Robustness and accuracy of functional modules in integrated network analysis

Daniela Beisser¹, Stefan Brunkhorst¹, Thomas Danekar¹, Gunnar W. Klau²,³, Marcus T. Dittrich¹* and Tobias Müller¹*

¹Department of Bioinformatics, Biocenter, University of Würzburg, Am Hubland, 97074 Würzburg, Germany. ²Life Sciences Group, Centrum Wiskunde & Informatica (CWI), Science Park 123, 1098 XG Amsterdam, The Netherlands. ³Netherlands Institute for Systems Biology.

ABSTRACT

Motivation: High-throughput molecular data provide a wealth of information that can be integrated into network analysis. Several approaches exist that identify functional modules in the context of integrated biological networks. The objective of this study is twofold: first to assess the accuracy and variability of identified modules and second to develop an algorithm for deriving highly robust and accurate solutions.

Results: In a comparative simulation study accuracy and robustness of the proposed and established methodologies are validated, considering various sources of variation in the data. To assess this variation, we propose a jackknife resampling procedure resulting in consensus modules. In phylogeny, Felsenstein (1985) introduced consensus trees. Similarly, resampling procedures can be used to define a confidence measure for splits in a phylogenetic tree and to calculate consensus trees. Various methods have been proposed to identify functional modules in an integrated network. In this study we focus on the popular approaches proposed by Ideker et al. (2002), Ulitsky and Shamir (2007) and Dittrich et al. (2008). While these algorithms differ in many important aspects, conceptually they all aim at identifying connected subnetworks that contain significantly deregulated genes. Ideker et al. (2002) introduced the problem and proposed a simulated annealing approach to identify subnetworks. Due to the heuristic nature of such sampling approaches, the resulting modules are not optimal in general. In an alternative approach Ulitsky et al. (2010) propose the algorithm DEGAS (Dysregulated Gene set Analysis via Subnetworks), based on a greedy approximation to identify subnetworks of dysregulated genes. In contrast to the above mentioned approaches, the algorithm of Dittrich et al. (2008) identifies optimally scoring subnetworks using an exact algorithm based on integer linear programming (ILP).

Besides the accuracy of a module identification method, the robustness of obtained solutions is of particular importance. A natural question is: How variable are the provided solutions (given the method)? A highly variable method produces largely differing solutions in different runs or on slightly perturbed input data and is thus less reliable. Clearly, well designed algorithms should ideally show both: high accuracy as well as high robustness. Here we investigate the accuracy and robustness of the three prominent module detection algorithms regarding (i) the integrated gene expression data and (ii) the network structure of the PPI network itself.

As a consequence of the investigation we propose a novel method to calculate accurate as well as robust modules in which robust parts are indicated by support values, introducing the new concept of consensus modules. In phylogeny, Felsenstein (1985) introduced resampling approaches (e.g., bootstrap, jackknife) to define a confidence measure for splits in a phylogenetic tree and to calculate consensus trees. Similarly, resampling procedures can be used to assess the robustness of functional modules in integrated network analysis.

Associate Editor: Dr. Trey Ideker
analysis. We use the delete-half jackknife (Felsenstein, 2004) to re-sample the input microarray data and construct a set of resulting modules. The consensus module summarizes the obtained modules as one highly accurate and robust module with support values assigned to its nodes and edges. For this purpose we extend the existing exact approach of Dittrich et al. (2008) from a purely node-based optimization to an node- and edge-based optimization problem. While this extension might be useful in various applications in network analysis, we first use this extension to define and calculate the consensus network. The major benefit of this procedure is twofold: first we identify the optimal accurate as well as optimal robust module. Second, we analyze and visualise the inner structure of the identified module by assigning support values to both nodes and edges.

The outline of the paper is as follows: we first investigate the robustness of obtained solutions by comparing our approach to other methods in a simulation framework. Therefore, we evaluate the resulting modules in terms of accuracy and variability using integrated microarray data, perturbed integrated data and perturbed interaction networks as different sources of variability. To assess and quantify the method-independent variability of the modules (by assigning support values) we introduce a novel consensus algorithm based on a resampling procedure. Finally we apply the consensus approach to two experimental data sets: microarray profiles regarding acute lymphoblastic leukemia and diffuse large B-cell lymphoma.

2 MATERIALS AND METHODS

2.1 Gene expression and network data

PPI data from HPRD (Mishra et al., 2006) were used, constituting a network of 9386 proteins and 36504 interactions as well as a human PPI network from the meta-database PINA (Wu et al., 2009) with 11354 proteins and 68257 interactions. Expression data were taken from a study on diffuse large B-cell lymphoma from Alizadeh et al. (2000) and a subset of a leukemia microarray collection for c-ALL/Pre-B-ALL with t(9;22) and without t(9;22) translocation (ArrayExpress experiment: E-GEOD-13159(Kohlmann et al., 2008; Haferlach et al., 2010). The DLBCL dataset contains 194 samples on custom microarrays, containing probes for 3583 genes. Mapping these to the PINA network resulted in a largest connected component of 2220 genes and 12074 interactions. The ALL data set contains 359 samples on Affymetrix hgu133plus2 gene chips with 54675 probes corresponding to 19738 genes.

2.2 Integrated network analysis

Integration of gene expression data and the search for optimal modules has been performed as described in (Dittrich et al., 2008) with an algorithm termed heinz (heaviest induced subgraph). Briefly, the distribution of raw p-values from a standard t-test, conducted on the microarray data, can be considered as a mixture of signal and noise, where the signal component is modelled to be Beta(a, 1) distributed (Pounds and Morris, 2003), whereas the distribution of the noise component is by definition given as the uniform distribution. By fitting a Beta-uniform mixture (BUM) model, maximum-likelihood estimates for all model parameters can be obtained that are subsequently used to score the network nodes. The node score is given by the likelihood ratio of the signal to the noise component and can be adjusted by a threshold \( \tau \) depending on a pre-selected false discovery rate (FDR).

\[
S_{FDR}(x) = \log \left( \frac{ae^{a-1}}{a^x} \right) = (a-1) \left( \log(x) - \log(\tau(FDR)) \right).
\]

2.3 Extensions of the heinz algorithm

Based on the node score defined in Sect. 2.2, we have proposed heinz (Dittrich et al., 2008), a method to identify functional modules by finding maximum-scoring connected subnetworks. In contrast to prevalent heuristic methods, heinz is an exact approach, i.e., it finds provably optimal and suboptimal solutions. The method exploits the close connection of maximum-scoring connected subnetworks and prize-collecting Steiner trees (PCST). In fact, we use an integer linear programming-based approach for the Steiner tree problem after an initial problem transformation. Here we have extended the heinz method to allow for the incorporation of (i) edge weights and (ii) computing modules of a predefined size.

(i) In Dittrich et al. (2008), we defined modules as optimal solutions to the following problem: Given an undirected, vertex-weighted graph \( G = (V, E, w) \) with weights \( w : V \rightarrow \mathbb{R} \), find a connected subgraph \( T = (V_T, E_T) \) of \( G, V_T \subset V, E_T \subset E \) that maximizes \( score(T) = \sum_{v \in V_T} w(v) \). We have shown that an optimal module can always be represented by a tree in case the edge scores are neglected. We now extend our formulation to incorporate edge scores in the following way: Given an undirected, vertex- and edge-weighted graph \( G = (V, E, w) \) with weights \( w : V \cup E \rightarrow \mathbb{R} \), find a subtree \( T = (V_T, E_T) \) of \( G, V_T \subset V, E_T \subset E \) that maximizes \( score(T) = \sum_{v \in V_T} w(v) + \sum_{e \in E_T} w(e) \). We can show a similar transformation to the PCST problem as in the original algorithms that allows only node weights.

(ii) It is easy to change our method such that it finds the optimal-scoring module of a fixed, predefined size \( k \). In our integer linear program, binary variables \( x_v \) determine the presence of nodes in the optimal subgraph \( T \), that is, \( x_v = 1 \) if \( v \in V_T \) and \( x_v = 0 \) otherwise. Just adding the constraint \( \sum_{v \in V_T} x_v = k \) limits the search space to contain only modules of size \( k \).

The heinz algorithm can be accessed from the open-source R package BioNet (Beisser et al., 2010), available from http://bioapps.biozentrum.uni-wuerzburg.de and the Bioconductor project. The package includes the integration of data, scoring of nodes and alternative methods for network search and visualization. The methods for the calculation of consensus modules are integrated in the BioNet package.

2.4 GO enrichment

For functional characterization of genes contained in the identified modules a gene ontology (GO) (Ashburner et al., 2000) term enrichment against the complete network was performed. This identifies the GO categories that are significantly overrepresented in a set of genes. The analysis was conducted using the R package GOstats (Falcon and Gentleman, 2007).

2.5 Simulation of reference modules

To evaluate the performance of the proposed algorithm and the improvement over other methods, a simulation framework has been created on the basis of the input microarray data. For this we use an induced PPI network from HPRD (human protein reference database) contained in the BioNet package. To compare the resulting modules to the true solution, a reference module \( S = (W, F) \), \( W \subseteq V \) and \( F \subseteq E \), of size \( k \) as a subgraph of graph \( G = (V, E) \) is created as follows:

1. Start with a given graph \( G = (V, E) \) and an empty subgraph \( S \)
2. Select random seed node \( v \in V \) and include node in \( W \)
3. Expand \( S \) by adding a node \( u \) and its induced edges from the neighborhood \( k(u) := \{ u \in V | (v, u) \in E, w \in W, u \in V \setminus W \} \), for which its average shortest path length \( s_p(u) \) within \( S \) is most similar to its average shortest path length \( s_p(u) \) within the full network.
4. Repeat step 3 until given size is reached, that is, \( |W| = k \).
The average shortest path length was chosen as a characteristic network measure, which remains approximately constant to the average shortest path length of the network for all extracted modules in real data sets. Modules with longer average shortest paths correspond to sparse subnetworks that are frequently not biologically relevant (Ulitsky et al., 2010).

The subnetwork is termed signal module in the following. Signal modules of varying sizes $k$ are generated. For the genes contained in this module expression values are simulated showing differential expression between two groups signal strength (see Suppl. Section 2). Subsequently, the simulated gene expression data are analyzed as detailed in Section 2.2.

2.6 Resampling procedure

The statistical method of jackknifing was first introduced by Quenouille (1956) and Tukey (1958) by deleting one observation to estimate the bias and variance of a statistic of interest. The more general delete-$j$ observations jackknife draws random subsets of the data without replacement by deleting $j$ observations. The delete-half jackknife has similar properties as another resampling method, the bootstrap (Felsenstein, 2004), and can be seen as an approximation of it (Efron, 1979). The difference between these resampling approaches is that the bootstrap is a random resampling procedure with replacement, and the jackknife draws random subsets of the data without replacement by deleting $j$ observations.

Often one is interested in the standard error or the confidence interval of a statistic estimator $T$ for a parameter of interest that is given as function $T$ of the data points $x_1, x_2, \ldots, x_n$.

$$T = T(x_1, x_2, \ldots, x_n).$$

Drawing $J$ times randomly a subset of $n-j$ values from the observed data $x_1, x_2, \ldots, x_n$, we obtain $J$ jackknife pseudo-replicates of $n-j$ data points. For each sample the estimates

$$T_i = T(x_1, x_2, \ldots, x_{n-j}), \quad i = 1, \ldots, J,$$

are calculated. Based on this jackknifed distribution of the estimator the standard error and confidence intervals can be estimated. A 50% jackknife was used and half of the observations dropped as recommended by Felsenstein (1985, 2004).

2.7 Perturbation of the network

The stability and variance of modules calculated on an integrated protein-protein interaction network are studied by investigating simulated data sets with respect to three types of perturbations of the network. The perturbations considered are random deletion, addition and rewiring of 10%, 25% and 50% of all edges in the network. The method and analysis is described in more detail in the Suppl. Section 3. In addition we analyze whether networks with different numbers of interactions, e.g., HPRD and PINA, alter the results obtained from real biological networks.

2.8 Comparisons to other methods

To assess the performance of the algorithms, we generated artificial expression data with signal modules as detailed in Section 2.5 for varying module sizes. The DEGAS algorithm (Ulitsky et al., 2008, 2010) implemented in the program Matisse, identifies minimal connected subnetworks in a PPI network in which the number of dysregulated genes from expression profiles exceeds a certain threshold. The tool jActiveModules (Ideler et al., 2002) is another heuristic approach to identify high-scoring subnetworks based on expression p-values by transforming p-values into scores and assigning each protein of a PPI network a score. The DEGAS algorithm is applied throughout the study with the following parameters: UP regulated, dysregulation ratio=1, number of outlier cases=1 (parameter $l$), heuristic=ExpandingGreedy, k-steps=1, parameter $k$ (number of significant genes per case) is varied in steps of 10 from minimum tested $k = 1$ and maximum tested $k = 10$ to minimum tested $k = n-9$ and maximum tested $k = n$ to obtain modules of varying sizes (for simulated modules of size 25 and 50: $n = 70$, for module of size 150: $n = 100$). For jActiveModules the number of modules is set to 1 and it is run iteratively on the previous solution until the smallest possible module size is reached.

The accurate identification of simulated modules is evaluated in a precision-recall (PR) curve, where the precision quantifies the number of correctly identified nodes among the ones identified as positives (TP/(TP+FP)) and the recall measures the fraction of correctly identified nodes among all correctly classified nodes (TP/(TP+FN)). A compact representation of the precision-recall curve is given by the $F_1$ score, which is the maximum over $F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$. The variability of obtained solutions is assessed by calculating the pairwise Jaccard coefficient $J_i = \frac{A_i \cap B_i}{A_i \cup B_i}$, where $A_i$, $B_i$ are the vertex sets of the modules A and B. Furthermore, the accuracy and variance of the solutions on perturbed data and on perturbed networks are assessed (for details see Suppl. Section 3). The algorithm is run on a network with perturbed gene expression data; or on a perturbed network with the original gene expression data.
3 RESULTS

3.1 The consensus algorithm

Here we propose a jackknife procedure to assess the robustness and variability of the network modules as depicted in Fig. 1. Briefly, the algorithm for the calculation of the consensus modules consists of the following steps:

1. Resampling of microarray data using jackknife for \( J \) pseudo-replicates (Fig. 1, first line)
2. Scoring network nodes and calculating the maximum-scoring subnetwork (MSS) of the size of the original module for each jackknife-replicate (Fig. 1, middle part)
3. Calculating the frequency of nodes and edges in the resulting jacknifed modules
4. Rescoring the network with a consensus score derived from the frequency of edges and nodes (Fig. 1, lower part)
5. Calculating the MSS of equal size to the original module. This constitutes the consensus module

In more detail, the algorithm starts with the generation of \( J \) jackknife samples of the expression data \((J = 100)\) throughout the study, for comparisons consensus modules with \( J = 1000 \) were computed, yielding very similar modules in the biological examples with a highly significant correlation \((r > 0.98)\) for node and edge support values. For each jackknife pseudo-replicate a node score is calculated in the same manner as for the original data as detailed in Section 2.2. Subsequent module searches results in a set of slightly differing modules \( G_i = (V_i, E_i) \), \( 1 \leq i \leq J \) for each pseudo-replicate. The frequency of each gene in the resulting \( J \) modules is used to define a consensus score for each node and each edge

\[
S^p_{\scriptscriptstyle \rho}(v) = \left( \sum_{i=1}^{J} \{v \cap V_i \} \right) - \rho, \quad S^p_{\scriptscriptstyle \rho}(e) = \left( \sum_{i=1}^{J} \{e \cap E_i \} \right) - \rho,
\]

for a given threshold \( \rho \in [0, J] \) (\( \rho = J/2 \) throughout the study). This means in particular, that only nodes or edges occurring more than \( \rho \) times in the set of resampled modules get a positive score. Support values are calculated from the resampling procedure for the nodes and edges of the network. On the one hand these can be used for annotation of the obtained original module. Alternatively, the support values can be used to derive a new score for the network and calculate a novel module, the consensus module. The original network is subsequently rescoring with the consensus score for the nodes and edges and the MSS is calculated with the size set to the size of the original module. We define this resultant optimal scoring subnetwork as the consensus module. This approach extends the methodology described in (Dittrich et al., 2008) by additionally optimizing over different module topologies, resulting in a highly accurate and robust module with optimal support values.

The frequencies of the nodes and edges in the jackknifed modules are used as support values in the consensus module. These scores are visualized in the plot of the modules via the node sizes and edge widths. The more often an edge or a node occurs in any of the perturbed modules, the more likely it is, that it is a robust part of the functional module and should be considered for further analysis.

3.2 Assessing the variance of resulting modules in a simulation study

Since our objective is not only to find a module which obtains a good accuracy, but also yields results that are robust to minor changes in the underlying data, we assessed the robustness and variance of obtained solutions in a simulation study. The analysis was performed on simulated perturbed data generated with jackknifing as described in Sects. 2.5 and 3.1. To compare our method to other approaches, we used the exact algorithm on which the consensus approach is based (Dittrich et al., 2008), the DEGAS method of the program Matisse (Module Analysis via Topology of Interactions and Similarity SEts) (Ulitsky and Shamir, 2007) and the module finding plug-in jActiveModules (Ideker et al., 2002) for Cytoscape. The comparisons were performed on a 50 node signal module with a signal strength of \( 1 \) (difference in means) between the two conditions in the microarray data. A 50% jackknife was used to generate 20 data sets of perturbed microarray data as input for all three methods. Further analyses with different simulated module sizes (25 and 150) are included in the Supplementary material (see SFig. 2.3).

The same PPI subnetwork derived from HPRD, consisting of the genes from the microarray was used for all algorithms. Different module sizes were obtained from the programs by either changing size parameters or iteratively applying the method on the resulting subnetwork. This allowed us to assess the performance and variability in precision-recall curves, \( F_1 \)-measure and Jaccard coefficient. The resulting PR-curves of the 20 resampled data sets for varying module sizes are depicted in Fig. 2A for jActiveModules, Matisse and heinz, respectively. A fitted lowess regression for all 20 PR-curves is depicted, for detailed plots see SFig 1. Fig. 2B shows the maxima of the \( F_1 \)-measure (\( F_{\scriptscriptstyle \text{max}} \)) for the 20 resamples of each method. Apparently, the heinz modules obtain the highest \( F_{\scriptscriptstyle \text{max}} \) values as well as the smallest variance in \( F_{\scriptscriptstyle \text{max}} \). The difference in means of the obtained \( F_{\scriptscriptstyle \text{max}} \) for the three methods is highly significant (Wilcoxon test) as well as the difference in variance between jActiveModules and heinz (p-value of 0.007 in the Brown-Forsythe version of the Levene-type test for equal variance (Levene, 1960; Brown and Forsythe, 1974)). We chose the 20 best solutions of each algorithm in terms of precision and recall to see how they perform regarding the variance (SFiG. 4A-C). Fig. 2C depicts a histogram of how often a node is found in a module. Methods with many stable nodes (i.e., that occur in almost all modules) and few unstable nodes (those that occur in almost no module) had robust solutions with a low variance, whereas methods with opposite characteristics were non-robust and gave very different solutions for each resampled data set. Here the most robust method was again heinz, with few nodes appearing only in few modules and most nodes appearing in all modules. Furthermore, the number of correctly assigned nodes was highest for our algorithm. The pairwise Jaccard coefficients of the 20 best solutions (190 comparisons for each method, Fig. 2D and SFiG. 4) showed the variability in the resulting modules. Whereas generally large Jaccard coefficients illustrated a high similarity between all 20 modules. The differences in means between the Jaccard coefficients of heinz and Matisse and jActiveModules were highly significant (two sample Wilcoxon test, p-value of \( 5 \cdot 10^{-64} \) and \( 3 \cdot 10^{-64} \)).

The analyses with different simulated module sizes indicate that accuracy and variability of the exact approach (heinz) is independent of signal size (module size) in contrast to the heuristic methods.
We applied the proposed algorithm to the ALL data set (Kohlmann et al., 2010). In particular we investigated the differential expression between pre B-cells with and without the t(9;22) translocation, also known as Philadelphia translocation. Using an FDR of 0.01 we calculated the consensus module from 100 (and 1000) jackknife resamples of the microarray data (Fig. 4). Support values were determined for the nodes and edges contained in the module, i.e., edges with high jackknife support values represent interactions between genes/proteins that appeared often together in resampled subnetworks. 27% of the genes from the original module were obviously an unrobust signal and appeared too infrequently in the jackknifed modules to be contained in the consensus. The high variability in the data is also reflected by the low jackknife support values of the consensus module. The consensus module, particularly the robustly connected component with jackknife support of the edges and nodes greater than 25% (Fig. 4 B, highlighted in yellow) contains essential genes for the analyzed cytogenetic translocation. Among these are prominently the genes BCR and ABL1, which form a fusion transcript due to the translocation and constitutively activate downstream signaling. Thereby inhibiting apoptosis through activation of a Ras-dependent signaling pathway (Cortez et al., 1996), including the involvement of RRAS, SOS1, GRB2, RHOA and TP53. Further essential associations to SCR were shown (Deininger, 2004) as well as to insulin-signaling pathways, including the proteins IGF1R, IRS1, PI3K and GRB2 (Traina et al., 2003).

GO term enrichment analysis was performed on the resulting modules (see STable 1-4) and the robust component of the consensus module. The enriched biological processes of the consensus module include several intracellular signaling cascades, among them the above mentioned connections to Ras protein and insulin signaling pathways. The robust component alone hints to DNA damage response and signal transduction resulting in induction of apoptosis and cell communication. Functional modules obtained with jActiveModules and Matisse are shown in the Supplementary material (SFig. 7). They lack important proteins deregulated in the disease, e.g., the most prominent: BCR and ABL1. Analogously we applied the proposed algorithm to the DLBCL data set (Rosenwald et al., 2002). In DLBCL we searched for modules which are differentially expressed between the two tumor subgroups, germinal center B-cell-like (GCB) DLBCL and activated B-cell-like (ABC) DLBCL. With an FDR of $10^{-7}$ we calculated...
Recall

2010), represented by the genes LCK, LYN, BLNK, BCL2 and BCL6. It was shown for the more aggressive ABC subtype, that ABC cell lines have a constitutive expression of STAT3 and activation of NFκB (Gupta et al., 2011; Davis et al., 2001). Acetylation of STAT3 by histone deacetylases (HDACs) was shown to be responsible for the activation (Gupta et al., 2011). The genes STAT3, HDAC1 and HDAC3 are part of the consensus module. Activation of NFκB induces the expression IRF4, which in turn inhibits BCL6 gene expression, and regulates the expression of further NFκB target genes, such as CCND2, (CFLAR), BCL2 (Davis et al., 2001) and PIM1. GO analysis (see STable 5-8) resulted in very similar rather unspecific biological processes for the optimal and the consensus module, due to the high similarity of these two modules. The stronger signal in the gene expression data and less noise also result in more similar modules from jActiveModules and Matisse which also include some of the above mentioned genes (see SFig. 8).

3.4 Run time

We use an integer linear programming-based approach for the prize-collecting Steiner tree problem after an initial problem transformation to calculate maximum-scoring subnetworks. In general we made the experience, that despite the NP-hardness of the problem the algorithm runs very fast on biologically relevant instance size, usually ranging between seconds to minutes for one calculation. The proposed jackknife resampling algorithm scales linearly in the number of replicates, which is reflected in its runtime behavior (see Table 1). The highly significant correlation between modules calculated on 100 and 1000 jackknife replicates shows that the smaller number of replicates are sufficient for an accurate modularization of the microarray data. Furthermore, the computation can easily be run in parallel on a cluster or multicore workstation.

4 DISCUSSION

Here we have presented a novel method for the identification of highly robust and accurate modules. We suggest a consensus approach, based on jackknifing, to calculate a resulting functional module, whose inner structure is characterized and highlighted by support values on nodes and edges. In an extensive simulation study we compare our approach to well-established heuristic module identification methods in terms of accuracy and robustness. Particularly, we distinguish between different sources of noise that affect the obtained solutions: (i) the variability of the integrated data (e.g., gene expression data); (ii) the variability of the underlying network (e.g., PPI network) and (iii) the intrinsic methodological variability for heuristic module identification methods. In general, the exact algorithm clearly outperforms the other validated heuristics not only in terms of accuracy but also in terms of robustness. In particular, the
The simulation of perturbed networks reveals that the performance of heuristic module detection methods declines faster and more pronounced with increasing level of perturbations in contrast to the exact approach. Interestingly, our simulation results indicate that even the inclusion of a large number of false positive edges has a much stronger effect. This holds true for all algorithms examined in this study. These simulation results have also implications for the analysis of real protein-protein interaction data: In the context of integrated network analysis low confidence interaction networks (e.g., in the STRING database (Szklarczyk et al., 2011)) could perform similarly good or better than high confidence protein-protein interaction networks based on high quality threshold which may lack a large number of true positive edges. Analyses on the PINA and HPRD network, comprising large differences in the number of interactions, show that on biological networks the results are almost identical.

On the algorithmic side we have extended an existing exact approach (Dittrich et al., 2008) in two directions: (i) by the incorporation of edge scores and (ii) by the calculation of optimal modules of a given size. On the biological side we have applied both algorithms, the original exact method as well as the new proposed consensus method, to two well-known microarray data sets (ALL, DLBCL). These data sets differ in their signal content, which is directly reflected by the support values of the consensus module. In the case of a high signal content, the results for the original module and the consensus module agree in most parts, while for weak signals they differ greatly from each other. In the latter case the consensus module is a clear improvement as it represents the optimal, robust solution and depicts substructures of high confidence. The results of our study underlined the importance to distinguish robust signals from noise by the use of resampling methods as implemented in the proposed consensus approach, which inherits the accuracy from the optimal algorithm while on the other side improving its robustness.

ACKNOWLEDGEMENT

Funding: DB thanks the BMBF (Funcrypta) and DFG (Da208/10-1) for support.

REFERENCES


Figure 4: Modules calculated for the ALL microarray data set. A: the original heinz solution was calculated with an FDR of 0.01. Blue-framed nodes emphasize nodes with negative scores in the original network, all other nodes have positive scores. Nodes only present in either the original module or the corresponding consensus module are depicted by squared node symbols. Coloring of the nodes represents differential expression of the genes (red: up-regulated in samples with the BCR/ABL translocation, green: down-regulated). B: consensus module where the sizes of the nodes and width of the edges and edge labels indicate node and edge jackknife support values. Highlighted in yellow is the largest robust submodule (support values ≥ 25). The most robust central nodes in the yellow shaded submodule are the genes ABL1 and BCR, which are directly affected by the translocation t(9;22). These central genes are not present in both modules found by the heuristic methods jActiveModules and Matisse (see Supplement SFig. 7).
Figure 5: Modules calculated for the DLBCL microarray data set. A: the original exact *heinz* calculation was calculated with an FDR of $10^{-7}$. Blue-framed nodes emphasize nodes with negative scores in the original network; all other nodes have positive scores. Nodes only present in either the original module or the corresponding consensus module are depicted by squared node symbols. Colouring of the nodes represents differential expression of the genes (red: upregulated in ABC, green: downregulated in ABC). B: consensus module. Node and edge jackknife support values are indicated by the sizes of the nodes and width of the edges and edge labels. Highlighted in yellow are regions with high robustness (support values $\geq 50$).


