BIOINFORMATICS ORIGINAL PAPER Vol. 28 no. 14 2012, pages 1887–1894
doi:10.1093/bioinformatics/bts265

Systems biology Advance Access publication May 11, 2012

Robustness and accuracy of functional modules in integrated network analysis

Daniela Beisser[1], Stefan Brunkhorst[1], Thomas Dandekar[1], Gunnar W. Klau[2,3], Marcus T. Dittrich[1,*] and Tobias Müller[1,*]

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

Associate Editor: Trey Ideker

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 2-fold: 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 an ensemble of optimal modules. A consensus approach summarizes the ensemble into one final module containing maximally robust nodes and edges. The resulting consensus module identifies and visualizes robust and variable regions by assigning support values to nodes and edges. Finally, the proposed approach is exemplified on two large gene expression studies: diffuse large B-cell lymphoma and acute lymphoblastic leukemia.

Contact: marcus.dittrich@biozentrum.uni-wuerzburg.de or tobias.mueller@biozentrum.uni-wuerzburg.de

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

Received on November 11, 2011; revised on March 21, 2012; accepted on May 1, 2012

1 INTRODUCTION

Multiple genome-scale datasets nowadays allow to model the cell as an intricate network of molecular interactions. Research in systems biology has changed accordingly, now focusing on network analysis of high-throughput genome-, transcriptome- and proteome data.

Reaching beyond the analysis of mere topological questions, integrated network analysis incorporates additional molecular data into a network. For gene expression data integrated approaches are used to search for pathways, functional modules or gene signatures containing differentially expressed genes in the context of gene networks or protein–protein interaction (PPI) networks (Dittrich *et al.*, 2008; Ideker *et al.*, 2002; Scott *et al.*, 2006; Ulitsky and Shamir, 2007). Given the integrated gene expression data, the objective is to find the maximal significantly deregulated (i.e. differentially expressed) set of interconnected genes in the cellular network. We refer to the resulting connected subnetwork as a functional module, which is also denoted as active or perturbed module (Ideker *et al.*, 2002). Please note that this is in contrast to other fields in biology (e.g. proteomics), where functional modules denote protein complexes (Pu *et al.*, 2007).

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). Although 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 and 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. We use the delete-half jackknife (Felsenstein, 2004) to

*To whom correspondence should be addressed.

resample 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 a node- and edge-based optimization problem. Although 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 2-fold: first, we identify the optimal accurate as well as optimal robust module. Second, we analyze and visualize the inner structure of the identified module by assigning support values to both nodes and edges.

The outline of the article 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 datasets: microarray profiles regarding acute lymphoblastic leukemia (ALL) and diffuse large B-cell lymphoma (DLBCL).

2 MATERIALS AND METHODS

2.1 Gene expression and network data

PPI data from HPRD human protein reference database; (Mishra *et al.*, 2006) were used, constituting a network of 9386 proteins and 36 504 interactions as well as a human PPI network from the meta-database PINA (Wu *et al.*, 2009) with 11 354 proteins and 68 257 interactions. Expression data were taken from a study on DLBCL 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)(Haferlach *et al.*, 2010; Kohlmann *et al.*, 2008). The DLBCL data comprise 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 12 074 interactions. The ALL dataset contains 359 samples on Affymetrix hgu133plus2 gene chips with 54 675 probesets corresponding to 19 738 genes with gene symbol and Entrez ID. Of these, 10 576 can be mapped to the human PINA network, resulting in a largest connected component of 10 576 genes and 63 015 interactions.

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 modeled 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 τ depending on a pre-selected false discovery rate (FDR).

$$S^{\mathrm{FDR}}(x) = \log\left(\frac{ax^{a-1}}{a\tau^{a-1}}\right) = (a-1)\big(\log(x) - \log(\tau(\mathrm{FDR}))\big).$$

2.3 Extensions of the `heinz` algorithm

Based on the node score defined in Section 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 (PCSTs). In fact, we use an ILP 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 $w: V \to \mathbb{R}$, find a connected subgraph $T=(V_T,E_T)$ of G, $V_T \in V$, $E_T \in E$ 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 $w: V \cup E \to \mathbb{R}$, find a subtree $T=(V_T,E_T)$ of G, $V_T \in V$, $E_T \in E$ that maximizes score$(T)=\sum_{v\in V_T} w(v)+\sum_{e\in E_T} w(v)$. 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 ILP, 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 $\sum_{v\in V} x_v = k$ 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://bionet.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 contained in the BioNet package with 2034 genes which are existent on the DLBCL microarray. 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 $\delta(S) := \{u \in V \mid (w,u) \in E, w \in W, u \in V \setminus W\}$, for which its average shortest path length $l_S(u)$ within S is most similar to its average shortest path length $l_G(u)$ 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 datasets. 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 (see Supplementary Section S2). 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 (CI) of statistical estimator $\hat{t}$ for a parameter of interest that is given as function T of the data points $x_1, x_2, \ldots, x_n$

$$\hat{t} = T(x_1, x_2, \ldots, x_n).$$

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

$$\hat{t}^{(i)} = T\left(x_1^{(i)}, x_2^{(i)}, \ldots, x_{n-j}^{(i)}\right), \quad i = 1, \ldots, J,$$

are calculated. Based on this jackknifed distribution of the estimator the standard error and CIs 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 datasets 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 Supplementary Section S3.2. 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 (Ideker *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.

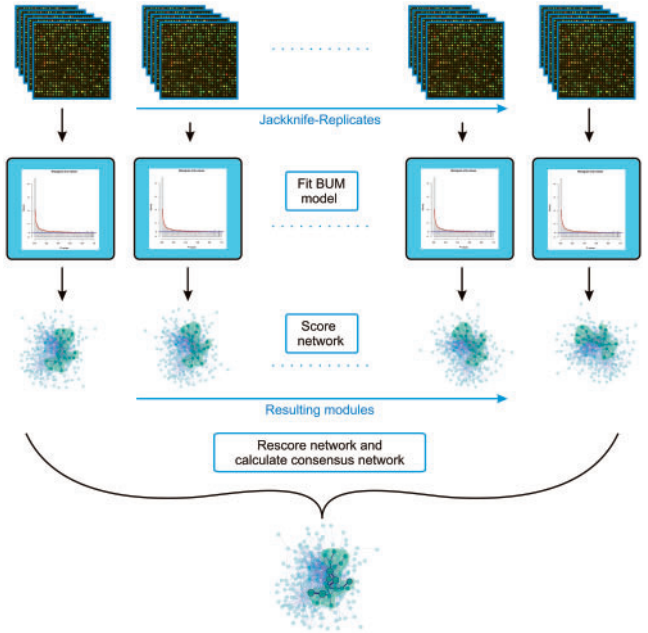

Fig. 1. Outline of the consensus algorithm. First, the microarray data are resampled, i.e. for each gene the expression values are resampled using jackknife. Differential expression is tested on all jackknife-replicates. A BUM model is fitted to the resulting p-value distribution. Based on the estimated parameters of the distribution, scores are calculated for all nodes of the network. For each scored network the maximum-scoring subnetwork (MSS) is calculated. Each of the resulting modules differs slightly due to the resampling, this set of modules is subsequently used as a basis for deriving consensus node and edge scores. Then, the original network is rescored with the consensus scores and the MSS is calculated. The resulting consensus module collects as many highly supported nodes and edges as possible (see Section 3.1) and thus constitutes the maximal robust and accurate module

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 PR curve is given by the $F_{\max}$ score, which is the maximum over $F_1 = 2 \cdot \text{precision} \cdot \text{recall} / \text{precision} + \text{recall}$. The variability of obtained solutions is assessed by calculating the pairwise Jaccard coefficient $J_v = A_v \cap B_v / A_v \cup B_v$, where A_v, B_v 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 Supplementary Section S3). 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 Figure 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 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 jackknifed 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

$$S_J^\rho(v)=\left(\sum_{i=1}^{J}|\{v\}\cap V_i|\right)-\rho,\quad S_J^\rho(e)=\left(\sum_{i=1}^{J}|\{e\}\cap E_i|\right)-\rho,$$

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 ρ 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 rescored 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 Sections. 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 datasets 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 (Supplementary Figs. S2 and S3). 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 PR curves, F_1-measure and Jaccard coefficient. The resulting PR-curves of the 20 resampled datasets for varying module sizes are depicted in Figure 2A for jActiveModules, Matisse and `heinz`, respectively. A fitted lowess regression for all 20 PR-curves is depicted, for detailed plots see Supplementary Figure S1. Figure 2B shows the maxima of the F_1-measure ($F_{\max}$) for the 20 resamples of each method. Apparently, the `heinz` modules obtain the highest $F_{\max}$ values as well as the smallest variance in $F_{\max}$. The difference in means of the obtained $F_{\max}$ 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 (Fig. S4A–C). Figure 2C depicts a histogram of how often a node is found in a module. Methods with many *stable* nodes (i.e. those 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 dataset. 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 Fig. S4) 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 `heinz` and jActiveModules were highly significant (two sample Wilcoxon test, p-value of $5\cdot10^{-64}$ and $3\cdot10^{-64}$).

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. Here in particular for small signal sizes the heuristic approaches show a high variability and a high drop in accuracy (see Fig. S1–3).

In addition to the noise arising from the integrated gene expression data, possibly false positive and false negative interactions of the network must also be taken into account. Therefore we perturbed the interaction networks as described in Section 2.7 and assessed the accuracy and variance similarly to the previous analysis. The effect of deleting, adding and rewiring random edges from the network is shown in Fig. S5A–C, by plotting precision versus recall of the obtained functional modules. Figure 3 shows the effect of the random rewiring of edges on the performance of the

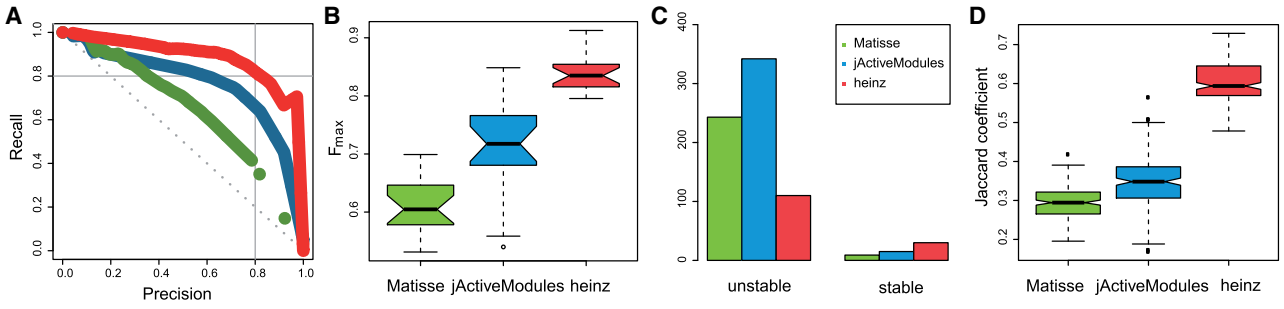

Fig. 2. Comparative analysis of module detection methods. On 20 simulated resamples of microarray data, with a module size of 50 nodes, the methods Matisse, jActiveModules and `heinz` have been applied to detect functional modules (Supplementary Fig. S1A–C). The accuracy is analyzed by precision-recall plots by varying the size of the resulting module over size parameter settings (Matisse, `heinz`) or iterative runs (jActiveModules). (**A**) PR curves for all three methods (for more details see Supplementary Fig S1). (**B**) $F_{\max}$ of the three methods for the 20 resamples. (**C**) Frequencies of the stable and unstable nodes in the 20 best modules resulting from each method (Supplementary Fig. S4). *Unstable* means that a node is found only in one or two modules. Stable means that a node is found in 19 or 20 modules. (**D**) Pairwise Jaccard coefficients of the 20 best solutions (190 comparisons for each method). Simulation results reveal the superior performance of the exact algorithm (`heinz`) in terms of accuracy (pr-curves, $F_{\max}$) and robustness (Jaccard coefficient). Similar results were obtained by smaller and larger simulated module sizes (Supplementary Figs. S1–3)

different algorithms. With varying degree of perturbation, from 10% to 50%, the performance of the methods decreases. Again, `heinz` shows most robust performance when changing the underlying network, even with 50% of the edges rewired. The other methods are more sensitive to destruction of the network and show a decreasing performance upon rewiring of edges. We also looked at the consistency of the solutions in Figure 3B by calculating the Jaccard coefficient of five independent solutions for each level of perturbation and each method. The results show that `heinz` is very robust in identifying the same solutions on perturbed networks.

3.3 Case studies on biological expression datasets

We applied the proposed algorithm to the ALL dataset (Haferlach *et al.*, 2010; Kohlmann *et al.*, 2008). 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. 4B, 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 (Table S1–S4) 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 (Fig. S7). 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 dataset (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 the consensus module based on 100 (and 1000) jackknife resamples (Fig. 5).

In contrast to the ALL dataset, nodes and interactions of the resulting DLBCL consensus module were much more robust, as indicated by the higher support values. Only 6% of the genes of the original module were not included in the consensus module. Therefore when using the edges and nodes with at least 50% support values, a large part of the consensus module was selected as robustly connected component (Fig. 5B, highlighted in yellow). In particular the robust component comprises genes with associations to known deregulated processes in DLBCL. The ABC subtype of DLBCL requires a chronic active B-cell signaling for cell survival (Davis *et al.*, 2010), represented by the genes LCK, LYN, BLNK, BCL2 and BCL6. It was shown for the more aggressive ABC DLBCL subtype, that ABC cell lines have a constitutive expression of STAT3 and activation of NFκB (Davis *et al.*, 2001; Gupta *et al.*, 2011). 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 (Table S5–S8) resulted in very

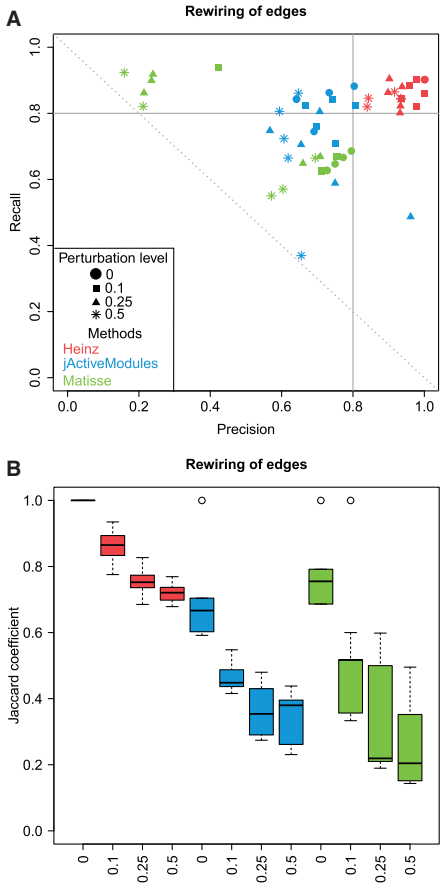

Fig. 3. Simulated networks were perturbed by randomly rewiring 0, 10, 25 and 50% of all edges. (**A**) Recall-precision plot of the modules after perturbations for the methods `heinz` (red), jActiveModules (blue) and Matisse (green). Perturbations were applied five times to obtain five independent solutions for each perturbation level, which is indicated by different symbols. (**B**) Pairwise Jaccard coefficients of the modules (10 comparisons for each method) as a measure of variability. The exact `heinz` method outperforms Matisse and Ideker in terms of accuracy and variability on various disturbed network topologies

Table 1. Elapsed time for the calculation of 1, 100 and 1000 instances of the ILP algorithm for the biological datasets

Dataset	1 instance on workstation	100 instances on cluster	1000 instances on cluster
DLBCL	0 min 9 s	0 min 34 s	6 min 30 s
ALL	10 min 0 s	98 min 47 s	349 min 50 s

Workstation: on single core, Intel Core i7 CPU, 3.4 GHz, 8 GB RAM, Cluster: 40 node, quad-core cluster, Xeon 5140 CPUs, 2.33 GHz, 8 GB RAM. DLBCL subnetwork: 2220 nodes, 12 074 edges; ALL subnetwork: 10 576 nodes, 63 015 interactions.

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 (Fig. S8).

3.4 Run time

We use an ILP based approach for the PCST problem after an initial problem transformation to calculate MSSs. 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 (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 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 only a limited influence on the accuracy of functional module identification, whereas the deletion of 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 PPI 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 PPI networks based on a 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 datasets (ALL, DLBCL). These datasets 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, whereas

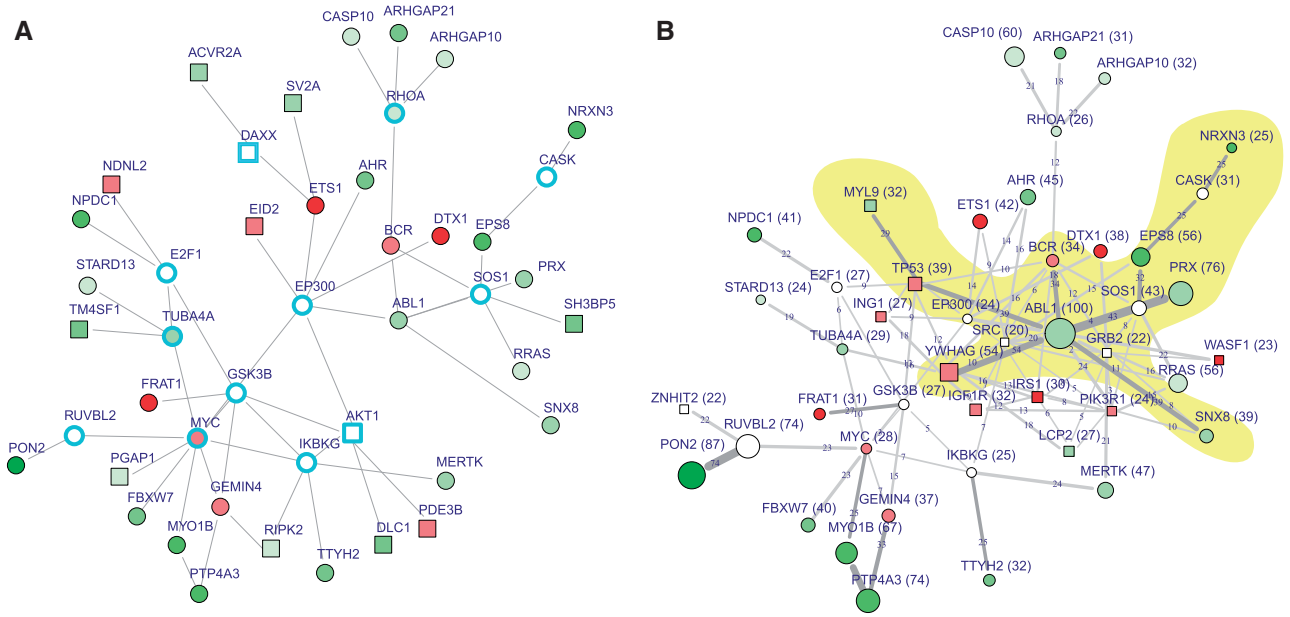

Fig. 4. Modules calculated for the ALL microarray dataset. (**A**) The original `heinz` solution was calculated with an FDR of 0.01. Blue-framed nodes emphasize nodes with negative scores in the original network, all other nodes have positive scores. Nodes only present in either the original module or the corresponding consensus module are depicted by squared node symbols. Coloring of the nodes represents differential expression of the genes (red: upregulated in samples with the BCR/ABL translocation, green: downregulated). (**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 $\geq$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 Supplementary Fig. S7)

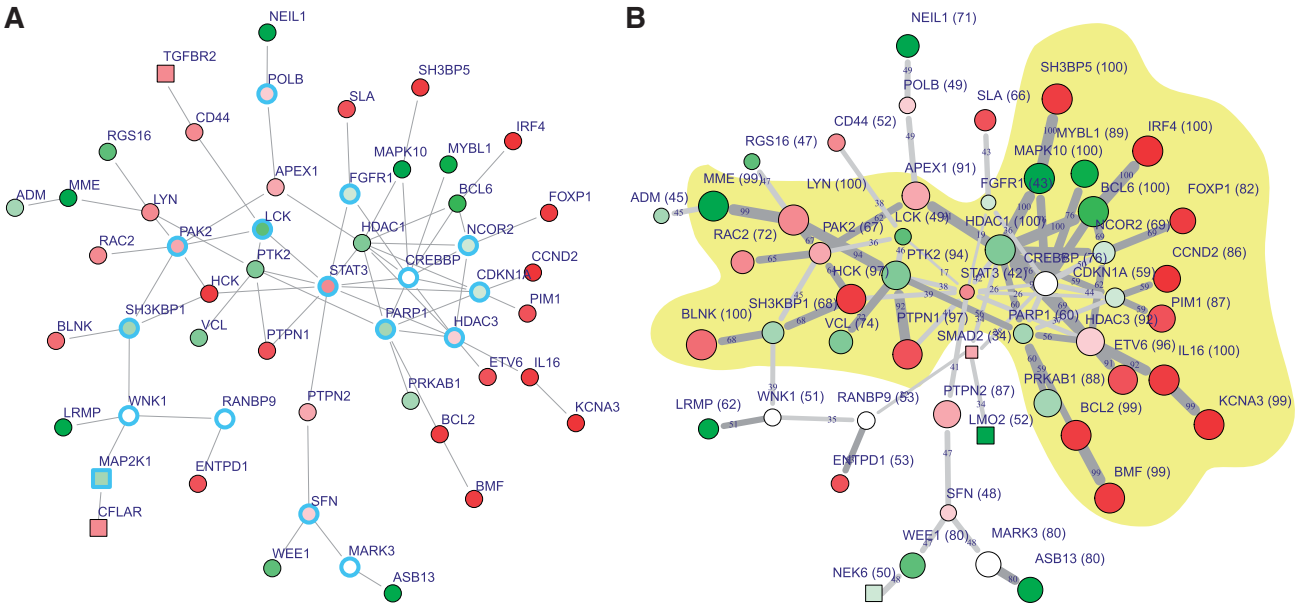

Fig. 5. Modules calculated for the DLBCL microarray dataset. (**A**) The original exact `heinz` solution was calculated with an FDR of 10^{-7}. Blue-framed nodes emphasize nodes with negative scores in the original network, all other nodes have positive scores. Nodes only present in either the original module or the corresponding consensus module are depicted by squared node symbols. Coloring 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)

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 whereas on the other side improving its robustness.

ACKNOWLEDGEMENTS

Funding: This project was supported by the German Federal Ministry of Education and Research, BMBF 0313838A and the German Research Foundation, DFG Da 208/12-1.

Conflict of Interest: none declared.

REFERENCES

Alizadeh,A.A. *et al.* (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

Beisser,D. *et al.* (2010) Bionet: an R-package for the functional analysis of biological networks. *Bioinformatics*, **26**, 1129–1130.

Brown,M. and Forsythe,A.B. (1974) Robust tests for the equality of variances. *J. Am. Statis. Assoc.*, **69**, 364–367.

Cortez,D. *et al.* (1996) The BCR-ABL tyrosine kinase inhibits apoptosis by activating a Ras-dependent signaling pathway. *Oncogene*, **13**, 2589–2594.

Davis,R.E. *et al.* (2001) Constitutive nuclear factor kappaB activity is required for survival of activated B cell-like diffuse large B cell lymphoma cells. *J. Exp. Med.*, **194**, 1861–1874.

Davis,R.E. *et al.* (2010) Chronic active b-cell-receptor signalling in diffuse large b-cell lymphoma. *Nature*, **463**, 88–92.

Deininger,M. (2004) Src kinases in Ph+ lymphoblastic leukemia. *Nat. Genet.*, **36**, 440–441.

Dittrich,M.T. *et al.* (2008) Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, **24**, i223–i231.

Efron,B. (1979) Bootstrap methods: Another look at the jackknife. *Ann. Stat.*, **7**, 1–26.

Falcon,S. and Gentleman,R. (2007) Using GOstats to test gene lists for GO term association. *Bioinformatics*, **23**, 257–258.

Felsenstein,J. (1985) Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, **39**, 783–791.

Felsenstein,J. (2004) *Inferring Phylogenies*. Sinauer Associates, Sunderland, Massachusetts.

Gupta,M. *et al.* (2011) Regulation of stat3 by histone deacetylase-3 in diffuse large b-cell lymphoma: implications for therapy. *Leukemia*, advance online publication, 25 November 2011; doi:10.1038/leu.2011.340.

Haferlach,T. *et al.* (2010) Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the international microarray innovations in leukemia study group. *J. Clin. Oncol.*, **28**, 2529–2537.

Ideker,T. *et al.* (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18** (Suppl. 1), S233–S240.

Kohlmann,A. *et al.* (2008) An international standardization programme towards the application of gene expression profiling in routine leukaemia diagnostics: the microarray innovations in leukemia study prephase. *Br. J. Haematol.*, **142**, 802–807.

Levene,H. (1960) Robust tests for equality variances. In Olkin,I. (ed.) *Contributions to Probability and Statistics*. Stanford Univ. Press., Palo Alto, CA.

Mishra,G.R. *et al.* (2006) Human protein reference database–2006 update. *Nucleic Acids Res*, **34**, D411–D414.

Pounds,S. and Morris,S.W. (2003) Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. *Bioinformatics*, **19**, 1236–1242.

Pu,S. *et al.* (2007) Identifying functional modules in the physical interactome of saccharomyces cerevisiae. *Proteomics*, **7**, 944–960.

Quenouille,M.H. (1956) Notes on bias in estimation. *Biometrika*, **43**, 353–360.

Rosenwald,A. *et al.* (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N. Engl. J. Med.*, **346**, 1937–1947.

Scott,J. *et al.* (2006) Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.*, **13**, 133–144.

Szklarczyk,D. *et al.* (2011) The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res.*, **39**, D561–D568.

Traina,F. *et al.* (2003) BCR-ABL binds to IRS-1 and IRS-1 phosphorylation is inhibited by imatinib in K562 cells. *FEBS Lett*, **535**, 17–22.

Tukey,J. (1958) Bias and confidence in not quite large sample. *Ann. Math. Statist.*, **29**, 614.

Ulitsky,I. and Shamir,R. (2007) Identification of functional modules using network topology and high-throughput data. *BMC. Syst. Biol.*, **1**, 8.

Ulitsky,I. *et al.* (2008) Detecting disease-specific dysregulated pathways via analysis of clinical expression profiles. In *RECOMB'08: Proceedings of the 12th Annual International Conference on Research in Computational Molecular Biology*, Springer, Berlin, Heidelberg, pp. 347–359.

Ulitsky,I. *et al.* (2010) Degas: de novo discovery of dysregulated pathways in human diseases. *PLoS ONE*, **5**, e13367.

Wu,J. *et al.* (2009) Integrated network analysis platform for protein-protein interactions. *Nat. Methods*, **6**, 75–77.