Estimating Large Scale Signaling Networks through Nested Effects Models from Intervention Effects in Microarray Data

Holger Fröhlich, Mark Fellmann, Holger Sültmann, Annemarie Poustka, Tim Beissbarth

German Cancer Research Center (DKFZ), Im Neuenheimer Feld 580, 69120 Heidelberg

{h.froehlich, m.fellmann, h.sueltmann, a.poustka, t.beissbarth}@dkfz-heidelberg.de

Abstract: Targeted interventions in combination with the measurement of secondary effects can be used to computationally reverse engineer features of upstream non-transcriptional signaling cascades based on the nested effect structure. We extend previous work by Markowetz et al., who proposed a statistical framework to score hypothetical networks for this purpose. Our extensions go in several directions: We show how prior assumptions on the network structure can be incorporated into the scoring scheme by defining appropriate prior distributions on the network structure as well as on hyperparameters. A new approach called module networks is introduced to scale up the original approach, which is limited to around 5 genes, to infer large scale networks (up to more than 50 genes). Instead of the data discretization step needed in the original framework, we propose the usage of a beta-uniform mixture distribution on the p-value profile, resulting from differential gene expression calculation, to quantify effects. Extensive simulations on artificial data and application of our module network approach to infer the signaling network between 10 genes in the ER-α pathway in human MCF-7 breast cancer cells show that our approach gives sensible results. Using a bootstrapping approach this reconstruction is found to be statistically stable.

1 Introduction

The advent of RNA silencing enables researchers to selectively silence genes of interest on large scale. DNA microarrays allow to measure the effects of a perturbation on a genome wide scale. This enables to reverse engineer interdependencies between gene products on a non-transcriptional level. The genes of interest are silenced individually, and the respective downstream effects on gene expression are measured by using genome-wide microarrays. By observing the nested structure of significant up or down regulations of affected genes, this allows to reconstruct features of the upstream signaling pathway [BAP02].

In a recent work Markowetz et al. [MBS05] introduced nested effects models as method to reverse engineer the signaling pathway flow between perturbed genes using the nested subset relationship of secondary downstream effects. They developed a Bayesian statistical framework, in which for a given network hypothesis one can calculate a score and thus
can reduce the set of all possible networks to the most likely ones. A severe limitation of this method lies, however, in the restriction to small networks of up to 5 genes, because the method completely enumerates all possible network hypotheses. Furthermore, a difficulty in the practical use is the required binary discretization of the data (“secondary effect present / not present”).

In our work we extend the framework by Markowetz et al. in several directions in order to overcome these restrictions: Instead of the data discretization step needed in the original framework, we propose the usage of a beta-uniform mixture distribution on the p-value profile, resulting from differential gene expression calculation, to quantify effects [PM03]. Moreover, we show how prior assumptions on the network structure can be incorporated into the network scoring scheme by defining appropriate prior distributions on the network structure as well as on its hyperparameter. Finally, and most important we present our so-called module networks to scale up the original approach, which is limited to small pathways with around 5 genes, to the inference of large scale networks (up to more than 50 genes). The idea is to build the complete network recursively from smaller pieces that are connected subsequently. In order to validate our approach we conduct extensive simulations on artificially created networks and compare it to the triplets inference scheme described in [MKTS07]. We show that module networks offer a better performance in terms of reconstruction quality while being significant computationally faster at the same time. We also apply our module networks infer the signaling network between 10 genes in the ER-α pathway in human MCF-7 breast cancer cells. Using a bootstrapping approach this reconstruction is found to be statistically stable.

2 Methods

2.1 Original Approach

We start with a brief review of the framework by Markowetz et al.: In general one distinguishes between silenced genes (S-genes) and genes showing a downstream effect (E-genes). It is assumed that each E-gene is attached to a single S-gene only (Fig. 1). Knocking down a specific S-gene $S_k$ interrupts signal flow in the downstream pathway, and hence an effect on the E-genes attached to $S_k$ and all S-genes depending on $S_k$ is expected. Let us assume $n$ knock-downs are performed and there exist $m$ E-genes in total. The outcomes of these experiments are summarized in an $m \times n$ data matrix $D$. According to Bayes’ formula a specific network hypothesis $\Phi \in \{0, 1\}^n \times \{0, 1\}^n$ can be scored as:

$$P(\Phi | D) = \frac{P(D | \Phi) P(\Phi)}{P(D)}$$  \hspace{1cm} (1)

The position of the E-genes is introduced as a model parameter $\Theta = \{\theta_i | \theta_i \in \{1, \ldots, n\}, i = 1, \ldots, m\}$, i.e. $\theta_i = j$, if E-gene $i$ is attached to S-gene $j$. Assuming independence of the observations (rows) $D_i$ in the data matrix $D$ (given a fixed network hypothesis $\Phi$ and
model parameters $\Theta$) one can write down the conditional likelihood $P(D|\Phi, \Theta)$ as:

$$P(D|\Phi, \Theta) = \prod_{i=1}^{m} P(D_i|\Phi, \theta_i)$$

(2)

It is furthermore assumed that all parameters $\theta_i$ are statistically independent, i.e.

$$P(\Theta|\Phi) = \prod_{i=1}^{m} P(\theta_i|\Phi)$$

(3)

The likelihood $P(D|\Phi)$ can now be written as:

$$P(D|\Phi) = \int P(D|\Phi, \Theta)P(\Theta|\Phi)d\Theta$$

(4)

$$= \prod_{i=1}^{m} \sum_{j=1}^{n} P(D_i|\Phi, \theta_i = j)P(\theta_i = j|\Phi)$$

(5)

2.2 Extensions

2.2.1 Generalized Inference Framework

Markowetz et al. suppose the data matrix $D$ to be discretized in binary values (“E-gene shows an effect / shows no effect”) and assume user specified type-I and type-II error rates for this procedure. The choice of these parameters is certainly critical for the inference procedure, because it directly influences (5) and appears to be difficult to estimate, especially in a two color cDNA microarray experiment with dye swaps (see section 3.2). In contrast, in our approach we make the assumption that $D$ is an $m \times n$ matrix of (raw) $p$-values, which specify the likelihood of E-gene $i$ being differentially expressed after knock-down of S-gene $k$. The $p$-values are calculated using an arbitrary method for detecting differential gene expression.
We now suppose a decomposition of $P(D_i|\Phi, \theta_i)$ as follows:

$$P(D_i|\Phi, \theta_i) = \prod_{k=1}^{n} P(D_{ik}|\Phi, \theta_i)$$

In accordance to [MBS05] this makes the assumption that knock-down experiments are statistically independent from each other. Hence, Eq. (5) can be written down as

$$P(D|\Phi) = \prod_{i=1}^{m} \sum_{j=1}^{n} \prod_{k=1}^{n} P(D_{ik}|\Phi, \theta_i = j) P(\theta_i = j|\Phi)$$

For the prior $P(\theta_i = j|\Phi)$ on the E-gene positions we suppose a discrete probability distribution.

The only thing missing is the definition of $P(D_{ik}|\Phi, \theta_i)$. For this purpose we suppose the $D_{ik}$ to be drawn from a mixture of a uniform $[0, 1]$ distribution reflecting the null hypothesis and another distribution $f_1$ reflecting the alternative hypothesis [PM03]:

$$P(D_{ik}) = \gamma_k + (1 - \gamma_k) \cdot f_1(D_{ik}), \gamma_k \in (0, 1)$$

Under the alternative hypothesis there is a high density for small $p$-values and a strong decrease for increasing $p$-values. Both distributions overlap with mixing coefficient $\gamma_k$. $P(D_{ik}|\Phi, \theta_i)$ can therefore be decomposed as:

$$P(D_{ik}|\Phi, \theta_i) = \begin{cases} f_1(D_{ik}) & \text{if } \Phi \text{ predicts an effect} \\ 1 & \text{otherwise} \end{cases}$$

The density function $f_1$ reflects the strength of the knock-down effect on E-gene $i$ under the alternative hypothesis. If it is greater 1 the alternative hypothesis would be accepted, and if it is smaller 1 rejected. In this work we assume $f_1$ to be a Beta$(1, \beta_k)$ distribution. The parameter $\beta_k$ can be obtained by fitting a beta-uniform mixture model (Eq. 8) via an EM-algorithm to the $p$-value distribution of knock-down experiment $k$ [DLR77].

### 2.2.2 A Bayesian Prior on the Network Structure

Eq. (1) allows to specify a prior $P(\Phi)$ on the network structure itself. This can be thought of biasing the score of possible network hypotheses towards prior knowledge. Given some assumed network structure $\hat{\Phi}$ we choose a prior of the form

$$P(\Phi) = \prod_{i,j}^{n} P(\Phi_{ij})$$

We set $P(\Phi_{ij}|\nu) = \frac{1}{2\nu} \exp\left(|\Phi_{ij} - \hat{\Phi}_{ij}|/\nu\right)$, i.e. we suppose $\Phi_{ij}$ to have a Laplacian distribution around $\hat{\Phi}_{ij}$. This choice allows us to specify our prior belief for each edge.
i \rightarrow j$ separately and therefore gives us a high flexibility in encoding prior knowledge or assumptions. An important special case would be $\Phi = 0$, i.e. the matrix only consisting of zeros, which prefers sparse networks. For the parameter $\nu$ we specify an inverse gamma distribution prior with hyperparameters $\alpha'$ and $\beta'$ to favor small values of $\nu$ (i.e. strong belief in the prior $P(\Phi_{ij} | \nu)$). The full prior $P(\Phi)$ can then be obtained via marginalization:

\[ P(\Phi) = \int_{0}^{\infty} \frac{1}{2^\nu} \exp\left(-\sum_{i,j} |\Phi_{ij} - \hat{\Phi}_{ij}| / \nu\right) \frac{\beta'^{\nu + \alpha' - 1} \exp(-\beta'/\nu)}{\Gamma(\alpha')} d\nu \] (11)

\[ = \frac{\alpha'}{2\beta'} \left(1 + \frac{\sum_{i,j} |\Phi_{ij} - \hat{\Phi}_{ij}|}{\beta'}\right)^{-\alpha'-1} \] (12)

where $\Gamma(\cdot)$ is the gamma function. In practice set $\alpha' = 10 \times n^2$ and $\beta' = 0.01 \times n^2$.

### 2.2.3 Large Scale Network Inference

The inference framework (Sec. 2.1, 2.2.1) does not answer the question how to come up with a candidate network topology, which we would like to score. Markowetz et al. [MBS05] completely enumerate all possible topologies. This is, however, only suitable for small networks of up to 5 S-genes. For 5 S-genes there already exist more than 1,000,000 and for 10 genes more than $10^{27}$ possible network topologies. In this context it should be noted that the scoring scheme (Sec. 2.1) cannot distinguish between two network hypotheses, if they only differ in transitive edges. This issue is known as prediction equivalence and is due to the fact that subset relationships, which are represented by a nested effects model, are transitive in principle. Hence, it only makes sense to consider the set of all transitively closed network hypotheses. However, restricting ourselves to this limited class of network structures does not generally solve the problem, since even then the number of networks to consider scales in a similar way with the number of S-genes. Hence, we have to resort to heuristics.

**Module Networks**

Rather than looking for a complete network hypothesis at once the idea of the module network is to build up a graph from smaller subgraphs, called *modules* in the following. The module network is thus a divide and conquer approach: We first split the complete node set into smaller subgroups. This can be done by using Ward’s clustering method on the density profiles of the S-genes. The idea is that S-genes with a similar E-gene response profile (here: with regard to the Manhattan distance) should be close in the signaling path. The number of clusters for the Ward clustering is chosen between 2 and half of the number of S-genes such that the average silhouette index becomes maximal. The silhouette value for each point in a cluster is a measure of how similar that point is to points in its own cluster vs. points in other clusters, and ranges from -1 to +1 [Rou87]. It is defined as:

\[ S(i) = \frac{\min_j(d_B(i,j)) - d_W(i)}{\max(d_W(i), \min_j(d_B(i,j)))} \] (13)
where $\bar{d}_W(i)$ is the average distance from the $i$-th point to the other points in its own cluster, and $\bar{d}_B(i, j)$ is the average distance from the $i$-th point to points in another cluster $j$.

Each cluster of S-genes now forms one module. These modules are eventually further subdivided into smaller submodules until each submodule contains only 4 S-genes at most. This way we obtain a tree structure of modules, where each node (module) has children (submodules). We begin with estimating the leaves in the module tree. As each leaf module can contain 4 S-genes at maximum this can be done by enumerating all possible transitively closed network hypotheses and taking the highest scoring one. After the leaves in the module tree have been built, their connections are estimated. For this purpose we test all pairwise connections between any pair of S-genes from leaves $L_1$ and $L_2$. Denoting by $|L_1|$ and $|L_2|$ the number of S-genes in both leaves, these are $4 \cdot |L_1| \cdot |L_2|$ tests altogether, because between any pair of S-genes $(n_1, n_2)$ we can either have no edge, an edge from $n_1$ to $n_2$, an edge from $n_2$ to $n_1$ or an edge in both directions. After the connection between $L_1$ and $L_2$ has been estimated, the corresponding subgraph is transitively closed. After all connections between leaves belonging to the same submodule in the module tree have been established, we recursively continue with connecting submodules in the same fashion as we did for leaf modules until the topology for the total network is completed.

3 Experiments

3.1 Large Scale Inference: Evaluation on Artificial Networks

To test our methods and to get better insights into the performance of our large scale inference methods, we generated data from artificial random networks. The sampling procedure is described in the Appendix. We sampled networks with $n = 8, 16, 64$ S-genes. For each number of S-genes we varied the number $m = n, 2n, 3n, 4n$ of E-genes and the parameter $\beta = \beta_k = 10, 50, 100 (k = 1, \ldots, n)$ of the $f_1$ distribution (Eq. 8). We compared the module network with the triplets inference approach described in [MKTS07]. The idea of the latter is to test all $\binom{n}{3} \cdot 29$ transitively closed triplet models in the graph to build up the complete network, i.e. individual edges are estimated in a redundant fashion. No prior knowledge on the network structure was used. We evaluated both methods in terms of average sensitivity (i.e. ratio of correctly learned edges to total number of edges in the original network) and specificity (i.e. ratio of correctly unconnected genes to total number of unconnected genes in the original network) over 100 generated networks for each parameter combination $(n, m, \beta)$.

Due to restrictions of space we only show the plots for $n = 16$ and $n = 64$ here (Fig. 2, 3). As seen in Fig. 2 the triplet inference offers a slightly higher specificity (due to the redundant estimate of individual edges), whereas with module networks sensitivity is significantly better. Moreover, both methods are very robust against a varying number of E-genes. For $m = 64$ E-genes the balanced accuracy (average of sensitivity and speci-
Figure 2: Sensitivity, specificity and balanced accuracy for randomly generated networks with $n = 16$ S-genes: $\beta = 100$ (solid), $\beta = 50$ (dashed), $\beta = 10$ (dotted). Top: module network; Bottom: triplet inference.

Table 1: Computation times (s) for module networks (MN) and triplet inference approach. For $n = 64$ triplet inference becomes prohibitively slow.

<table>
<thead>
<tr>
<th>method</th>
<th>$n = 8$</th>
<th>$n = 16$</th>
<th>$n = 64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN</td>
<td>2.5 ± 1.4</td>
<td>7.3 ± 2.4</td>
<td>607.5 ± 236.8</td>
</tr>
<tr>
<td>triplets</td>
<td>3.4 ± 0.7</td>
<td>48.3 ± 12.5</td>
<td>NA</td>
</tr>
</tbody>
</table>

Sensitivity) for module networks was $93 \pm 5\%$ ($\beta = 100$) compared to $85 \pm 9\%$ with triplet inference. The difference is statistically significant ($p < 0.001$) according to a paired two-sided $t$-test. Likewise, the differences of the balanced accuracies for $\beta = 50, 10$ are statistically significant ($p < 0.001$). Therefore, module networks seem offer a higher robustness than triplet inference. Moreover, they are significantly faster to compute. The speed-up compared to triplet inference for $n = 8$ is around 35%, for $n = 16$ around 700%, and for $n = 64$ triplet inference becomes prohibitively slow (c.f. Tab. 1).

3.2 Application to RNAi Data from Human ER-$\alpha$ pathway

We applied the module network to infer the complete topology for a network of 10 silenced genes in the ER-$\alpha$ pathway. The 10 genes were selected from previous microarray studies...
in our department to be influenced by ER status in breast cancer patients. Each of the 10 genes was silenced individually using two different siRNAs, respectively, and the effect on gene expression was studied on whole genome cDNA microarrays. The data were generated in our department. Details are omitted here due to restrictions of space, but can be obtained from the authors. The number of E-genes is \( m = 395 \).

For the network reconstruction we had knowledge on 3 interdependences between E- and S-genes: For known interdependencies we set \( P(\theta_i = j|\Phi) = 1 \) and \( P(\theta_i \neq j|\Phi) = 0 \), while otherwise we have a uniform prior \( P(\theta_i = j|\Phi) = \frac{1}{n} \). Additionally we used literature knowledge for known interdependencies between S-genes. The prior for the network structure \( \hat{\Phi} \) was chosen such that \( \hat{\Phi}_{ij} = 1 \), if an interaction between S-gene \( i \) and S-gene \( j \) was expected and \( \hat{\Phi}_{ij} = 0.25 \) otherwise. That means missing a known interdependency was punished more than introducing an edge where nothing was known. It should be noted here that due to experimental circumstances in RNAi knock-down experiments and due to the used cell-line there might be deviances of the literature knowledge to the measured data.

To ensure the statistical stability of the inferred network we employed bootstrapping: We sampled \( m \) E-genes from the total set of E-genes 50 times with replacement and each time ran the module network for topology induction. At the end we only considered edges, which were found in more than 50% of all bootstrap trials. Fig. 4 shows our obtained network drawn as a transitively reduced graph\(^1\): Most of the edges were inferred with very high consistency, i.e. can be viewed as highly reliable. Many of the upstream-downstream relations known from the literature were also part of our inferred network. Interestingly, AKT1 in our network was strictly downstream of AKT2. On the other hand, BCL2, FOXA1 and HSPB8 were seen at the same level as AKT1. Both results need further experimental verification.

\(^1\)A transitive reduction \( G' \) of a directed graph \( G \) is defined as the minimal graph such that the transitive closure of \( G' \) is the same as the transitive closure of \( G \) [AAU72].
4 Conclusion

We proposed a method for reconstructing signaling pathways from secondary effects, which were observed on microarray after silencing genes of interest via RNAi. Our approach systematically extends and generalizes previous work by Markowetz et al: Instead of data discretization a beta-uniform mixture distribution on the $p$-value profile resulting from differential gene expression calculation was used, to quantify effects. A Bayesian prior on the network structure was developed to incorporate assumptions on the network structure. Moreover, an algorithm for large scale inference of signaling pathways was developed and evaluated in a systematic fashion on artificially created data. Our module networks, which recursively build up the complete topology from smaller pieces, where found to have a good network reconstruction sensitivity and specificity. At the same time they could be computed very quickly and therefore allowed for the inference of large scale networks of more than 50 genes. We used the module network to infer the signaling pathway for 10 genes in the ER-$\alpha$ pathway in human MCF-7 breast cancer cells and used a bootstrapping approach to ensure the statistical stability of the result. The induced edges in our inferred network were found with high consistency and could in many cases be also confirmed by the literature. Future biological experiments are planned to validate our reconstructed network in a systematic way. In conclusion of our results we think that our approach offers a scale-able, reliable and fairly general way for large scale inference of signaling pathways from secondary effects and therefore provides researchers with a valuable tool to gain insight into complex cellular processes.

The code for the module network inference method is available in the latest version of the $R$-package nem, which can be obtained from the Bioconductor homepage.
Appendix

Generation of Artificial Networks

Artificial random networks were generated as follows: For each node $S_k$ we randomly chose the number $o$ of outgoing edges between 0 and 3. We then selected $o$ nodes having at most 1 ingoing edge, connected $S_k$ to them and transitively closed the graph. Averaged over 100 random networks for $n = 8$ this procedure yielded an average node degree of $5.8 \pm 0.6$ (min. 0, max. 13). That means each node on average had 3 ingoing and 3 outgoing edges.

After network topology construction, the $mE$-genes were attached uniform randomly over all $S$-genes. We then simulated knock-downs of the individual $S$-genes. For those $E$-genes, where no effects were expected, the "p-values" were drawn uniform randomly from $[0, 1]$. For the others there was an independent prior effect probability depending on the distance $d$ to the "knocked-down" $S$-gene of $1 - \frac{d}{n-1}$, i.e. at the maximal achievable distance of $d = n - 1$ there was only a 50% chance to observe an effect. For each $E$-gene we threw a biased coin with the corresponding prior effect probability, and depending on the outcome the "p-value" was either again drawn uniform randomly from $[0, 1]$ or sampled from the distribution Beta$(1, \beta)$ and $\beta$ being specified manually (c.f. Sec. 2.2.1). After having generated all "p-values", they were processed by the density function Beta$(1, 100)$.

Acknowledgments

This work was funded by the National Genome Research Network (NGFN) of the German Federal Ministry of Education and Research (BMBF) through the platforms SMP Bioinformatics and SMP RNA. We thank Florian Markowetz, Rainer Spang, Andreas Buneß, Markus Ruschhaupt and Ruprecht Kuner for help and discussions, and Dirk Ledwinda for IT support.

References


