Reconstructing Consensus Bayesian Network Structures with Application to Learning Molecular Interaction Networks

Holger Fröhlich[1] and Gunnar W. Klau[2]

1 **University of Bonn, Bonn-Aachen International Center for IT, Algorithmic Bioinformatics**
Dahlmannstr. 2, 53113 Bonn, Germany
frohlich@bit.uni-bonn.de

2 **Centrum Wiskunde & Informatica (CWI), Life Sciences**
Science Park 123, 1098 XG Amsterdam, The Netherlands
gunnar.klau@cwi.nl

Abstract

Bayesian Networks are an established computational approach for data driven network inference. However, experimental data is limited in its availability and corrupted by noise. This leads to an unavoidable uncertainty about the correct network structure. Thus sampling or bootstrap based strategies are applied to obtain edge frequencies. In a more general sense edge frequencies can also result from integrating networks learned on different datasets or via different inference algorithms. Subsequently one typically wants to derive a biological interpretation from the results in terms of a consensus network. We here propose a log odds based edge score on the basis of the expected false positive rate and thus avoid the selection of a subjective edge frequency cutoff. Computing a score optimal consensus network in our new model amounts to solving the maximum weight acyclic subdigraph problem. We use a branch-and-cut algorithm based on integer linear programming for this task. Our empirical studies on simulated and real data demonstrate a consistently improved network reconstruction accuracy compared to two threshold based strategies.

1998 ACM Subject Classification J.3 Life and Medical Sciences, I.5 Pattern Recognition, G.1.6 Optimization, G.2.2 Graph Theory, G.3 Probability and Statistics

Keywords and phrases Bayesian Networks, Network Reverse Engineering, Minimum Feedback Arc Set, Maximum Acyclic Subgraph, Molecular Interaction Networks

Digital Object Identifier 10.4230/OASIcs.GCB.2013.46

1 Introduction

Reverse engineering of biological networks is key to the understanding of biological systems. The exact knowledge of interdependencies between genes and proteins not only provides deep insight into the functionality of a cell, but is crucial for the identification of drug targets for various diseases. Apart from literature driven approaches, a wide range of methods from statistics and machine learning for estimating regulatory networks from experimental data has been proposed. Examples thereof are static and dynamic Bayesian Networks (BNs) [5, 16, 18, 25, 14, 7], correlation and mutual information based algorithms, such as ARACNE [11, 27], and Gaussian Graphical Models (GGMs) [19, 13, 17, 22], see [12] for a review. In this paper our focus lies on Bayesian Networks as an established probabilistic graphical modeling approach for network reverse engineering.

German Conference on Bioinformatics 2013 (GCB'13).
Editors: T. Beißbarth, M. Kollmar, A. Leha, B. Morgenstern, A.-K. Schultz, S. Waack, E. Wingender; pp. 46–55

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

When learning BN structures from experimental data the uncertainty about individual network structures has to be taken into account. Each edge can only be inferred up to a certain probability. In a Bayesian sense this amounts to compute posterior probabilities, which can be approximated via Markov Chain Monte Carlo (MCMC) sampling techniques. From a frequentist point of view, bootstrapping [4] provides a possibility to assess edge confidences. Both approaches result in a matrix of relative frequencies for each edge, which then can be thresholded appropriately to decide which interactions are supported by the data with high confidence and which not. Similarly, a matrix of relative edge frequencies can result from integrating networks inferred on different datasets [8] or via different inference algorithms [10].

In any of these cases the choice of the threshold above which edges are considered to be relevant is typically highly subjective and can influence the subsequent biological interpretation significantly. In addition, a complicating but most often ignored issue is that applying a cutoff typically leads to network structures that are inconsistent with the original model assumptions for BNs. That means that the obtained network graph is no longer acyclic (more specifically: a completed partially directed acyclic graph). As a consequence this may result in a graph that is principally impossible to model via BNs. This can in turn lead to a reduced network reconstruction accuracy, because it only makes sense to choose BNs as a modeling approach, if indeed the true biological network can be assumed to be acyclic.

Our contribution in this paper is two-fold: First, we propose a method to avoid the selection of an arbitrary frequency cutoff by estimating the proportion of false positive edges. This results in an edge-wise score, which is essentially an adjusted log odds ratio. Second, we show that computing a score optimal consensus network structure in this new model amounts to solving the maximum weight acyclic subdigraph problem. We propose a branch-and-cut algorithm based on integer linear programming (ILP) for this task. Our simulation results indicate that we enhance the network reconstruction accuracy significantly in comparison to a threshold guided as well as the Consensus Bayesian Network algorithm by Steele and Tucker [21], which is not guaranteed to yield a valid completed partially directed acyclic graph (CPDAG) structure. The utility of our approach is further demonstrated by applications on learning the yeast heat-shock response network [24] as well as the yeast Raf signaling pathway structure [18]. Notably, our proposed approach is fast enough to run on a standard desktop computer within a few seconds.

2 Problem Definition

Given a set $\mathcal{G} = \{G_1, G_2, \ldots, G_\ell\}$ of ℓ Bayesian network structures on the same set of nodes V. For example, each $G_k \in \mathcal{G}$ may be inferred on a different dataset, a different data sub-set or via a different learning algorithm. We propose to find a consensus CPDAG $G^* = (V, E^*)$, which explains a maximal fraction of the observed edges. A CPDAG is defined as the set of all Markov equivalent directed acyclic graph (DAG) structures and can be represented conveniently by a partially directed graph (see [15] for an excellent description). It can be computed from a DAG by making all edges undirected that are neither part of a v-structure (i.e., a motif of the form $A \to C \leftarrow B$) nor by reversal introduce any new v-structure.

Let $F = (f_{ij}), i, j = 1, \ldots, n$ with $n = |V|$ be a matrix of observed relative edge frequencies. Then we define G^* as the CPDAG that characterizes the equivalence class of the maximum weight DAG $D^* = (V, A^*)$, where $A^* \subset V \times V$ and the weight of an edge $(i, j) \in V \times V$ is given by a suitable weight function derived from the frequencies. In other words, the task is to find the maximum weight acyclic subdigraph D^* of a complete digraph with edge weights

$w(F)$. A natural weight function is, for example, $w(ij) = f_{ij} - \theta$, for $1 \leq i, j \leq n$, where θ is a suitable frequency threshold. In the next section, we propose a more sophisticated weight function based on the expected false positive rate of edges.

Once D^* is computed, the CPDAG that characterizes the corresponding equivalence class can be calculated via the algorithm of Chickering [3]. It has to be emphasized that Bayesian Network structures can principally only be resolved up to these equivalence classes.

3 Expected FPR Adjustment

In the absence of any gold standard network the selection of a frequency cutoff θ becomes a practically difficult problem. We here suggest a rational approach based on the expected false positive rate of edges. The underlying idea is that relative edge frequencies $\{f_{ij}\}$ can be assumed to be drawn from a mixture of two beta distributions, namely $\text{Beta}(\alpha, 1)$, if in reality the edge exists, and $\text{Beta}(1, \beta)$, if in reality there is no such biological interaction. Correspondingly we have:

$$f_{ij} \sim \pi_0 \text{Beta}(1, \beta) + (1 - \pi_0)\text{Beta}(\alpha, 1) \tag{1}$$

The parameters of this mixture model can be estimated conveniently via an expectation maximization (EM) algorithm. We can now compute the posterior log odds ratio for each edge (i, j):

$$r_{ij} = \log \frac{\text{Beta}(f_{ij}, \alpha, 1)(1 - \pi_0)}{\text{Beta}(f_{ij}, 1, \beta)\pi_0} \tag{2}$$

If we predicted every edge with a positive log odds ratio, then the overall expected fraction of false positive edges would be given by the area under the $\text{Beta}(1, \beta)$ distribution, which is below the $\text{Beta}(\alpha, 1)$ distribution, see Figure 1 left. However, instead of choosing a log odds ratio cutoff of 0, we could also take any other one. In particular we can select a cutoff τ such that $\int_\tau^1 \text{Beta}(x, 1, \beta)dx = q$, where q is a prescribed false positive rate (here: 10%). This is equivalent to setting $w(ij) = r_{ij} - \tau$, for $1 \leq i, j \leq n$, which is an adjusted log odds ratio score.

4 Finding Score Optimal Network Structures

Finding the maximum weight acyclic subgraph is a well-known NP-hard problem. It is equivalent to the minimum weight feedback arc set problem, which was one of the 21 problems for which Karp showed hardness in his famous work [9]. We model the problem as an integer linear programming (ILP) formulation and solve it using branch-and-cut.

Given matrix W containing the weights of the complete digraph on n nodes, our goal is to find the maximum weight subgraph $D = (V, A^*)$ such that A^* contains no directed cycles. We therefore introduce binary variables $x \in \{0, 1\}^{|V \times V|}$ with the interpretation $x_a = 1$ for $a \in A^*$ and $x_a = 0$ otherwise. The ILP is as follows:

$$\begin{aligned} \max \quad & \sum_{a \in V \times V} w_a x_a && && (3) \\ \text{subject to} \quad & \sum_{a \in C} x_a \leq |C| - 1 && \text{for all directed cycles } C \\ & x_a \in \{0, 1\} && \text{for all } a \in V \times V \end{aligned}$$

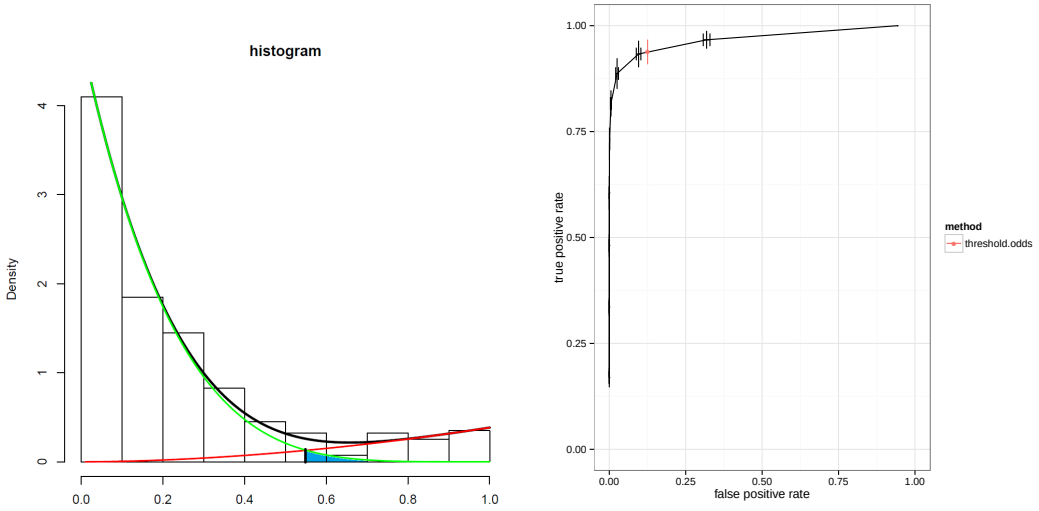

Figure 1 Left: Example of a fitted two component beta mixture model. Green = null distribution (edge does not exist in reality); red = alternative distribution (edge exists in reality); black = mixture. The shaded region indicates the expected false positive rate at a log odds ratio cutoff of 0. **Right:** ROC curve. True positive (TPR) and false positive rates (FPR) averaged over repeated data drawings for 20 networks. The plot shows the median TPR and FPR, respectively as well as the median absolute deviation (MAD) (depicted as error bars) for these 20 networks. Orange: point selected by the expected false discovery rate based method.

Since there is an exponential number of directed cycles in the complete digraph, we resort to a cutting plane approach to solve the linear programming relaxation of (3), which is the ILP without integrality constraints. This means, we initially leave out the directed cycle constraints and iteratively solve the following separation problem: Given a fractional solution $\bar{x}$ of an intermediate linear program, find a violated directed cycle inequality, that is, a directed cycle C with $\sum_{a \in C} \bar{x}_a > |C| - 1$ or state that no such violated inequality exists. This problem can be solved in polynomial time by defining $\bar{y} = 1 - \bar{x}$ and by computing shortest paths with respect to $\bar{y}$ between all pairs of nodes j and i. It is easy to see that a violated inequality is found if and only if the $\bar{y}$-weight of such a path plus $\bar{y}_{ij}$ is less than one. In this case, we add the inequality $\sum_{a \in C} x_a \leq |C| - 1$ and iterate. If no such cycle exists, we have solved the LP relaxation.

As we can solve the separation problem in polynomial time, we can also solve the full LP relaxation of (3) in polynomial time [6]. We use the LP bound within a branch-and-bound algorithm to obtain an optimal solution for the integer linear program and thus a maximum weight DAG in our input data. Subsequently, we compute the corresponding CPDAG with the algorithm of Chickering [3].

5 Results

5.1 Simulating KEGG Signaling Sub-Pathways

In order to test our approach we conducted simulation studies with sub-networks of KEGG signaling pathways with 10, 20, 40 and 60 nodes. We generated 20 directed, acyclic networks for each number of nodes. To obtain our ground truth networks we parsed XML files of all KEGG signaling pathways and converted them into graphs via the R-package KEGGgraph

[26]. Then we randomly picked one of these graphs and performed a random walk starting from a randomly selected core node. The random walk was stopped once the predefined number of distinct nodes had been visited. In case that the resulting sub-network between visited nodes contained cycles we repeated the whole procedure until a DAG structure had been found.

5.1.1 Network and Edge Frequency Sampling

We now simulated edge frequency matrices for the simulated KEGG DAGs by drawing for each existing edge from a Beta$(\alpha, 1)$ and for each non-existing edge from a Beta$(1, \beta)$ distribution for varying parameters α and β, see below. We made sure that $f_{ij} + f_{ji} \leq 1$ for each existing edge (i, j). This is necessary, because an undirected edge $\{i, j\}$ can never appear with a frequency of more than 100%. We repeated the simulation procedure five times for each network. For each frequency matrix we reconstructed a consensus Bayesian Network and calculated the corresponding CPDAG. We measured the performance in terms of balanced accuracy (BAC), i.e., average of sensitivity and specificity, by comparing the inferred CPDAG to the CPDAG of the original network. We conducted the comparison on the level of CPDAGs, because Bayesian Network structures can only be resolved up to these equivalence classes.

5.1.2 Effect of Edge Scoring Scheme

We exemplify the effect of our adjusted log odds scoring scheme compared to a varying cutoff applied to raw edge frequencies. This is done for a five times repeated simulation with $n = 20$ network nodes and $\alpha = 2$, $\beta = 10$. By varying the edge frequency threshold we obtain a ROC curve, which depicts true positive against false positive rates for network reconstruction (Figure 1 right). Our proposed edge scoring method corresponds to picking a point at the prescribed FPR of $\sim 10\%$ in the ROC curve. This demonstrates that our adjusted log odds scoring scheme can be used as an objective criterion to select a suitable edge frequency threshold.

5.1.3 Dependency on Number of Network Nodes

We went on to investigate the dependency of our proposed exact consensus Bayesian Network method (*exact*) on the number of network nodes for fixed $\alpha = 2$, $\beta = 10$. This was done in comparison to two other methods: (i) simple thresholding of the adjusted log odds ratio (*threshold*) at 0 and (ii) the consensus Bayesian Network algorithm proposed by Steele and Tucker (*CBN*) [21], which was also applied to adjusted log odds ratios here in order to have a fair comparison. Despite its name the CBN method does *not* guarantee to obtain a valid CPDAG structure, since it simply assigns each edge a direction based on a majority vote. The *threshold* method is equivalent to applying an edge frequency cutoff such that the expected FPR is 10%.

We observed a better network reconstruction performance of our *exact* method compared to both competing methods for all number of nodes. To assess the statistical significance of the observed differences we conducted one-sided paired Wilcoxon signed rank tests (with FDR multiple testing correction [1]). This indicated a highly significant result in all cases (Figure 2). Interestingly, significance levels increased with more network nodes. This may be explained by the larger space of CPDAG structures being consistent with the observed edge frequencies. The larger this space the more likely it is that a simple threshold based strategy (or variation thereof) fails. Hence, there is an increasing advantage of our *exact* approach.

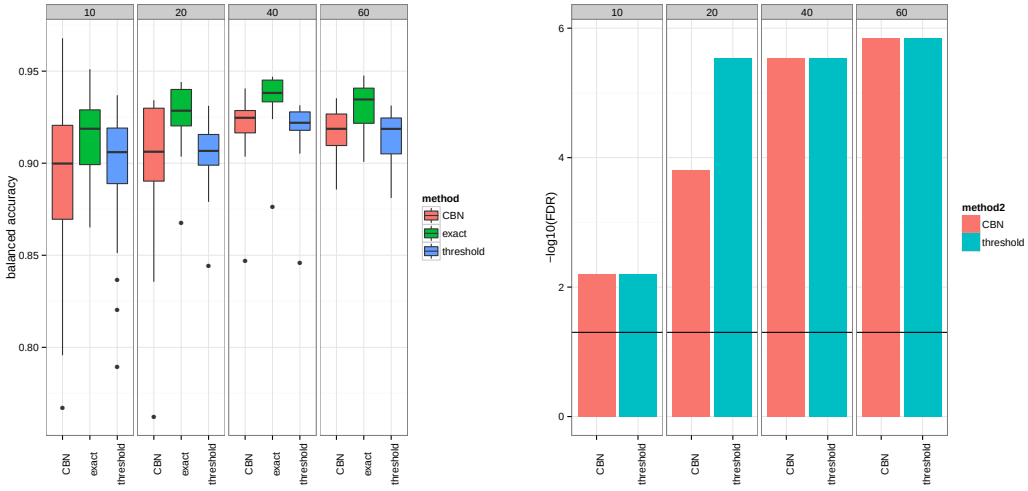

Figure 2 Left: Balanced accuracies (BACs) for network reconstruction with different consensus methods in dependency on number of network nodes. **Right:** Significance level ($-\log_{10}$ FDR) of *exact* versus *threshold* and *CBN*.

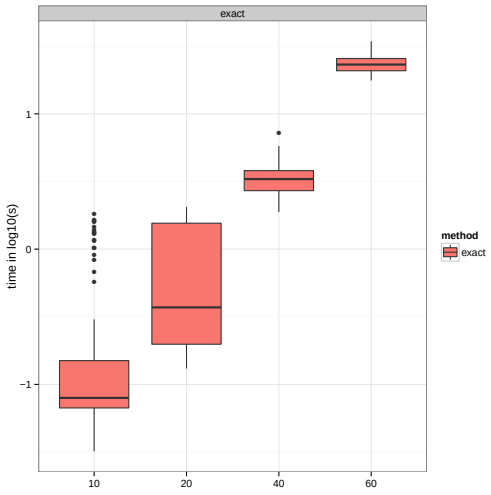

Figure 3 Empirical run time behavior for our *exact* method ($\log_{10} s$). Times were measured on an Intel Xeon 8 core machine with 96GB RAM.

Notably, even for networks with $n = 60$ nodes with our proposed method a consensus Bayesian Network that is provably optimal with respect to our scoring function could be found below 30 CPU s (Figure 3) on a standard desktop computer.

5.1.4 Dependency on Distribution Parameters

We repeated our above described simulation with different combinations of α and β for $n = 20$ nodes. This showed that in all but the extreme case ($\alpha = 2$, $\beta = 2$) a highly significant improvement of our method compared to *threshold* and *CBN* could be achieved (Figure 4). Please note that in the case ($\alpha = 2, \beta = 2$) the two beta distributions are very flat and

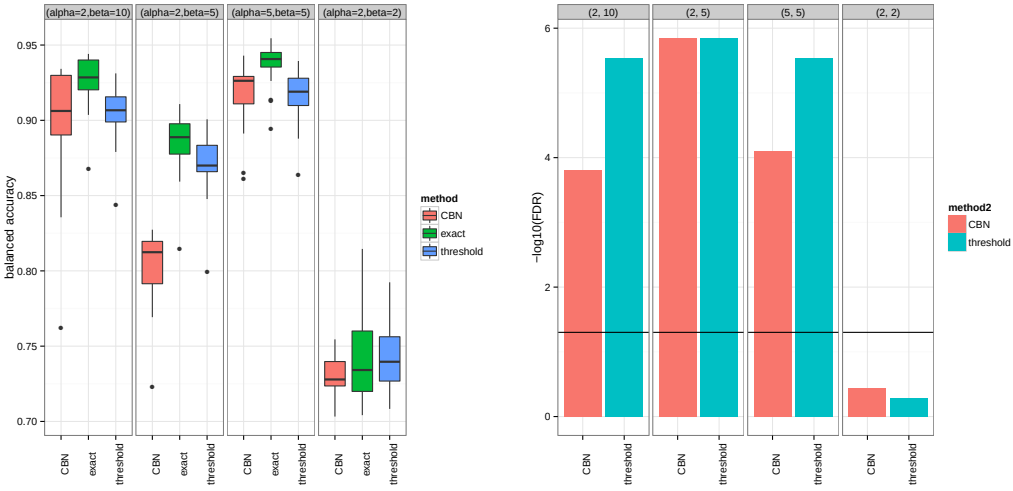

Figure 4 Left: Balanced accuracies (BACs) for network reconstruction with different consensus methods in dependency on different distribution parameter settings. **Right:** Significance level ($-\log_{10}$ FDR) of *exact* versus *threshold* and *CBN*.

extremely overlapping, making a reliable distinction of likely existing and non-existing edges highly difficult.

5.2 Yeast Heat-Shock Network

We applied all tested methods to a network of nine transcription factors (TFs) related to heat-shock response in yeast (Figure 5 left). The network was taken from [21]. We collected four microarray datasets (GSE3406, GSE3316, GSE40073, GSE40817) consisting of gene expression measurements after heat shock for varying conditions (GSE40817), time points (GSE3406, GSE3316, GSE40073) and strains or strain mutations (GSE3406, GSE40073). After k-NN imputation of missing values [23] quantile normalization was applied to each dataset [2]. We then used a quantile based discretization of each dataset. As variables in the Bayesian Network structure, which we wanted to learn from these datasets, we considered—besides the nine TFs—three additional variables, namely condition, time and strain. This was done, because these factors could potentially influence TF expression. The additional three variables were only allowed to have ingoing, but no outgoing edges. We then applied Bayesian Network learning using a greedy hill climber, as implemented in the R-package *bnlearn* [20]. This was done using a non-parametric bootstrap with 10,000 replicates. Accordingly we arrived at an edge frequency matrix, which we used to calculate a consensus network structure.

We compared the CPDAG of the consensus network on each of the four datasets against the CPDAG of the gold standard network from [21] and computed the balanced accuracies, see Figure 5 right. This demonstrates that our *exact* approach achieved the best performances on all four datasets. Interestingly, the performance of all methods showed a high variability across datasets. Whereas on dataset GSE3316 a balanced accuracy of 82% with our method could be achieved, on GSE3406 this was only 55%. The reason is probably the high number of different conditions, time and strain combinations coupled with a comparably low number of replicates in GSE3406.

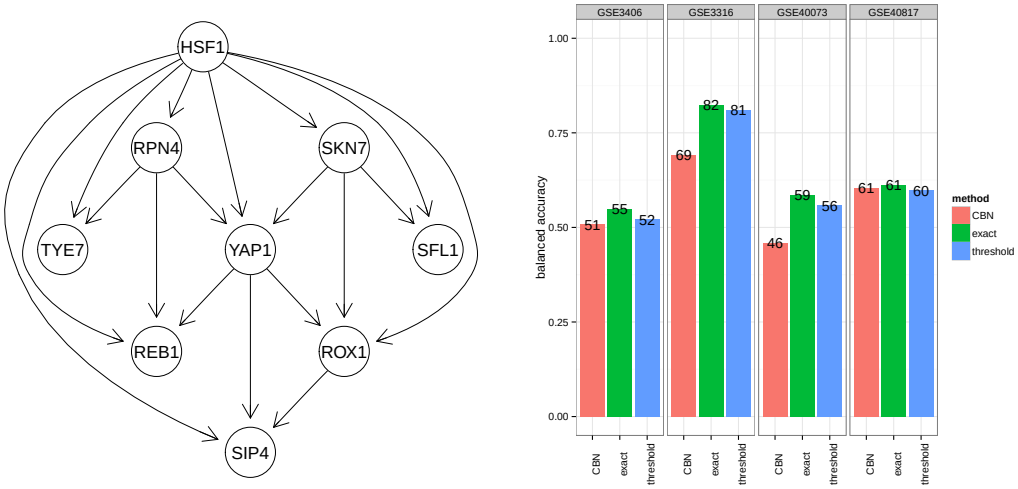

Figure 5 Left: Gold standard heat-shock network [21]. **Right:** Balanced accuracies for network reconstruction using a 10,000 times bootstrapped greedy hill climber and different consensus network approaches.

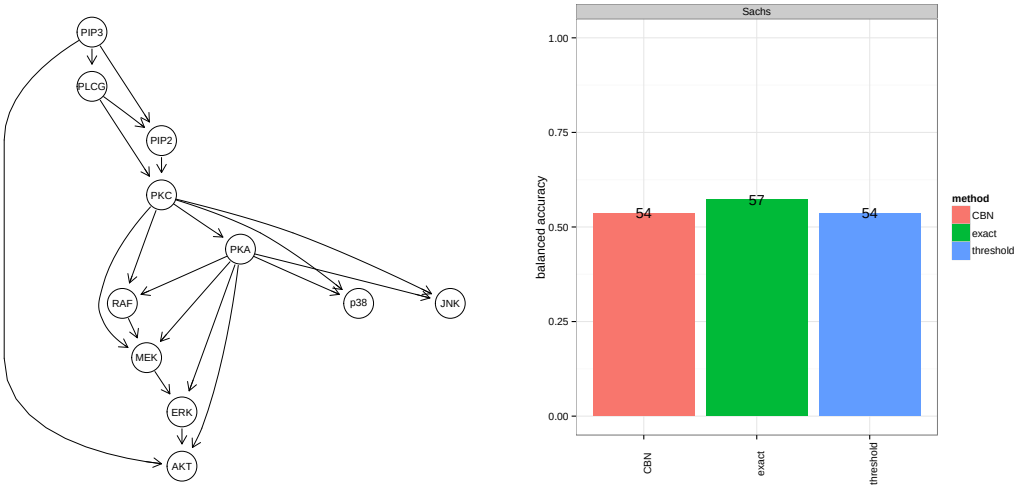

Figure 6 Left: Gold standard Raf signaling pathway [18] **Right:** Balanced accuracies for network reconstruction using a 10,000 times bootstrapped greedy hill climber and different consensus network approaches.

5.3 Yeast Raf Signaling Pathway

In addition we applied all tested methods to inferring parts of the yeast Raf signaling pathway based on the dataset by Sachs et al. [18], see Figure 6 left. The dataset contains measurements of 11 proteins under 14 different treatment conditions. The same technique for Bayesian Network inference as for the yeast heat-shock network with the same number of bootstraps was applied. Comparison of our *exact* consensus method against the *CBN* and *threshold* approaches yielded an improvement of 3% in terms of balanced accuracy (Figure 6 right).

6 Conclusions

In reverse engineering of biological networks edge frequency matrices can result from aggregating bootstrap or MCMC results, combining outputs of different inference algorithms [10] or integrating experimental datasets [8]. In order to interpret these meta results one usually has to come up with a way to compute a consensus network. So far most authors do this by applying a defined threshold to the edge frequencies, which is (i) highly subjective and (ii) leads to consensus models inconsistent with the model assumptions. Here we propose an automated threshold selection based on the expected false positive rate. This yields an adjusted log odds ratio as an edge-wise score. Based on this score we showed that computing provably score optimal consensus Bayesian Network structures amounts to solving the maximum weighted directed subgraph problem. We proposed a branch-and-cut algorithm based on an integer linear programming formulation for this task.

Our simulation studies as well as our results on two yeast networks showed that our new approach consistently improves network reconstruction accuracy. Our simulations showed that the expected gain increases with the number of network nodes. At this point we should emphasize that both of our tested yeast networks were comparably small. Higher significant results can thus be expected for larger datasets. Although the computation time of our proposed method scales exponentially with the number of network nodes we did not observe any practical limitation by this fact. Even with 60 nodes networks our algorithm took only a few seconds on a standard desktop computer. This should specifically be seen in relation to the high computation time needed for bootstrapping or sampling based network inference.

References

1 Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. Royal Statist. Soc., Series B*, 57:289 – 300, 1995.

2 B. M. Bolstad, Irizarry R. A., M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*, 19:185–193, 2003.

3 D. M. Chickering. Learning equivalence classes of Bayesian network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

4 A.C. Davison and D.V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, Cambdrige, UK, 1997.

5 N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J Comput Biol*, 7(3-4):601–620, 2000.

6 M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition, 1993.

7 M. Grzegorczyk and D. Husmeier. Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes. *Bioinformatics*, 27(5):693–699, Mar 2011.

8 C. Huttenhower, K. T. Mutungu, N. Indik, W. Yang, M. Schroeder, J. J. Forman, O. G. Troyanskaya, and H. A. Coller. Detailing regulatory networks through large scale data integration. *Bioinformatics*, Oct 2009.

9 R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., New York: Plenum*, pages 85–103, 1972.

10 D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, The Dream Consortium, M. Kellis, J. J. Collins, and G. Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8), 2012.

11 A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 Suppl 1:S7, 2006.

12 Florian Markowetz and Rainer Spang. Inferring cellular networks–a review. *BMC Bioinformatics*, 8 Suppl 6:S5, 2007.

13 N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

14 S. Mukherjee and T. P. Speed. Network inference using informative priors. *Proceedings of the National Academy of Sciences*, 105(38):14313–14318, 2008.

15 R. Neapolitan. *Learning Bayesian Networks*. Pearson Prentice Hall, Upper Saddle River, NJ 07458, 2004.

16 D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(Suppl 1):S215 – S224, 2001.

17 V. Pihur, S. Datta, and S. Datta. Reconstruction of genetic association networks from microarray data: a partial least squares approach. *Bioinformatics*, 24(4):561–568, Feb 2008.

18 K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, and G. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 208(5721):523 – 529, 2005.

19 J. Schäfer and K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, Mar 2005.

20 M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

21 E. Steele and A. Tucker. Consensus and meta-analysis regulatory networks for combining multiple microarray gene expression datasets. *J Biomed Inform*, 41(6):914–926, Dec 2008.

22 A. Tenenhaus, V. Guillemot, X. Gidrol, and V. Frouin. Gene association networks from microarray data using a regularized estimation of partial correlation based on PLS regression. *IEEE/ACM Trans Comput Biol Bioinform*, 7(2):251–262, 2010.

23 O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, Jun 2001.

24 Y. Wang, T. Joshi, X.-S. Zhang, D. Xu, and L. Chen. Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics*, 22(19):2413–2420, Oct 2006.

25 A. V. Werhli and D. Husmeier. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat Appl Genet Mol Biol*, 6:Article15, 2007.

26 S. Zhang, H. Chen, K. Liu, and Z. Sun. Inferring protein function by domain context similarities in protein-protein interaction networks. *BMC Bioinformatics*, 10:395, January 2009.

27 P. Zoppoli, S. Morganella, and M. Ceccarelli. TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11:154, 2010.