Module Extraction from Ensembles of Networks

Lukas Windhager[*], Robert Küffner, Jonas Zierer, Ralf Zimmer

Institut für Informatik, Ludwig-Maximilians-Universität München,
Amalienstrasse 17, 80333 München, Germany
Lukas.Windhager@bio.ifi.lmu.de

Abstract

We argue for the necessity to identify and attend cases of ill-constraint reverse-engineering problems where ensembles of predicted gene-regulatory networks consist of intermixed topology classes. A stepwise procedure is proposed, where first pairwise dependencies (associations) between interactions from ensemble predictions are derived by association rule learning and approximate reasoning. Pairwise associations are subsequently extended to derive variable modular subregions of networks. Modules identify topology classes and constitute single topology network ensembles if joined with core regions and variable single interactions. Finally, single topology ensembles constitute interpretable hypotheses about network structure and guide further experimental design.

Keywords: reverse-engineering, network ensembles, network modules, approximate reasoning, experiment design

1 Introduction

Reverse-engineering of gene regulatory networks from gene expression measurements is widely applied to identify direct effector-target relations, i.e. to identify transcription factors binding to the promoter regions of genes to regulate gene expression (for reviews see [1, 2, 3, 4, 5]). Dynamical model based reverse-engineering algorithms describe the actual effector-target relations by a mathematical framework (ODEs [6, 7, 8], Petri Nets [9, 10, 11], Boolean Nets [12, 13, 14], and others). Models are confronted with expression data from different perturbation scenarios (knock-outs, over-expression, chemical treatment) and the predicted gene expression levels are compared to experimental data to assess the validity of the models. The underlying basic assumption of reverse-engineering is that a dynamical model which is able to reproduce experimental observations (i.e. expression changes as reaction to perturbations) is a good (mathematical) approximation to the biological system in question [15]. Often, several models are (equally) good.

[*] Corresponding author.

Putative effector-target relations can then be directly and easily derived from the model, typically by simplifying functional relations to either up-regulating (activating) or down-regulating (inhibiting) effects. The final prediction is represented as a network of activating and inhibiting interactions between pairs of genes.

Dynamical models can be created and optimized based on iterative procedures (e.g. genetic algorithms, Monte Carlo methods, see [16, 17]) involving sampling or modification of model parameters, evaluation by a scoring function which assesses model complexity and fit to experimental data (usually involving model simulation). Probabilistic optimization is typically repeated several hundred or thousand times to create an ensemble of candidate networks. Several procedures have been proposed to derive effector-target relations from such ensembles (see discussion in [18] and references therein). Voting schemes (majority vote, weighted voting, signed voting) can be applied to derive scores for each possible effector-target relation. Scores (weights) typically correspond to the frequency of an interaction in the network ensemble. In the case of signed voting, scores might range from -1 (assured inhibiting interaction) via 0 (assured non-interaction) to 1 (assured activating interaction), where “assured” is synonymic to “observed in all relevant (high scoring) networks”.

On average, ensemble voting was found to be superior to the selection of a single, best fitting network in terms of precision, recall and robustness (see [18]). This is plausible when considering the nature of probabilistic optimization algorithms which might not find *the* optimal solution, but might converge to close, suboptimal parameterization. Reasons are, first and foremost, noise, errors and lack of experimental data, the limitations of the mathematical framework (or an inadequate choice of it), and inadequate penalization of model complexity. Let alone the common cases where the scoring function which guides model creation achieves its optimum not at the “true” parameterization (i.e. the one yielding the actual reference gene-regulatory network), but close to it at best. Often non-reference models simply predict the data better than the “true” model. Still, as true interactions are expected to be enriched in an ensemble of predicted high-scoring networks, a voting scheme which creates some kind of ensemble average will reveal them.

Flaws and Prospects of Ensemble Voting Ensemble voting is an adequate technique when applied to reverse-engineering problems with an ill-constraint search space, but only if the considered networks are sufficiently similar to each other (and also similar to the reference topology).

However, if there exist distinct classes of network topologies, which can not be clearly discriminated by their fit to experimental data and differ strongly in overall or local topology, then ensemble voting “leads to a meaningless blur of alternative structures” [18]. Although this flaw is known, it has not been systematically addressed yet.

We will briefly discuss how the presence of multiple topology classes influences ensemble averages, how a topology class can be defined by modules of interdependent interactions and that knowledge about modules is useful for interpretability and experimental design. The following discussion is exemplified by a small gene regulatory network (figure 1).

Assume that the scoring function is not restricted to a single optimum, but allows several different topologies, i.e. there are two or more similarly scoring optima. A probabilistic reverse-engineering algorithm would now create and accept models from all optima with roughly equal probabilities. This argument also holds for non-uniform probabilities, as long as no single topology is clearly favored.

The interactions derived from the ensemble average can now be divided into four classes:

- Class 1 interactions (score close to 1 or -1, low entropy) are present in nearly all ensemble networks, no matter what topology class they belong to.
- Class 2 interactions (score close to 0, low entropy) are missing in nearly all ensemble networks.
- Class 3 interactions (medium scores, high entropy) are present in nearly all ensemble networks of one topology class, but are not present in networks of others.
- Class 4 interactions (medium scores, high entropy) might appear in all topology classes or a subset of classes, but in both cases appear only in subsets of corresponding networks.

Class 1 interactions constitute the *core topology*. The search space is somehow restricted such that these interactions are apparently necessary to reproduce experimental data and thus are present in all relevant networks. The presence of class 2 interactions (or *rare* interactions) might contradict experimental data, thus is disfavored by the scoring function. They also might have a functionality which is already realized by class 1 interactions, so they are redundant and their presence would only increase model complexity. If the ensemble has been created by signed voting, a third explanation for scores close to 0 is possible. An effector-target relation might be present in most networks of both topology classes, but with opposite signs. This case is easily detectable by simply counting the presence of the relation, ignoring its sign. Such interactions would then be classified as class 3 interactions.

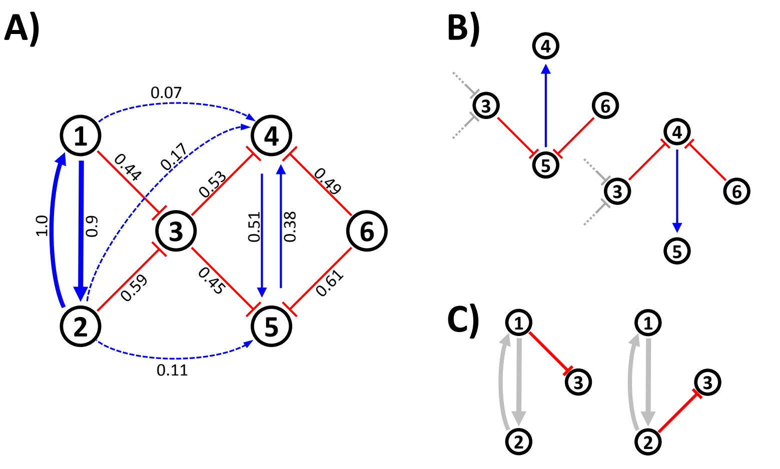

Figure 1: A) Ensemble average with annotated frequencies. Activating (blue), inhibiting (red). Class 1 interactions (bold) are present in nearly all networks. Class 2 interactions (dashed) are missing in most. B) Interactions connecting genes 3 to 6 constitute two modules (class 3 interactions). Modular structure is not clearly visible in the ensemble average. C) Interactions affecting gene 3 are mutually exclusive class 4 interactions.

Class 3 interactions are necessary and unique to a topology class. Thus the presence/absence of such an interaction can be used for classification. All class 3 interactions belonging to the same topology class constitute a *module*. Modules arise when two or more interactions are mutually dependent and their complete presence is necessary for a (biological) functionality (see figure 1B). Thus either all interactions of a module are present or none are (and a competing module exhibits this functionality instead). Notice that modules must contain at least two interactions affecting different targets, as otherwise they would be indistinguishable from class 4 interactions.

Class 4 interactions are *single variable* interactions. They typically arise from multiple effector candidates for a single target which cannot be distinguished by the available data, and the choice of the actual effector is *not* affecting effector-target relations of other targets (otherwise they would constitute a module). Notice that the presence of a class 4 interaction typically negatively affects the presence of another class 4 interaction if they share the target, as they exhibit a redundant functionality and increase model complexity (see figure 1C). Class 4 interactions constitute very variable sub-regions.

Scores of an ensemble voting give no reliable information about dependencies between interactions. Assigned scores alone can not distinguish between class 3 and class 4 interactions, nor can they be utilized to assign class 3 interactions to modules. The presence of interactions with medium scores is not even an indicator for the presence of modules.

Knowing the modularity of ensemble networks and dependencies (and independencies) between interactions has some clear-cut benefits. First, effective experimental design is possible if modules are known. As either all interactions of a module are present or all interactions of the competing one are, knowledge about the presence or absence of a single interaction is sufficient to identify which module is actually realized in the biological system. In contrast, knowledge about a single variable interaction might only affect a small number of interactions, namely those class 4 interactions affecting the same target.

Second, the results of reverse-engineering become interpretable. The "meaningless blur" disappears if models are grouped according to their topology class and not a single, but multiple ensemble averages are created by a voting scheme. Dependencies between interactions, previously hidden, become accessible for interpretation, namely the redundancies of shared-target interactions and interactions acting in concert as competing or complementary modules.

2 A Module Extraction Approach

The following section describes the details of an approach suited to classify interactions, extract dependencies and modules. It utilizes the properties of core regions, modules and single variable interactions described before. As variations between networks, although of the same topology class, are still very common in ensemble predictions, we decided to apply an approach which utilizes approximate reasoning to allow a certain amount of *noise* in the ensemble networks.

Step 1: Extraction of Core, Rare and Potential Module-Interactions Interactions are classified according to their relative frequency in the network ensemble as

1. interactions of core topology (class 1) if their relative frequency is above a cutoff,
2. rare interactions (class 2) if their relative frequency is below a cutoff,
3. potential module interactions (class 3 or 4) otherwise.

Only potential module interactions are considered for association rule learning and module creation in the following steps. Notice that the sign of an interaction (activating or inhibiting) is considered in this step. Thus zero, one or two potential module interactions can be present in the ensemble for each possible effector-target pair.

Step 2: Transactions and Association Rule Learning Each network is converted into a transaction, thus the network ensemble yields a database for association rule learning. For each target-effector pair P_i connected by any potential module interaction of the ensemble, an item $[P_i, s \in \{+,-,0\}]$ is added to a transaction, representing whether the respective network contains an activating, inhibiting or no interaction from this target to effector. Thus each transaction contains the same number of items, one for each observed target-effector pair.

Association rules are derived from the transaction database using a standard apriori-algorithm implementation [19, 20], restricted to rules with a single antecedent and single consequent only. This step yields a set of "$[P_i, s] \Rightarrow [P_j, t]$" rules, where $s, t \in \{+, -, 0\}$, as well as confidence and support for each rule following the standard definition for association rule mining [21].

Notice that the notion $[P_i, s]$ with $s \in \{+, -\}$ uniquely identifies an interaction, describing the regulatory relation between an effector and a target gene.

Step 3: Pairwise Relations and Module Extraction Association rules are utilized to derive AND and XOR associations for pairs of interactions. The confidence of an association of interactions $[P_i, s], [P_j, t]$ with $s, t \in \{+, -\}$ is defined as:

$$\begin{aligned}
\text{conf(AND)} &:= conf([P_i, s] \Rightarrow [P_j, t]) \otimes conf([P_j, t] \Rightarrow [P_i, s]).\\
\text{conf(XOR)} &:= conf([P_i, 0] \Rightarrow [P_j, t]) \otimes conf([P_j, 0] \Rightarrow [P_i, s]) \otimes\\
&\qquad conf([P_i, s] \Rightarrow [P_j, 0]) \otimes conf([P_j, t] \Rightarrow [P_i, 0]).
\end{aligned}$$

Where *conf()* is the confidence of an association rule or association and $\otimes$ is a conjunction operator. We apply the *min* operation. The confidence of either AND or XOR association has to exceed a cutoff to consider a pair of interactions AND or XOR related. Notice that if this cutoff is > 0.5, it is mathematically impossible that both confidences exceed it at the same time. An activating and inhibiting interaction connecting the same effector-target pair is considered to be XOR related by default. If none of these conditions hold, a pair of interactions is considered unrelated.

All interactions contained in any AND association are considered as an individual module, thus as class 3 interactions. Two modules M_a, M_b are merged if there is an AND association between any $[P_i, s] \in M_a$ and $[P_j, t] \in M_b$, but no XOR association. Thus an AND-related pair of interactions is always part of the same module. Modules with at least one XOR association between them are considered competing, i.e. one of these modules can be present in a predicted network, but not both. All XOR related interactions which are not part of modules are classified as mutually exclusive single variable interactions (class 4).

Step 4: Network Classification All networks of the ensemble are then classified according to the combination of modules they contain. E.g. if three modules A, B and C have been identified, where A and B are competing, then there are 6 possible module combinations which might be realized in a predicted network (no module at all, only A, only B, only C, A and C, B and C).

All networks of the same class constitute a (purified) single topology ensemble. The resulting single topology ensembles thus contain not only module interactions, but also include core topology, rare interactions and single variable interactions. Single-topology ensembles might in turn be checked for (sub)modules which are constituted by now possibly enriched rare interactions.

3 Evaluation Results

To create ensembles for subsequent module extraction and validation, we applied a genetic algorithm (GA) to reverse-engineer dynamical models based on Petri Nets and Fuzzy Logic (PNFL, [22, 23, 24]). This GA and PNFL were already successfully applied in the DREAM4 network reconstruction challenge [2]. In general, the performance of the applied reverse-engineering method is not crucial for the module extraction procedure, as long as a sufficiently large ensemble of predictions can be created, such that a few hundred predictions fit the reference data well enough to be considered as (potentially) valid hypotheses.

We created 500 random reference networks of 7 genes with a given indegree distribution of $(p_{ind}(1) = 0.7, p_{ind}(2) = 0.2, p_{ind}(3) = 0.1)$, i.e. all genes had between one and three effectors. For each reference, a steady state time-series and the effects of all single and four double knockout perturbations were simulated. Reverse-engineering was performed using the expression values of the steady state time-series and initial and final time-points of perturbations only. For each reference, 1000 predictions were created and the 20% with best fit to the simulated data were used for ensemble prediction (table 1).

A	B	C	D	E	F	G
7/7/4/500	3%	37%	0.61	0.22/1.0	14.2/12.0/26.2	11.7/ 6.3/18.0
7/7/2/200	3%	30%	0.60	0.16/1.0	15.1/13.9/29.0	12.3/ 7.4/19.9
7/7/0/200	5%	32%	0.71	0.36/1.0	15.9/15.2/31.1	12.4/ 8.0/20.4
7/5/0/200	2%	20%	0.56	0.22/0.9	16.4/18.1/34.5	13.2/10.7/23.9

Table 1: Performance of reverse-engineering and module extraction for four experimental settings. (A) Number of genes / number of single knockout experiments / number of double knockout experiments /number of references. (B) Percentage of cases where a predicted network was identical to the reference. (C) Percentage of cases where modules have been identified. (D) Average AUPRC of the full ensemble. (E) Minimum and maximum AUPRC of the full ensemble. (F) Average number of class 1 + class 2 / class 3 + class 4 / all interactions of the full ensemble and (G) of single topology ensembles.

The presented evaluations are based on the predictions for the 500 reference networks. Additionally, 3 times 200 reference networks with a varying number of single- and double-knockouts were created and reverse-engineered for comparison. Their evaluation yields similar results (data not shown).

The following cutoffs were applied during module extraction: class 1 and class 2 interaction frequency cutoffs 0.8 and 0.1, association rule support cutoff 0.1, AND and XOR association confidence cutoff 0.8.

About 30% of predicted network ensembles contained identifiable modules (table 1C). Contrary to the full ensemble (figure 2B), interdependent interactions are clearly visible in the single topology ensembles (figure 2CD). Derived single topology ensembles reflect possible hypotheses about the reference structure. Specific testing of few module-interactions allows to test for these alternative hypotheses.

An ensembles entropy can be used as a measure of its fuzziness, i.e. ensembles with a large proportion of class 3 and class 4 interactions (medium frequencies, high entropy) have a higher total entropy compared to ensembles of the same size with many class 1 and class 2 interactions (high or low frequencies, low entropy). Ensemble entropies were defined as $-\sum_i \log_2(f(P_i)) \cdot f(P_i)$, where $f(P_i)$ is the frequency of interaction i. Single topology ensemble entropies are in general reduced compared to the according full ensemble entropies (on average by 50%, figure 3A). The total number of interactions is reduced, as well as the proportion of class 3 and class 4 interactions to class 1 and class 2 interactions (table 1F and G). Thus, single topology ensembles are less “blurry” than the according full ensembles.

Network ensembles can be compared to the reference topology by calculating the area under the precision recall curve (AUPRC). AUPRC values range between 1 (all reference interactions are top-ranked in the ensemble) and 0 (no reference interaction is present in the ensemble). If two (or more) mutually exclusive modules have been identified, it can be assumed that only one of them is actually present in the reference network. Thus, only some of the resulting single topology ensembles should exceed the full network ensemble in terms of AUPRC (figure 3B).

To assess whether experimental design can be guided using knowledge about modules and single variable interactions, we checked how knowledge about the presence of individual interactions influences the AUPRC of network ensembles. For each class 3 and class 4 interaction, we checked whether it was actually

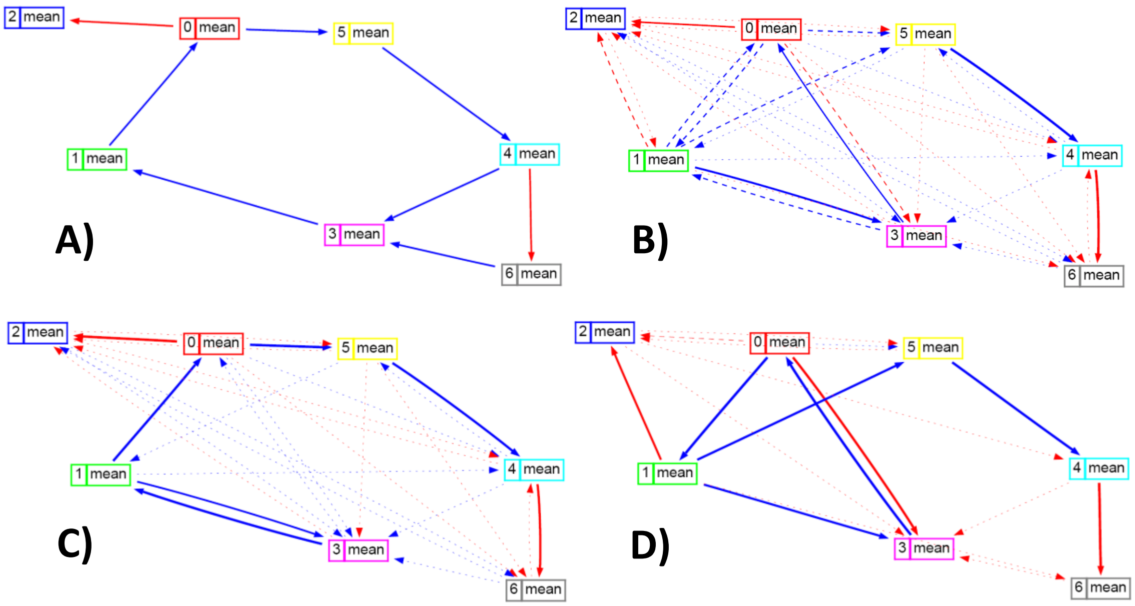

Figure 2: Using in-silico data from a reference network (A), the applied reverse-engineering algorithm created an ensemble of networks (B). Two single topology ensembles (C,D) were derived using the described module extraction approach. Both single topology ensembles explain the in-silico experimental data very well (average RMSD 0.075 and 0.081) but effector-target relations differ strongly (AUPRC to reference 0.898 and 0.311).

present in the reference and accordingly restricted the ensemble to those networks containing the tested interaction. In case of class 3 interactions, this corresponds to selecting the corresponding single topology ensemble. On average, AUPRC values increased only when testing for class 3 interactions (figure 3C).

4 Discussion

Motivation of Association Rule Learning, Pairwise Relations and Module Extraction The proposed module extraction procedure is intended to be simple, comprehensible, traceable and computationally fast. Using association rule learning to derive support and confidence for rules of the type "$[P_i, s] \Rightarrow [P_j, t]$" is straightforward and, in case of single antecedent-single consequent rules, is basically done by counting individual and joint occurrences of observed target-effector pairs. Notice that the typical number of observed target-effector pairs is much less than the number of possible pairs, thus counting joint occurrences can be done very fast.

The provided definitions of AND and XOR associations with respect to association rules are analogous to the definitions of *implication* and *exclusive or* operations in Boolean logic. If one replaces the *min* conjunction operator by *and* and ignores the *conf()* functions, the logical equivalence of left-hand and right-hand sides is obvious or can be shown easily. Using a *min* conjunction is quite common in approximate reasoning (e.g. fuzzy logic [25]). Other conjunctions like *product* or *bounded product* are possible. The interpretation of pairwise associations is straightforward. Both AND-related interactions have to be present to achieve a biologically meaningfully effect. XOR-related interactions might either represent redundancies, which are discouraged as sparseness of networks is typically demanded in reverse-engineering, or the presence of both interactions might simply contradict the reference data.

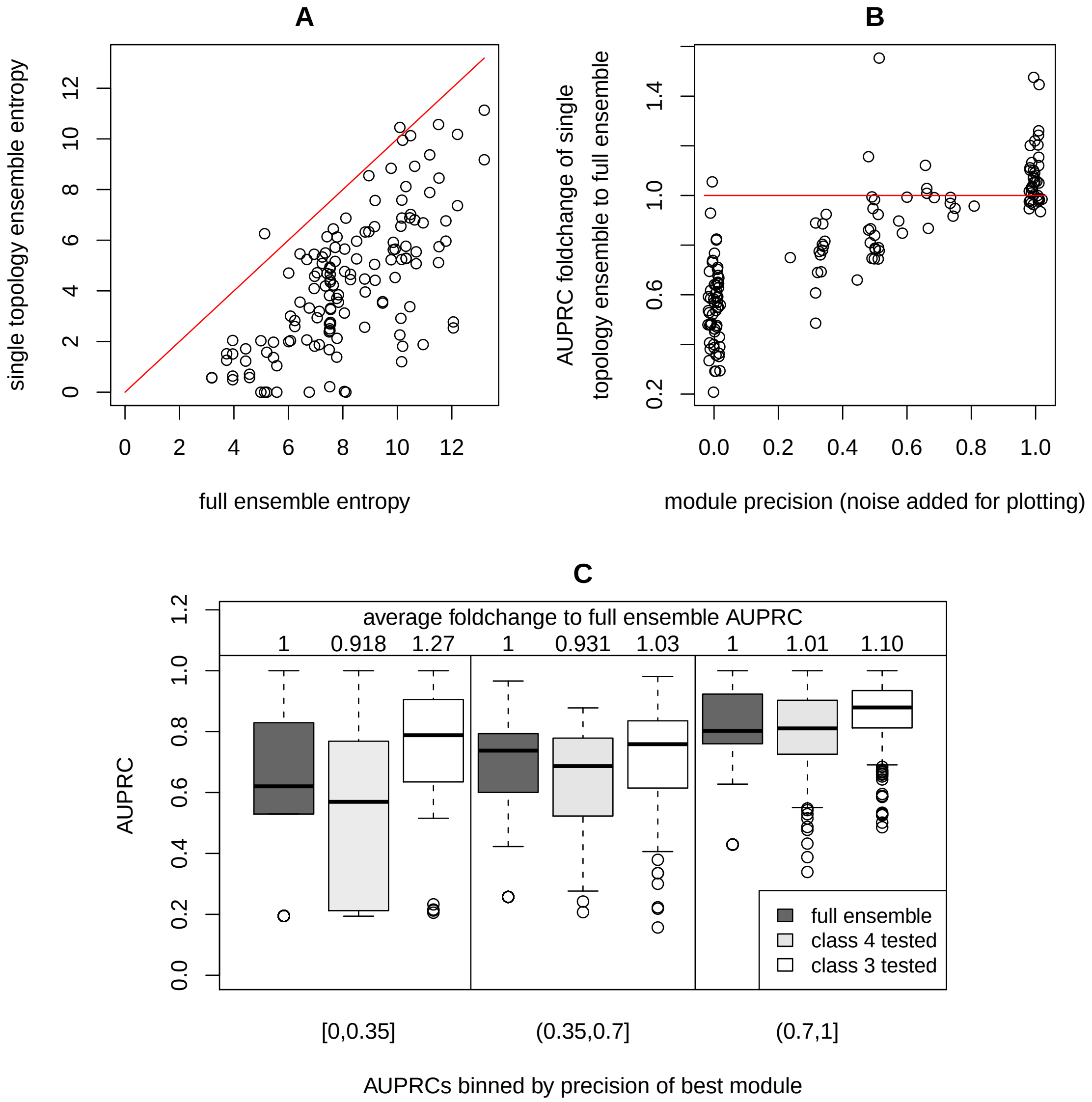

Figure 3: Evaluation results for the 7/7/4/500 experiment (table 1). (A) Entropy of single topology ensembles is decreased due to the reduced number of interactions with medium frequencies. (B) AUPRCs of single topology ensembles increase if contained modules are present in the reference. Module precision ranges between 1 (all module interactions are present in the reference) and 0 (no module interactions are present in the reference). (C) Restricting the full ensemble by testing for the presence of class 3 interactions in the reference network in general increases AUPRC, while testing for class 4 interactions has a less pronounced or even negative effect.

Influence of Cutoffs Cutoffs applied during module extraction are used to distinguish relevant observations (here, occurrence of certain interactions or modules) from spurious ones, which might be artifacts of the applied reverse-engineering approach (noise). The specific choice of cutoffs is interdependent with the number of modules hidden in the prediction ensemble as well as their frequency. Additionally, the number of distinct good-fit topologies (and therefore modules) in turn depends on references topology, available data, and used dynamical model. Thus cutoffs have to be assessed case-specific, e.g. by starting with relatively stringent cutoffs, repeatedly reducing them and assessing extracted modules by number, size and biological meaning. Future work will address an automated procedure for this task.

5 Conclusion

The presented module extraction procedure is able to detect modules in ensembles of networks and utilizes these to classify networks according to their topology. It can be applied to all ensembles of networks of directed activating or inhibiting interactions, independent of the applied probabilistic reverse-engineering algorithm. Resulting single topology ensembles as well as associations between interactions improve interpretability of prediction results and can guide experimental design to test for the proposed alternative network topologies. The procedure can be extended in many ways by including scores of predicted networks for ensemble average creation, automatic adjustment of cutoffs, fuzzy assignment of interactions to modules, fuzzy classification of networks, and using single topology ensembles as priors for repeated rounds of reverse-engineering and module detection.

References

[1] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews. Molecular Cell Biology*, 9(10):770–780, October 2008.

[2] Daniel Marbach, Robert J. Prill, Thomas Schaffter, Claudio Mattiussi, Dario Floreano, and Gustavo Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. 107(14):6286–6291, April 2010.

[3] Florian Markowetz and Rainer Spang. Inferring cellular networks - a review. *BMC Bioinformatics*, 8(Suppl 6):S5, 2007.

[4] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Mol Syst Biol*, 3, February 2007.

[5] Chao Sima, Jianping Hua, and Sungwon Jung. Inference of gene regulatory networks using Time-Series data: A survey. 10(6):416–429, September 2009.

[6] Edda Klipp, Bodil Nordlander, Roland Kruger, Peter Gennemark, and Stefan Hohmann. Integrative model of the response of yeast to osmotic shock. *Nat Biotech*, 23(8):975–982, 2005.

[7] Shenghua Li, Paul Brazhnik, Bruno Sobral, and John J Tyson. A quantitative study of the division cycle of caulobacter crescentus stalked cells. *PLoS Comput Biol*, 4(1):e9, January 2008.

[8] Moritz von Stosch, Joana Peres, Sebastiao de Azevedo, and Rui Oliveira. Modelling biochemical networks with intrinsic time delays: a hybrid semi-parametric approach. *BMC Systems Biology*, 4(1):131, 2010.

[9] Claudine Chaouiya. Petri net modelling of biological networks. *Briefings in Bioinformatics*, 8(4):210 –219, July 2007.

[10] Ming Chen and Ralf Hofestädt. Quantitative petri net model of gene regulated metabolic networks in the cell. *In Silico Biology*, 3(3):347–365, 2003.

[11] Mor Peleg, Iwei Yeh, and Russ B. Altman. Modelling biological processes using workflow and petri net models. *Bioinformatics*, 18(6):825 –837, June 2002.

[12] Graham J Hickman and T Charlie Hodgman. Inference of gene regulatory networks using boolean-network inference methods. *Journal of Bioinformatics and Computational Biology*, 7(6):1013–1029, December 2009.

[13] Haseong Kim, Jae K Lee, and Taesung Park. Boolean networks using the chi-square test for inferring large-scale gene regulatory networks. *BMC Bioinformatics*, 8:37, 2007.

[14] Shawn Martin, Zhaoduo Zhang, Anthony Martino, and Jean-Loup Faulon. Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics (Oxford, England)*, 23(7):866–874, April 2007.

[15] Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene van Someren, and Reinhard Guthke. Gene regulatory network inference: Data integration in dynamic models–A review. *Biosystems*, 96(1):86–103, April 2009.

[16] Jonathan Tomshine and Yiannis N. Kaznessis. Optimization of a stochastically simulated gene network model via simulated annealing. *Biophysical Journal*, 91(9):3196–3205, November 2006.

[17] Jun Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, January 2008.

[18] Daniel Marbach, Claudio Mattiussi, and Dario Floreano. Combining multiple results of a Reverse-Engineering algorithm: Application to the DREAM Five-Gene network challenge. *Annals of the New York Academy of Sciences*, 1158(1):102–113, March 2009.

[19] Michael Hahsler, Christian Buchta, Bettina Gruen, and Kurt Hornik. *arules: Mining Association Rules and Frequent Itemsets*, 2011. R package version 1.0-6.

[20] Michael Hahsler, Bettina Gruen, and Kurt Hornik. arules – A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, October 2005.

[21] Rakesh Agrawal, Tomasz Imieliski, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207216, June 1993. ACM ID: 170072.

[22] R Küffner, Tobias Petri, Lukas Windhager, and Ralf Zimmer. Petri Nets with Fuzzy Logic (PNFL): Reverse Engineering and Parametrization. *PLoS One*, 5:e12807, 09 2010.

[23] Lukas Windhager and Ralf Zimmer. Intuitive Modeling of Dynamic Systems with Petri Nets and Fuzzy Logic. In *German Conference on Bioinformatics*, volume P-136 of *Lecture Notes in Informatics*, pages 106–115, September 9-12, 2008, Dresden, Germany, 2008. Gesellschaft für Informatik.

[24] Lukas Windhager, Florian Erhard, and Ralf Zimmer. Fuzzy modeling. In Ina Koch, Wolfgang Reisig, and Falk Schreiber, editors, *Modeling in Systems Biology: The Petri Net Approach*. Springer, 2010.

[25] J.M. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345–377, 1995.