Searching for functional gene modules with interaction component models

Juuso Parkkinen and Samuel Kaski
Helsinki Institute for Information Technology and Adaptive Informatics Research Centre
Laboratory of Computer and Information Science, Helsinki University of Technology
juuso.parkkinen@tkk.fi,samuel.kaski@tkk.fi

Abstract

Genetic functional modules and protein complexes are being sought from combinations of gene expression and protein-protein interaction data with various clustering-type methods. As far as we know, up to now these methods have used the interaction data as constraints on the clustering of expression data, instead of modeling the noise in the interactions. We model the interaction links with a simple generative "topic model," which is augmented to generate also the expression data. The results outperform a representative set of earlier models in the task of finding modules having enriched functional classes. Moreover, it turns out that the generative model for the links alone works very well in this task.

1 Introduction

Search for functional modules of genes, and of interacting protein complexes, are tasks where high-throughput measurement techniques combined with machine learning may help. Many probabilistic and other approaches have recently been introduced, some quite successful in finding functionally enriched modules. In particular methods for combining different data types, relational interaction data and functional gene expression data, have been intensively studied.

Ulitsky and Shamir [1] very recently used similarities between gene expression patterns as a kind of interaction data between proteins. They combined these links with protein-protein interaction data in order to seek *Jointly Active Connected Subnetworks* (JACS). Their novel computational method called MATISSE found biologically relevant modules better than a set of earlier methods (Co-clustering [2] and CLICK [3]).

Another recent method [4] uses an interaction network to form prior constraints on the clustering of gene expression data. The method is an extension of Markov random fields, called *hidden modular random fields*. The constraints improved performance in the task of finding functionally enriched modules, compared to only using gene expression data.

Although these models derive the interaction network from different sources and use it in different ways, it is common to both that they consider the interaction network as constraints for the clustering and do not seriously take into account uncertainties in the network. This is sensible in case the interaction network is very reliable, being manually curated for instance. But protein-protein interaction data are known to be notoriously noisy, and even manual curation may not be able to remove all uncertainties in data. Hence it would make sense to include a generative model for the interaction data as well. We have recently introduced a generative model for graphs [5] which might be suitable for the task. It assumes that the edges or here *interactions* come from latent components. In this paper we introduce ways of incorporating functional data related to the nodes, the genes or proteins, into the

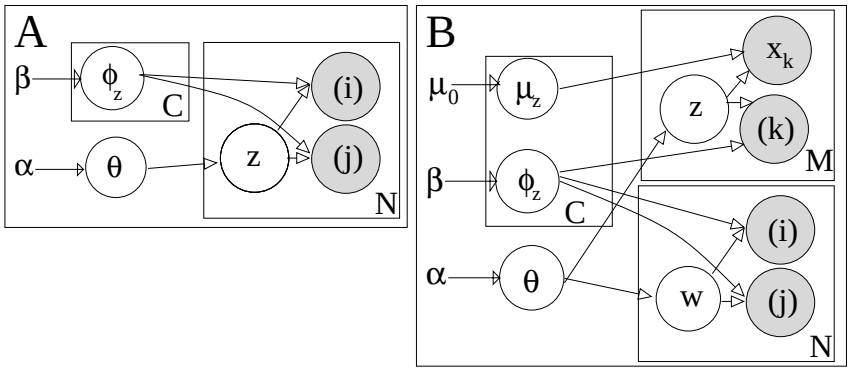

Figure 1: **Plate graphs of two interaction component models.** (A) Generative model of interactions. Each interaction is generated from a component z by sampling independently the end points (i) and (j) from the multinomial distribution ϕ_z. (B) Extension with gene expression data x_k in the nodes; lower part of the plate is the same as in (A). Node data is generated from a component z by sampling each node k from ϕ_z and then its data from the normal distribution $N(\mu_z, \sigma^2 I)$. Symbols: multinomial distribution θ over components, component-specific multinomial distributions ϕ_z over nodes, distribution hyperparameters α and β, prior and component-specific means μ_0 and μ_z.

model. The underlying assumption is that interacting complexes or modules share functional properties as well.

2 Combining relational and expression data

The generative model we use for the interaction links [5] assumes that each link comes from a latent component, each component having a characteristic distribution over nodes. The links are generated (Fig. 1A) by first choosing the component z based on the multinomial distribution parameterized by θ, and then choosing the endpoints i and j of the link according to the multinomial distribution ϕ_z of the component z. Note that in the generative process each link belongs to one component; nodes may belong to several.

This model has been proven effective in detecting meaningful communities in very large social networks. Here we use the same model structure in searching for functional modules among protein interaction networks; then the model might be called *interaction component model*.

The existing model is capable of handling uncertainties in the protein-protein interaction data; next we introduce two ways of extending the model to take into account functional data available about the nodes, here gene expression data.

One simple way of incorporating the gene expression data is to assume that the same components generate the gene expression data as well (Fig. 1B). This leads to modules which are both strongly interconnected and also share similar expression profiles. In practice, the component z from which the gene expression profile x_k is generated is sampled from the same distribution ϕ_z as the endpoints of the links. Note that for computational reasons we have simplified the model by not including the known fact that each node has exactly one gene expression profile.

Another, even simpler way of including functional data about the genes is to transform the data into links and to include these links into the graph. Here we compute the correlation of expression for each pair of genes, and include all pairs where the correlation exceeds a fixed threshold.

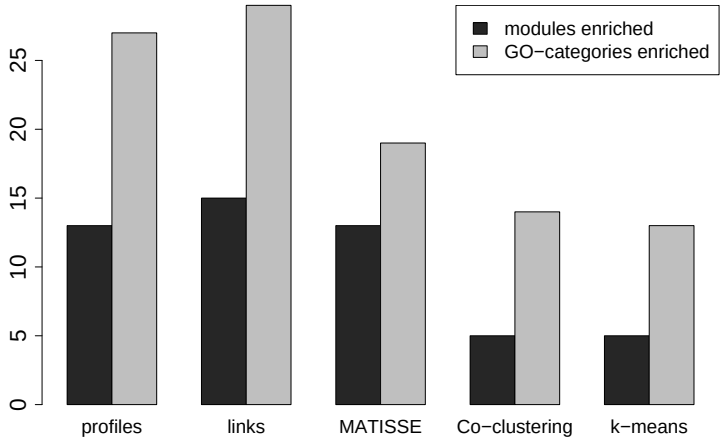

Figure 2: **Performance of the methods on combined interaction and expression data.** Black bars: the number of modules in which at least one category is enriched. Grey bars: GO categories enriched in at least one module. Methods: *profiles*: generative model shown in Fig. 1B. *links*: expression converted to links.

2.1 Details

The model is applied to data from the budding yeast *Saccharomyces cerevisae* [6]. The data set, "full data," contains 21679 interactions for 4543 proteins, and expression profiles for 1990 genes. Each profile consists of 133 measurements. In a part of the study we include only the proteins for which full expression profiles are available, the reason being that handling of missing data has not been implemented in the models. This subset, "small data," contains 7123 interactions, and expression profiles for 1475 proteins.

The hyperparameters were chosen a priori and not optimized. The network hyperparameters were chosen to favor discrete clusters ($\alpha = 10^{-5}$ and $\beta = 10^{-4}$). The parameters for expression profiles were set to describe variations around the base value of zero ($\mu_0 = 0$, $\sigma_0^2 = 1$ and $\sigma^2 = 0.1$). The threshold of gene expression correlation, above which the genes were considered interacting, was fixed (0.82) to produce roughly the same number of links (7057) that was originally in the "small data." The number of components was set to 20, which is the same number of modules found by MATISSE.

The models were computed with collapsed Gibbs sampling. The length of the burn-in period was 20000 iterations, after which 20 samples were taken with an interval of 50 iterations. The results were computed by averaging over those 20 samples.

For comparing the methods we use the same measures as Ulitsky et al. [1]; the found modules are compared to existing annotation databases. The TANGO-algorithm, provided by Ulitsky et al. in their MATISSE software, performs a comprehensive GO biological process annotation enrichment analysis, with correction for multiple testing.

2.2 Experiments and results

Components were computed from the "small data" set with our two models, MATISSE, Co-clustering, and k-means. Co-clustering uses both interaction and expression data, while k-means uses only expression data. Default settings were used for all methods, and our models sought for 20 modules (same amount that MATISSE found, to give MATISSE slight advantage). Figure 2 shows, for each method, the number of modules where at least one GO-category is significantly enriched, and the number of GO-categories enriched in at least one module ($p < 0.05$).

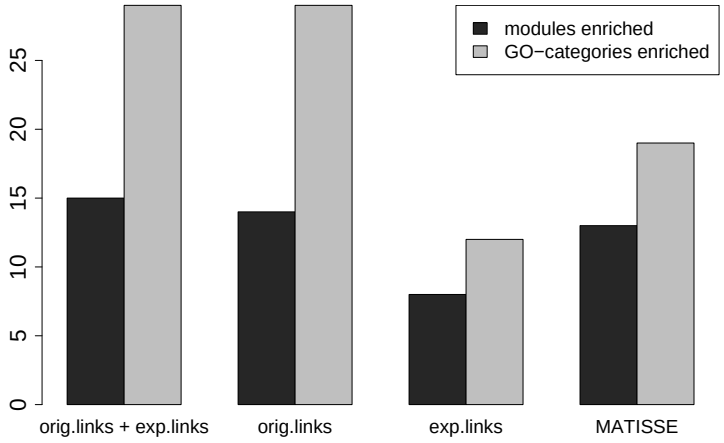

Figure 3: **Effect of data sources on the results in the generative model.** "orig.links" uses only the interaction links, "exp.links" only links derived from expression data, and "orig.links + exp.links" has pooled the two kinds of links together. MATISSE has been included for reference. Experiments were done on the "small data."

Our two models and MATISSE found roughly the same number of enriched modules, while the other two methods perform seemingly poorer. By contrast, our methods seem to clearly outperform MATISSE in finding enriched GO-categories. Using expression data as interactions seems to give slightly better results than generating the gene expression profiles.

3 Gene expression data vs. interaction data

During the previous experiments we noticed that the generative models worked surprisingly well even without the gene expression data. We will next check how much the different data sources contribute to the results. The conclusions are naturally only valid for the chosen goodness measure and for the specific ways of combining the data sources.

3.1 Experiments and results

Figure 3 shows that including the expression links does not improve the results compared to only using the interaction links. Using only expression links seems not to be reasonable at all.

For this set of methods it is possible to run the same experiments on the full data set as well. The results shown in Figure 4 reveal that the interaction links perform better than the combination of expression and interaction links. The reason probably is that using the expression links biases the modules to favor the subset of nodes for which expression links are available. These results additionally suggest that the generative models outperform MATISSE.

4 Conclusions

We have used a simple generative graph model for relational data to search for functional modules of genes. We have also introduced two approaches for combining gene expression data with the interaction data.

Experiments with data from the budding yeast suggest two main conclusions: (i) Generative modeling of the interaction data is advantageous. The proposed models outperformed the

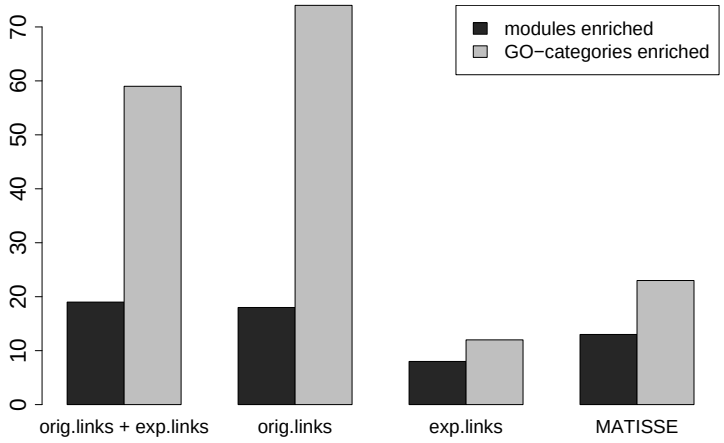

Figure 4: **Effect of data sources on the full data set.** Key: see caption of Figure 3

state-of-the-art method MATISSE which in turn has been shown to outperform several alternatives. The tradeoff is computational time, for MATISSE runs clearly faster. (ii) Models that are based on interaction data alone perform surprisingly well, even better than the introduced simple methods of combining interaction and expression data.

The results may naturally depend on the chosen goodness measure. Here we used enrichment of GO biological process categories in the modules.

The results of MATISSE differ from the results of the introduced methods also in that MATISSE leaves many of the nodes out from the discovered modules, in effect finding smaller modules. MATISSE additionally has a restriction for a module size. We find two very big modules, which effectively model the background and not functional sets, and the rest of the modules vary in size between 80 and 300 genes. In this work we set the number of modules to equal that of MATISSE; it would probably improve the results further if the priors were made to favour several small modules and one or at most a few large ones to cover the background.

Most GO-categories enriched in our solutions are still found in distinct modules (only about 10% overlap) and further, only a few GO-categories found by MATISSE were not found by our methods. These results suggest that the generative models are good in finding biologically relevant modules. Increasing the number of modules could lead to an even better distinction between functional groups.

Acknowledgments

This work was supported in part by the Academy of Finland, decision number 207467, and in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

[1] Igor Ulitsky and Ron Shamir. Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology*, 1:8, 2007.

[2] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18:S145–S154, 2002.

[3] R. Sharan and R. Shamir. CLICK: A clustering algorithm with applications to gene expression analysis. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, volume 8, pages 307–316. AAAI Press, 2000.

[4] M. Shiga, I. Takigawa, and H. Mamitsuka. Annotating gene function by combining expression data with a modular gene network. *Bioinformatics*, 23:i468–i478, 2007.

[5] Janne Sinkkonen, Janne Aukia, and Samuel Kaski. Inferring vertex properties from topology in large networks. In *MLG'07, the 5th International Workshop on Mining and Learning with Graphs*. 2007.

[6] Matisse web page. [http://acgt.cs.tau.ac.il/matisse/].