (Fröhlich *et al.*, 2011) is applicable. To do so, cell trajectories were automatically aligned to the standard cell cycle, and a likelihood ratio score for the detection of feature changes was calculated. Here, this was done by comparing the distribution of each feature across different cells in a control and treated population, which makes the detection of significances sensitive given the large number of observed cells. One might also imagine comparing each feature in a single treated cell against the population of control cells as an alternative. This would probably be

less sensitive, but could also be valuable, if cell by cell differences are of relevance.

We have developed a novel MCMC sampler for network structure learning to estimate the posterior likelihood of each interaction. Our method allows for the inclusion of prior knowledge in a Bayesian fashion. Our whole approach aims at the reconstruction of small networks on the order of 10 nodes, for which the search space of signaling graphs can be sampled representatively. For larger networks, possible solutions include the restriction of the number of possible interaction partners of each node as well as divide and conquer approaches (Fröhlich *et al.*, 2007, 2008), but this is subject to future research.

Our simulations demonstrate the high sensitivity and specificity of our approach. This is also reflected in our application to movies of 22 siRNA knock downs from the Mitocheck database, where all estimated interactions were explainable by literature-known pathways.

We detect and exploit phenotypic differences that develop in response to a perturbation of the cell. In our present application, all phenotypic differences are derived from merely one fluorescence staining of the cell (GFP-histone tagging). We expect a significant increase in power of our method if multiple stainings were used, and the individual fluorescence labels were indicative of the biological process under investigation.

A basic assumption of our present MovieNEM approach is that phenotypic perturbation effects should occur at some point during the cell cycle and then remain until the end of the cell cycle. This assumption may not be fulfilled in all cases. Even more, some perturbations may not yield a phenotypic effect at all. In such cases, MovieNEM is not applicable. Finally, it is worth emphasizing that MovieNEM up to now is certainly not an approach for large-scale network reconstruction. This is due to the huge network space (increasing exponentially with the number of network nodes), which makes sampling-based methods (as well as heuristic approaches) computationally slow.

The reconstruction of cellular networks affords a better insight into the organization of a cell. Targeted RNAi screens have revealed their great potential for this purpose during the past years. Given the fact that the combination with omics-data is not always possible or affordable, image-based techniques can offer a promising alternative. Our MovieNEM approach is therefore a step toward better exploiting this information-rich data to uncover biological mechanisms.

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

Conflict of Interest: none declared.

REFERENCES

- 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.
- Bakal,C. *et al.* (2007) Quantitative morphological signatures define local signaling networks regulating cell morphology. *Science*, **316**, 1753–1756.
- Boland,M.V. *et al.* (1998) Automated recognition of patterns characteristic of sub-cellular structures in fluorescence microscopy images. *Cytometry*, **33**, 366–375.
- Chen,X. *et al.* (2006) Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. *IEEE Trans. Biomed. Eng.*, **53**, 762–766.
- Dimri,G.P. *et al.* (1996) Inhibition of e2f activity by the cyclin-dependent protein kinase inhibitor p21 in cells expressing or lacking a functional retinoblastoma protein. *Mol. Cell. Biol.*, **16**, 2987–2997.
- Driessche,N.V. *et al.* (2005) Epistasis analysis with global transcriptional phenotypes. *Nat. Genet.*, **37**, 471–477.
- Dulic,V. *et al.* (1998) Nuclear accumulation of p21cip1 at the onset of mitosis: a role at the g2/m-phase transition. *Mol. Cell. Biol.*, **18**, 546–557.
- 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**, 386.
- 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.* (2011) Fast and efficient dynamic nested effects models. *Bioinformatics*, **27**, 238–244.
- Fröhlich,H. *et al.* (2009) Nested effects models for learning signaling networks from perturbation data. *Biom. 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.
- Kaderali,L. *et al.* (2009) Reconstructing signaling pathways from RNAi data using probabilistic boolean threshold networks. *Bioinformatics*, **25**, 2229–2235.
- Kamentsky,L. *et al.* (2011) Improved structure, function and compatibility for cellprofiler: modular high-throughput image analysis software. *Bioinformatics*, **27**, 1179–1180.
- Kanabar,P. *et al.* (2009) Inferring disease-related pathways using probabilistic epistasis model. In: *Pacific Symposium on Biocomputing*. Vol. 14, pp. 480–491.
- Klamt,S. *et al.* (2010) TRANSWESD: inferring cellular networks with transitive reduction. *Bioinformatics*, **26**, 2160–2168.
- Maathuis,M. *et al.* (2009) Estimating high-dimensional intervention effects from observational data. *Ann. Stat.*, **37**, 3133–3164.
- Maathuis,M.H. *et al.* (2010) Predicting causal effects in large-scale systems from observational data. *Nat. Methods*, **7**, 247–248.
- Malpica,N. *et al.* (1997) Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, **28**, 289–297.
- Markowitz,F. *et al.* (2005) Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics*, **21**, 4026–4032.
- Markowitz,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.
- Neumann,B. *et al.* (2010) Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, **464**, 721–727.
- Niederberger,T. *et al.* (2012) Mc eminem maps the interaction landscape of the mediator. *PLoS Comput. Biol.*, **8**, e1002568.
- Otsu,N. (1979) A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.*, **9**, 62–66.
- Pe'er,D. *et al.* (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17** (Suppl. 1), S215–S224.
- 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*, **208**, 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 Markowitz,F. (2008) Structure learning in nested effects models. *Stat. Appl. Genet. Mol. Biol.*, **7**, Article 9.
- 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. Bioinform. Syst. Biol.*, 195272.