Module Identification from Heterogeneous Biological Data Using Multiobjective Evolutionary Algorithms

Michael Calonder, Stefan Bleuler, and Eckart Zitzler

Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland
michael.calonder@alumni.ethz.ch, {bleuler,zitzler}@tik.ee.ethz.ch

Abstract. This paper addresses the problem of identifying gene modules on the basis of different types of biological data such as gene expression and protein-protein interaction data. Given one or several genes of interest, the aim is to find a group of genes—containing the prespecified genes—that are maximally similar with respect to all data types and sets under consideration. While existing studies follow an aggregation approach to tackle the problem of data integration in module identification, we here propose a multiobjective evolutionary method that provides several advantages: (i) no overall similarity measure needs to be defined, (ii) the interactions and conflicts between the data sets can be explored, and (iii) arbitrary data types can be integrated. The usefulness of the presented approach is demonstrated on different biological scenarios, also in comparison to standard clustering.

1 Motivation

With the advent of different high-throughput measurement technologies it is possible to investigate biological mechanisms on a systems level. While each type of measurements quantifies a different aspect of the cellular behavior such as gene expression, protein-protein interactions or metabolic fluxes, most existing computational analysis methods are designed for a single specific type of measurements. However, many important biological questions cannot be addressed by the analysis of just one type of measurements as they provide a limited view of the biological system under investigation. In such cases, a combined analysis of multiple data types is crucial to reveal the underlying mechanisms and accordingly a lot of research is currently devoted to this topic. Data integration represents a major challenge as the relation between the different data types are often complex.

In this work, we focus on a central part of the computational analysis of high-throughput data, namely *module identification*, i. e., the identification of groups of genes that share a similar biological function or regulation mechanism. Several methods exist for the identification of modules on multiple types of biological data. In [1], Hanisch et al. propose a co-clustering approach of biological networks and gene expression data in which a combined distance function is defined which is used in hierarchical clustering. This approach works with arbitrary networks but was tested on a metabolic network. A different approach proposed in [2] combines distances on the Gene Ontology graph with gene expression data and applies a memetic algorithm for identifying high scoring clusters. A further data type that has been used in a joint analysis is sequence data; the method presented in [3] aggregates three types of distances, namely similarity of gene expression, operon membership, and intergenic distance, into one distance function and applies hierarchical clustering. All these approaches aggregate the similarities on the different data types into one similarity measure. This strategy has two main drawbacks: i) it is often difficult to define a suitable aggregation function as similarity relates to completely different properties in the different data types such as distance on a protein-protein interaction graph and similarity of gene expression patterns; ii) the resulting modules do not give information about the relation of the data types, e. g., it is not possible to determine whether accepting a slightly worse similarity on one data type could increase the similarity on the other data types substantially.

In this paper, we present a multiobjective optimization approach to the problem of module identification based on multiple data types. In particular we pursue the query gene concept as presented in citeOSMV2003b,IFBS2002a for single data types; here a module containing a specified gene is sought. In this respect our approach is designed for a different problem than the only multiobjective approach to module identification previously published which is targeted to partitioning and does not deal with multiple data types citeHK2004a. The advantages of the proposed approach over the aforementioned data integration methods are:

- The method does not require any aggregation function as each data type is associated with a distinct objective function.
- It allows to explore the trade-offs between different data types.
- The framework is applicable to arbitrary data types and similarity measures.

The application of this framework to combinations of three different data types, namely protein-protein interaction networks, metabolic pathways and gene expression data in Arabidopsis and yeast reveals that the amount of conflict between two data types depends heavily on both the specific data types as well as the query genes chosen. Thus, visualizing the trade-off provides additional insight compared to aggregation strategies. Furthermore the proposed multiobjective method can produce better results than multiple runs of a single objective optimizer and the classical k-means algorithm on the considered data set.

2 Optimization Framework

2.1 Model

Given a small set of user defined query genes Q and a target size $s_{\min}$ for the resulting modules, the goal is to identify the best module containing the query gene(s) with respect to the n data sets $D_1, ..., D_n$. The quality of a module for a specific data set D_i can be defined in multiple ways: a straight forward method is to calculate the mean distance to the query genes. An alternative score is given by the mean distance of the module genes to the module centroid. While the former places the query genes in the "middle" of the module the latter allows query genes to be placed on the "border" of a tight module. Note that in the case of a single data set and correspondingly a single objective function ($k = 1$) the former score provides a trivial way of identifying the optimal module by sorting the genes according to their distance to the query genes. In contrast, in the case of multiple data sets ($k > 1$) genes that are close on one data set will in general not also be close on the other data set(s). This results in a multiobjective optimization problem where the score on each data set represents one objective.

Formally, a *module* is defined as a subset of genes $G \subseteq \{1, \ldots, m\}$. Note that the binary search space $\mathcal{X} = 2^{1,\ldots,m}$ of all possible modules is exponential in m, $|\mathcal{X}| = 2^m$. The optimization task consists of solving a minimization problem over several objective functions subject to a size constraint $s_{\min}$,

$$\begin{array}{lcl} \arg\min_{G \subseteq \{1,\ldots,m\}} & \mathbf{f} & = \begin{pmatrix} \mathrm{dist}(G, D_1) \\ \ldots \\ \mathrm{dist}(G, D_n) \end{pmatrix} \\ \text{subject to} & & |G| \geq s_{\min} \in \{2, 3, \ldots, m\} \end{array},$$

where $\mathrm{dist}(G, D_k)$ is the mean distance from all genes to the query gene(s) on data set k.

2.2 Biological Data Types and Distance Measures

In general, arbitrary types of biological data can be used as long as it is possible to define a useful measure of distance between genes based on them. In this work, we analyze three different types of biological data: gene expression (GE), protein-protein interaction (PPI) and metabolic pathway data. This section introduces these data types and describes how distances are measured on these data.

Gene Expression Data Gene expression reflects the current activity of genes. The expression levels of all genes are measured under different conditions or at different time points resulting in a $m \times n$-matrix, E, where m is the number of considered genes and n the number of experimental conditions. In general, genes that exhibit highly similar expression patterns are thought to have a similar biological function. The most common approach for calculating similarity of expression is to define a distance function on the gene expression vectors, and a variety of distance measures has been proposed. Here we apply a distance metric based on ranking of the gene expression values which provided good results in previous studies [7, 8] and in preliminary comparisons to Euclidean distance and

Pearson correlation. Formally the distance measure is defined as

$$\operatorname{dist}(G, D_k) = \frac{1}{|Q|}\frac{1}{|G|}\sum_{\substack{q\in Q\\ g\in G}}\left[\frac{1}{n}\sum_{0\leq i\leq n}\left(r_{gi}^{(k)} - r_{qi}^{(k)}\right)^2\right] \tag{1}$$

where Q is the set of query genes and $r_{ij}^{(k)}$ is an element of the k-th row-wise ranked expression matrix. Ranks are scaled to be in $[0, 1]$.

Protein-Protein Interaction Data Another widely used high-throughput measurement technology provides information about pairwise physical interaction of proteins. In general each protein can be associated to the gene that codes for it. Correspondingly, the interactions between proteins can be regarded as linking the corresponding genes. This yields a symmetrical interaction matrix $I \in \mathbb{R}^{m\times m}$ where m denotes the number of genes. In principal one could use a distance measure for pairs of rows as in (1). However from a biological point of view it is more informative to consider another distance metric similar to the one used in [1]. I can be represented by a graph: there is one node per gene and an edge if I is indicating an interaction. The straightforward measure for the distance between two genes on the graph is the number of hops that lie between them, or the maximum occurring distance if they are not connected. For the PPI data, the distance function is defined as

$$\operatorname{dist}(G, D_k) = \frac{1}{|Q|}\frac{1}{|G|}\sum_{\substack{q\in Q\\ g\in G}} S(g, q), \tag{2}$$

$$S(g, q) = \begin{cases}\sigma_{gq} & \text{if } g \text{ and } q \text{ are connected}\\ \max\limits_{q\in Q, g\in G}\sigma_{gq} + 1 & \text{else}\end{cases},$$

where σ_{gq} is the shortest path from g to q in the interaction graph.

Metabolic Pathway Maps A second type of biological networks consists of a map of metabolic pathways which for many organisms are well studied. By linking enzymes that are active in neighboring reactions, this reaction network can be transformed into a network of enzymes which in turn can be regarded as a network of the corresponding genes. Similar arguments as for the PPI apply here and accordingly the distance on the metabolic pathways can be defined like in the case of PPI data.

3 Implementation of the Evolutionary Algorithm

In the following we will describe the architecture and implementation of a general EA for this optimization problem. As we will see the representation and most of the operators are generic while the initialization and the mutation operator are more specific to the proposed optimization problem.

Each individual represents one module. For reasons of simplicity we use a binary representation with a bit string of length m where a bit is set to 1 if the corresponding gene is included in the module. We apply uniform crossover and a repair mechanism that adds the closest gene (in the average over all data sets) that is not yet in the module until the constraint is met.

For the initial population, it is desirable to have modules of different size. A simple strategy, for example, which sets each bit to 1 with a probability of 0.5 produces a set of modules containing different genes but all modules will have similar sizes. Instead, we choose the size of the modules deterministically such that the they are equally distributed between 2 and m genes. The genes themselves are chosen randomly.

A standard mutation operator as independent bit mutation is not well suited for this problem as only a small fraction of the bits are 1, e. g., 15 out of 22000 and thus many more genes are added to the module than removed. To prevent this we apply a fair single bit mutation where a randomly chosen bit is flipped from 1 to 0 and one randomly chosen bit is flipped from 0 to 1 thus leaving the module size unchanged.

Fitness assignment and selection in this multiobjective scenario is handled by an indicator-based selection method, namely IBEA, a recent method that compared favorably to other state-of-the-art multiobjective evolutionary algorithms [9]. We implemented this problem setting in PISA [10], an interface separating the problem specific parts of an EA from the problem independent parts. All algorithms have been implemented in C++ and are available on `www.tik.ee.ethz.ch/sop/mo_module/`.

4 Results

Several extensive experiments have been carried out in order to evaluate the performance of the proposed algorithm and the capabilities of the proposed methodology in general by applying it to different biological data. As to the first aspect, we have investigated whether a local search strategy improves the overall performance and how the multiobjective approach compares to a scalarization approach with multiple independent runs. Concerning the second aspect we studied the characteristics of the trade-off fronts resulting from different data type combinations, and in addition compared the outcomes with the ones of a classical clustering algorithm, namely k-means.

4.1 Experimental Setup

For the simulation runs we have used three different combinations of data types: two diverse time course gene expression data sets on Arabidopsis provided by the ATGenExpress consortium (containing 6 and 11 time points and 22746 genes each), the first of these gene expression data sets in combination with a manually

Parameter	*Value*
min # genes $s_{\min}$	15
use local search	false
mutation rate p_m	0.1
mutation type	fair single bit
crossover rate p_c	0.1
tournament size	2
population size	100
# generations	100

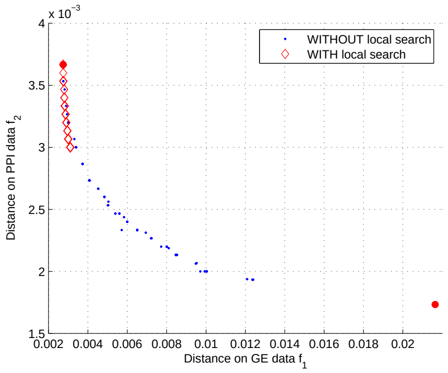

Fig. 1. (left) Default parameter settings for this study. (right) A run with the local search enabled (◇) or disabled (·), respectively. The two bold dots (●) indicate the extreme values.

curated metabolic pathway map [11] (986 genes) and a yeast gene expression data set [12] (3665 genes) in combination with PPI data [13].

Figure 1 (left) summarizes the parameter settings we used for this study. All simulations were run on one Intel Xeon 3.06 GHz CPU with 2 GB RAM.

4.2 Performance of the Proposed Algorithm

Local Search We addressed the question whether the incorporation of a local search heuristic could improve the performance of the EA. The local search proceeds in two steps: first it reduces the size of the module until the minimum number of genes constraint is reached. Then it adds those genes which are closest to the query gene(s) based on the average distance over all objectives and do not increase the mean dissimilarity value of the module at the same time. For a minimum number of genes constraint of 15, this is typically zero to three genes. The effect of the local search is clearly visible in Figure 1 (right): obviously, a preferred search direction is introduced by averaging over the different objectives. This inhibits the EA in settling individuals in the lower f_2 region. This behavior is reflected in a much faster convergence for the optimization runs with local search than without local search. Since we do not want to impose such strong preference for a specific direction, we consider such a local search inappropriate for this type of problem.

Comparison to Chebyshev Scalarization For the purpose of validation of the multiobjective approach we set up several single-objective runs that follow the idea of a Chebyshev scalarization. To generate an approximation of the Pareto set, the single-objective optimizer was run subsequently for 21 weight combinations (5% steps) that were uniformly distributed over the range of all possible weight combinations. The results of these runs were combined into a single non-dominated front. For both the single- and multiobjective runs the number of generations was held constant, i. e., the run time of the single-objective

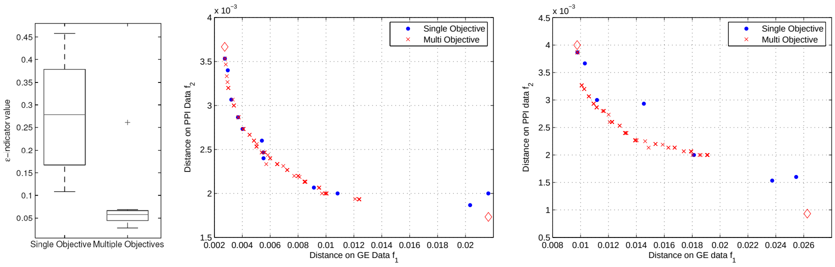

Fig. 2. (left) Scalarization vs. multiobjective optimization. Box plot of ϵ-indicator values. (middle and right) Scalarization versus multiobjective optimization. Representative fronts for two selected query genes. Results of one multiobjective run and a series of single objective runs with different weights. The two diamonds indicate theoretically possible extreme values.

approach was accordingly longer[1]. The input data for this evaluation was the GE/PPI pairing. The simulation was run for 5 randomly chosen query genes (one query gene per run) and 10 different random number generator seeds were used for each query gene. Figure 2 shows the result for two different query genes. On the left, the two algorithms produced comparable fronts. In contrast, the right plot shows a rare case for a query gene where both algorithms encounter problems in advancing towards low f_2 values. This problem is alleviated when the number of generations is increased. The outcomes for the other query genes are somewhere in between these extrema. We would expect the Chebyshev approach to find nearly as many non-dominated points as there are weight combinations, namely 21. This is obviously not the case and we find the multiobjective EA yielding many intermediate points of the Pareto set approximation that the single-objective algorithm did not find. In order to do a statistical assessment, we use the ε-indicator to compare the quality of the fronts, cf. [9]. Roughly speaking, this measure calculates a reference front by collecting all non-dominated solutions from both fronts and then determines the distance by which each front needs to be shifted such that no solution from this front is dominated by the reference front anymore. Based on the Kruskal-Wallis test, the ε values for the multiobjective approach are significantly lower than those of the Chebyshev approach for all query genes with a p-value of 10^{-6} or less. This provides evidence for a superiority of the multiobjective approach over the single-objective algorithm with respect to the ε-indicator, cf. Figure 2 (left). The high variance in the single-objective case results from the above mentioned difficulties to advance into the lower f_2 region which mainly appeared in the single-objective approach. The outlier in the multiple objective case corresponds to the run of Figure 2 (right).

[1] The multiobjective EA took about 16 sec to complete where the single-objective EA needed about 18 times longer (289 sec).

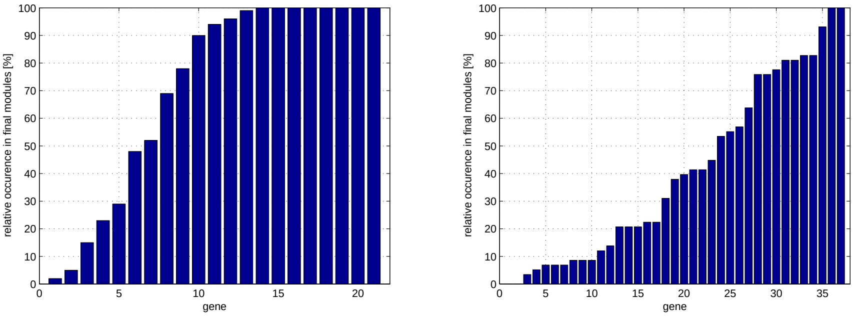

Fig. 3. Comparing the trade-off of GE/GE (left) vs. GE/PPI data (right). The plots show in how many of the non-dominated modules each gene is contained, e. g. in the left case, Gene number 10 occurs in about 90% of all modules that the algorithm found.

4.3 Application to Different Biological Scenarios

Exploring the Trade-offs For all of the three pairings of input data (GE/GE, GE/PPI, GE/metabolic) we quantify the trade-offs and show that they widely vary for the different data by comparing them against each other. All runs in this section comply with the default configuration of Figure 2 (left) and each simulation was run for a single query gene. Five query genes were randomly chosen for this analysis and ten runs with different seeds were performed for each query gene, leading to a total of 50 runs.

a) GE vs. GE data (Arabidopsis). For this pairing we found only little trade-off; the front closest to the origin in Figure 4 corresponds to this case. In none of the runs we encountered more conflict than indicated by this plot. The absence of conflict is also reflected in the diversity of the modules: Figure 3 (left) shows that more than half of all genes occur in 90% of the modules.
b) GE vs. PPI data (Yeast). In the case of a GE/PPI pairing we found a much stronger trade-off between the two objectives, compared to the preceding case. Figure 4 again depicts the resulting front (the middle one). This can be clearly verified from Figure 3 (right) that reveals a much larger diversity among the modules: less than 10% of the genes occur in 90% of the modules.

c) GE vs. metabolic data (Arabidopsis). Between these two data types we observed the largest trade-off, as shown in Figure 4 (left).

Figure 4 (right) shows the statistical distribution of the hyper-volume indicator [14] for each of the three fronts on the left and 10 different random generator seeds [2]. Again, this clearly documents that we find the most conflict in GE/metabolic data pairs as the plot on the left would imply. These differences demonstrate clearly the advantage of the multiobjective approach compared to an aggregation based method where only one point on the front is generated.

[2] The objective values are scaled to $[1, 2]$ and the reference point is $(2.1, 2.1)$

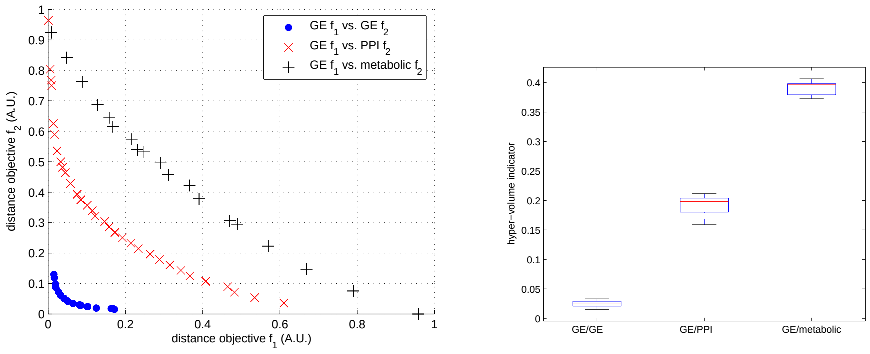

Fig. 4. (left) Comparing representative fronts of the different data type combinations for one query gene. (right) Related boxplot for the hyper-volume indicator for five query genes and ten seeds each.

Comparison to k-means We substantiate the usefulness of an evolutionary approach compared to a standard clustering method by comparing the proposed algorithm to the well-known k-means algorithm.

For a test data set we selected the GE/PPI pairing and proceed in four steps: first, we ran k-means only on the GE data. Second, we selected randomly a query gene. For the cluster that contains the query gene we calculated *both* objective values, on the GE and PPI data and received the k-means “front”, consisting of only one point. Third we used the same query gene as input to the EA and set the minimum number of genes to the size of the k-means cluster. Finally we compared the front produced by the EA to the one point resulting from the k-means algorithm and repeated this procedure 50 times, varying seeds and query genes. Note that the ε value indicates whether at least one EA module dominates the k-means cluster, i. e., it is better in both objectives.

Using the two-sided Wilcoxon signed rank test we showed that the EA performs significantly better in this respect than k-means with a p-value of $1.1 \cdot 10^{-9}$. Thus, k-means is not able to produce results that compare well to the evolutionary approach, not even when comparing on the GE objective only.

5 Conclusion

Several approaches exist for co-clustering of multiple biological data types [1–3]. All these approaches are based on an aggregation function that combines distance measures on the different data types into one distance measure, thereby fixing the relative importance of the different data types and obscuring potential conflict between the data types. In order to overcome these shortcomings, we have presented a flexible framework for module identification that is based on multiobjective optimization which does not need any aggregation function to be defined and additionally makes potential conflicts between data types visible.

The second main difference is that our approach provides a way to guide the search by specifying one or a few query genes which are contained in the resulting modules.

The effectiveness of the suggested approach was demonstrated on gene expression, protein-protein interaction and metabolic pathway data sets from Arabidopsis and yeast. Comparisons to a scalarization approach and to the k-means algorithm clearly show the advantage of the multiobjective optimization. The simulation results also revealed that the amount of conflict between two data types varies largely depending on the data sets and the specific query genes. This demonstrates that by defining a single aggregation function important information about the resulting modules may be missed.

Interesting future steps in this line of work include additional comparisons to existing algorithms such as [1] and [4] and more elaborated measures of similarity on biological graphs. Weighted edges that represent the probability or strength of an interaction can be easily included in order to adapt the method to specific biological problems.

References

1. Hanisch, D., Zien, A., Zimmer, R., Lengauer, T.: Co-clustering of biological networks and gene expression data. Bioinformatics **18**(Suppl. 1) (2002) S145–S154
2. Speer, N., Spieth, C., Zell, A.: A memetic co-clustering algorithm for gene expression profiles and biological annotation. In: CEC04, IEEE (2004) 1631–1638
3. Steinhauser, D., et al.: Hypothesis-driven approach to predict transcriptional units from gene expression data. Bioinformatics **20**(12) (2004) 1928–1939
4. Owen, A.B., et al.: A gene recommender algorithm to identify coexpressed genes in c. elegans. Genome Res **13**(8) (2003) 1828–1837
5. Ihmels, J., et al.: Revealing modular organization in the yeast transcriptional network. Nature Genetics **31**(4) (2002) 370–377
6. Handl, J., Knowles, J.: Evolutionary multiobjective clustering. In: PPSN VIII. Volume 3242 of LNCS., Springer (2004) 1081–1091
7. Bleuler, S., Zitzler, E.: Order preserving clustering over multiple time course experiments. In: EvoWorkshops 2005. Number 3449 in LNCS, Springer (2005) 33–43
8. Prelić, A., et al.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9) (2006) 1122–1129
9. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: PPSN VIII. LNCS, Springer (2004)
10. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA — a platform and programming language independent interface for search algorithms. In: EMO03. LNCS, Springer (2003) 494–508
11. Wille, A., et al.: Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. Genome Biol **5**(11) (2004)
12. Gasch, A.P., et al.: Genomic expression programs in the response of yeast cells to environmental changes. Mol Biol Cell **11**(12) (2000) 4241–57
13. Salwinski, L., et al.: The database of interacting proteins: 2004 update. Nucleic Acids Res **32**(Database issue) (2004) D449–51
14. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology (ETH) Zürich, Switzerland (1999)