Uncovering the structure of heterogeneus biological data: fuzzy graph partitioning in the k-partite setting

Florian Blöchl[1,2], Mara L. Hartsperger[1,2,*], Volker Stümpflen[1], Fabian J. Theis[1]

[1]Institute for Bioinformatics and Systems Biology, Helmholtz Zentrum München
[2]Equal contributors

Abstract: With the increasing availability of large-scale interaction networks derived either from experimental data or from text mining, we face the challenge of interpreting and analyzing these data sets in a comprehensive fashion. A particularity of these networks, which sets it apart from other examples in various scientific fields lies in their k-partiteness. Whereas graph partitioning has received considerable attention, only few researchers have focused on this generalized situation. Recently, Long et al. have proposed a method for jointly clustering such a network and at the same time estimating a weighted graph connecting the clusters thereby allowing simple interpretation of the resulting clustering structure. In this contribution, we extend this work by allowing fuzzy clusters for each node type. We propose an extended cost function for partitioning that allows for overlapping clusters. Our main contribution lies in the novel efficient minimization procedure, mimicking the multiplicative update rules employed in algorithms for non-negative matrix factorization. Results on clustering a manually annotated bipartite gene-complex graph show significantly higher homogeneity between gene and corresponding complex clusters than expected by chance. The algorithm is freely available at `http://cmb.helmholtz-muenchen.de/fuzzyclustering`.

1 Introduction

With the relatively cheap availability of biological high-throughput methods such as microarrays, machine learning techniques gain more and more importance in the field of bioinformatics. Learning approaches often focus on the analysis of homogeneous data sets that can be represented as a network having vertices of a single type only. However, many real-world networks are heterogeneous and involve objects of multiple, related types, thus forming k-partite graphs consisting of diverse types of vertices. A key question of clustering-based approaches is how to interpret the global organization of these networks as the coexistence of their structural subunits associated with more highly interconnected parts. Identifying these a priori unknown building blocks such as for instance the common genetic origin of different diseases is crucial for the understanding of the structural and functional properties of such networks.

Most available clustering methods cannot be applied to k-partite networks because they do not treat the single node types (partitions) separately and therefore do not represent the global community structure correctly. While this has been addressed in terms of algorithms for some time now [Bar07, GL04, KAKS97, ZHS07, LWZY06], not many possible applications exist yet in bioinformatics, although the field commonly deals with such networks [KHT09]. A particular issue that may hamper application to bioinformatics may be that most existing algorithms identify separated, disjoint clusters by assigning each data point to exactly one cluster [Mac67, JD88], whereas most biological networks consist of highly overlapping cohesive groups of vertices. A single data point can therefore belong to more than only one cluster, e.g. a large fraction of proteins belong to several protein complexes simultaneously [RBDK$^+$08]. So far only a few approaches exist that allow the detection of overlapping clusters by assigning either each data point a degree of belonging to clusters or to several clusters respectively [Bez81, PDFV05].

In order to identify clusters in heterogeneous data and moreover connect these clusters between the different node types, we developed a fuzzy partitional clustering method based on a non-negative matrix factorization (NMF) model [LS99]. We demonstrate that we can identify biological meaningful overlapping clusters in k-partite graphs. We applied our method to a bipartite gene-protein complex graph representing the manually annotated Corum core set [RBDK$^+$08]. The extracted clusters show significantly higher homogeneity between gene and corresponding complex clusters than expected by chance.

2 A multiplicative update rule for fuzzy k-partite clustering

Recently, an algorithm for the partitioning of k-partite graphs has been put forward in [LWZY06]. It clusters each node set of the graph separately; then the clusters are connected via a smaller, weighted k-partite graph. The algorithm consists of an alternating minimization procedure: first the nodes in each layer are clustered in order to minimize the distance to the small representative graph (change). Then the hidden graph (backbone graph) is updated according to the same cost function.

A key assumption made in [LWZY06] is that the assignment in the first step is made in a binary fashion. This hard clustering is a feature that often is achieved by soft clustering algorithms when not forcing explicit cluster overlap [Bez81]. However it can be easily seen that the cost function put forward in [LWZY06] is not fully minimized by this approximation.

Here, we address the minimization using a multiplicative update algorithm. In contrast to the above method, by not choosing any binary assignment a priori, we observe a close to binary assignment mostly in the hidden nodes, whereas the clustering in each node-type is soft. The resulting algorithm is similar in structure to multiplicative algorithms for NMF, with the difference that we address a three-matrix factorization problem, see e.g. [DS06], and have to deal with a multi-summand cost function.

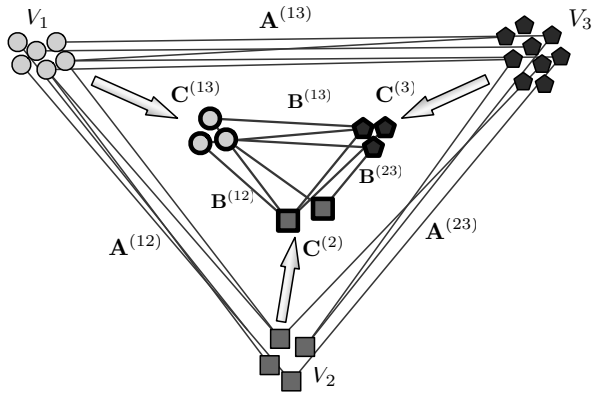

Figure 1: (a) definition of a 3-partite graph G with notation used. (b) approximation of G using a smaller 3-partite graph H defined on fuzzy node clusters.

2.1 Definitions and factorization model

A k-partite graph is a graph $G = (V, E)$ and a partition of the vertices V into k disjoint sets V_i such that no two vertices in the same subset are adjacent. So edges are only allowed between different subsets ('colors'). Let $n_i := |V_i|$ be the number of vertices in partition i. We represent the graph as a set of $n_i \times n_j$ matrices $\mathbf{A}^{(ij)}$ with $1 \leq i < j \leq k$. Commonly, each matrix element is either 0 or 1, but we only restrict the matrices to have non-negative coefficients thereby allowing weighted graphs as well. We can readily include directed instead of undirected k-partite graphs by specifying incidence matrices also for $i > j$. It is easy to see that the following cost function and optimizations generalize to this situation.

We want to approximate G by a smaller cluster network H (*backbone network*), which is defined on fuzzy clusters of each G-partition V_i. For simplicity we for now fix the number of V_i-clusters to m_i. We say that a non-negative $n_i \times m_i$-matrix $\mathbf{C}^{(i)}$ is a *fuzzy clustering* of V_i, if it is right-stochastic i.e. $\sum_l c_{kl}^{(i)} = 1$ for all k. Then we search for a k-partite graph H with $m_i \times m_j$ incidence matrices $\mathbf{B}^{(ij)}$ and fuzzy clusterings $C := (\mathbf{C}^{(i)})_{i=1,\ldots,k}$ such that the connectivity explained by H is as close as possible to G after clustering.

We can measure this difference in many different ways. In [LWZY06], this choice has been circumvented by specializing on arbitrary Bregman divergences, which still allow efficient reformulation of gradient-type algorithms without knowing the specific formula. This is also possible in our case of multiplicative update rules, as has been shown for NMF in [DS06]. However, for simplicity, we choose the minimum square distance as cost function. This implies minimization of

$$f(H, C) := \sum_{i<j} \left\| \mathbf{A}^{(ij)} - \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \right\|_F^2 \tag{1}$$

where $\|.\|_F^2$ denotes the squared Frobenius norm, i.e. the square sum of the matrix elements. The model, the used definitions and the approximation are illustrated in figure 1.

2.2 Derivation of the algorithm

We want to minimize $f(H, C)$ from (1) using a local algorithm extending gradient descent. We assumed an undirected k-partite graph, so $\mathbf{A}^{(ij)}$ is undefined for $i > j$. Hence, we now set $\mathbf{A}^{(ij)} := (\mathbf{A}^{(ji)})^\top$ for $i > j$ (and similarly for $\mathbf{B}^{(ij)}$). Then we find

$$\frac{\partial f}{\partial b_{rs}^{(ij)}} = -2\left((\mathbf{C}^{(i)})^\top \mathbf{A}^{(ij)} \mathbf{C}^{(j)} - (\mathbf{C}^{(i)})^\top \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)}\right)_{rs}$$
$$\frac{\partial f}{\partial c_{rs}^{(i)}} = -2\sum_{j\neq i}\left(\mathbf{A}^{(ij)} \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top - \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}.$$

Minimizing f by alternating gradient descent, we now simply start from an initial guess of $\mathbf{B}^{(ij)}, \mathbf{C}^{(i)}$ and alternate between updates of the $\mathbf{B}^{(ij)}$ and the $\mathbf{C}^{(i)}$ with according learning rates. Such update rules however have two disadvantages: for one, the choice of update rate η (possibly different for $\mathbf{B}$, $\mathbf{C}$ and i, j) is unclear; in particular, for too small η convergence may take too long or may not be achieved at all, whereas for too large η we may easily overshoot the minimum. Moreover, the resulting matrices may become negative. Therefore, we follow Lee and Seung's idea for NMF [LS99] and rewrite this into multiplicative update rules. Hence, let us choose update rates

$$\eta_{rs}^{(ij)} := \frac{b_{rs}^{(ij)}}{2\left((\mathbf{C}^{(i)})^\top \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)}\right)_{rs}}$$
$$\eta_{rs}^{(i)} := \frac{c_{rs}^{(i)}}{2\left(\sum_{j\neq i} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}}$$

Plugging this into the gradient descent equations, this results in the desired multiplicative update rules

$$b_{rs}^{(ij)} \leftarrow b_{rs}^{(ij)} \frac{\left((\mathbf{C}^{(i)})^\top \mathbf{A}^{(ij)} \mathbf{C}^{(j)}\right)_{rs}}{\left((\mathbf{C}^{(i)})^\top \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)}\right)_{rs}} \tag{2}$$

$$c_{rs}^{(i)} \leftarrow c_{rs}^{(i)} \frac{\left(\sum_{j\neq i} \mathbf{A}^{(ij)} \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}}{\left(\sum_{j\neq i} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}} \tag{3}$$

2.3 Algorithm formulation and relation to other work

We note that we can readily show that these update rules do not increase the cost function (1). This can be shown via auxiliary functions similar to NMF [LS01] and multi-factor NMF [DS06], which has been applied in a related model for co-clustering of microarray data [CDGS04]. This theoretical result implies convergences of the update rules. However in contrast to early statements in NMF [LS01], this does not necessarily imply convergence

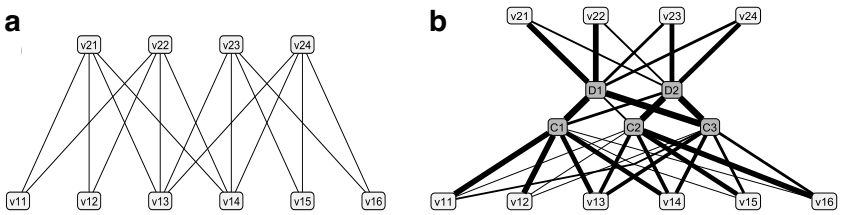

Figure 2: Toy example of a bipartite graph (a) from [LWZY06], with its backbone network and fuzzy clusters (b). Note that neither of the two clusterings are binary.

to stationary points of the Euclidean norm (zero of the differential from (1)), since the update steps may be too small to reach those points. Another possible drawback of such multiplicative updates is the fact that once a matrix entry has been set to zero (which may happen due to zeros in $\mathbf{A}^{(ij)}$ or to numerics), the coefficient will never then be able to become positive again during learning.

We have not yet taken into account the constraint that the cluster matrices $\mathbf{C}^{(i)}$ are required to be right-stochastic i.e. $\mathbf{C}^{(i)}\mathbf{e} = \mathbf{e}$ for $\mathbf{e} = (1, \ldots, 1)$. For simplicity, we force this constraint by regularly projecting each row of $\mathbf{C}^{(i)}$ onto the sphere of the 1-norm. Alternatively, we may introduce this constraint as Lagrange parameter, and get modified cost function with weighted Lagrange parameters. We can still prove non-increasingness of the multiplicative update rule along the lines of [DS06]. The final fuzzy k-partite clustering algorithm is summarized in algorithm 1. An implementation is freely available at `http://cmb.helmholtz-muenchen.de/fuzzyclustering`. In figure 2, we illustrate the feasibility of the algorithm on a small bipartite toy example.

Our algorithm contains two nested loops over the number of partitions. The update steps are fully vectorized and contain only matrix products of non-square matrices. The total time complexity of the algorithm can therefore be estimated as

$$\#\text{iterations} \times \mathcal{O}(k^2\, n_{\max 1} n_{\max 2}\, m_{\max})\,. \tag{4}$$

Here, $n_{\max 1}$ and $n_{\max 2}$ denote the sizes of the largest and the second-largest partition, $m_{\max}$ is the maximum number of clusters to extract within any partition. Hence, the algorithm is fast and efficient. The runtime is linear in each partition size and grows only quadratic in the total number of nodes in the case of graphs with similarly large partitions.

In order to extend cost functions in (unipartite) data clustering to include fuzzy clusters, commonly a so-called *fuzzification factor* $m > 1$ is introduced [Bez81, Dun73]. Instead of squared norm minimization of the residuals, a higher residual power is minimized, which results in overlapping non-trivial cluster assignments. However, we will find that already the standard case $m = 1$ may suffice to introduce non-trivial overlapping clusters. This is because we are interested in co-clustering, which is different from standard data clustering where only a unipartite graph and hence $\mathbf{C}^{(i)} = \mathbf{C}^{(1)}$ is assumed.

Algorithm 1: fuzzy k-partite clustering

Input: k-partite graph G with possibly non-negatively weighted edge matrices $\mathbf{A}^{(ij)}$, $i < j$, number of clusters $m_1, \ldots, m_k$

Output: fuzzy clustering $\mathbf{C}^{(i)}$ and k-partite cluster graph H given by matrices $\mathbf{B}^{(ij)}$

Initialize $\mathbf{C}^{(i)}, \mathbf{B}^{(ij)}$ to random non-negative matrices.
Normalize $c_{rs}^{(i)} \leftarrow c_{rs}^{(i)} / (\sum_t c_{rt}^{(i)})$ for all i, r, s
repeat
  update fuzzy clusters
  for $i \leftarrow 1, \ldots, k$ **do**
    $\mathbf{C}^{(i)} \leftarrow \mathbf{C}^{(i)} \otimes (\sum_{j \neq i} \mathbf{A}^{(ij)} \mathbf{C}^{(j)} \mathbf{B}^{(ij)\top}) \oslash (\sum_{j \neq i} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} \mathbf{C}^{(j)\top} \mathbf{C}^{(j)} \mathbf{B}^{(ij)\top})$
    Normalize $c_{rs}^{(i)} \leftarrow c_{rs}^{(i)} / (\sum_t c_{rt}^{(i)})$ for all r, s
  end
  update k-partite cluster graph H
  for $i \leftarrow 1, \ldots, k-1$ **do**
    for $j \leftarrow i+1, \ldots, k$ **do**
      $\mathbf{B}^{(ij)} \leftarrow \mathbf{B}^{(ij)} \otimes (\mathbf{C}^{(i)\top} \mathbf{A}^{(ij)} \mathbf{C}^{(j)}) \oslash (\mathbf{C}^{(i)\top} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} \mathbf{C}^{(j)\top} \mathbf{C}^{(j)})$
    end
  end
until *convergence*;
Note: $\otimes$ and $\oslash$ symbolize elementwise multiplication and division, respectively.

3 Fuzzy clusters and backbone of a gene-complex hypergraph

In order to illustrate the applicability of our method to heterogeneous biological data we employ the Corum core set [RBDK$^+$08] that reflects a non-redundant catalogue of experimentally verified mammalian protein complexes manually annotated at MIPS. A bipartite graph $G = (V, E)$ with $|V| = 4877$ and $|E| = 8738$ was constructed from these data. The two disjoint node sets are represented by protein complexes and their associated genes further referred to as V_c and V_g, respectively. We then focused on a reduced data set G' with $|V'| = 4090$ and $|E'| = 7946$ retrieved by extracting the maximally connected subgraph. The remaining graph consisted of 1728 complex (V_c) and 2362 gene (V_g) vertices.

The determination of the number of clusters for each node type, in which the graph has to be decomposed, is difficult, and even in the case of unipartite k-means does not allow a direct and computationally simple answer. To address this issue we approximated the number of clusters to be found in the complex and the gene partition respectively by limiting the maximal number of clusters k_c for V_c according to $k_c = \lfloor \sqrt{|V_c|/2} \rfloor$, and then scaled the number of clusters k_g for V_g by $k_g = \lceil k_c \sqrt{|V_g|/|V_c|} \rceil$. We calculated the value of the cost function for each pairwise combination starting from k_c=1. Due to random initial conditions, the algorithm is inherently indeterministic. Therefore, we discuss performance over 10 runs each. Figure 3(a) shows the distribution of cost func-

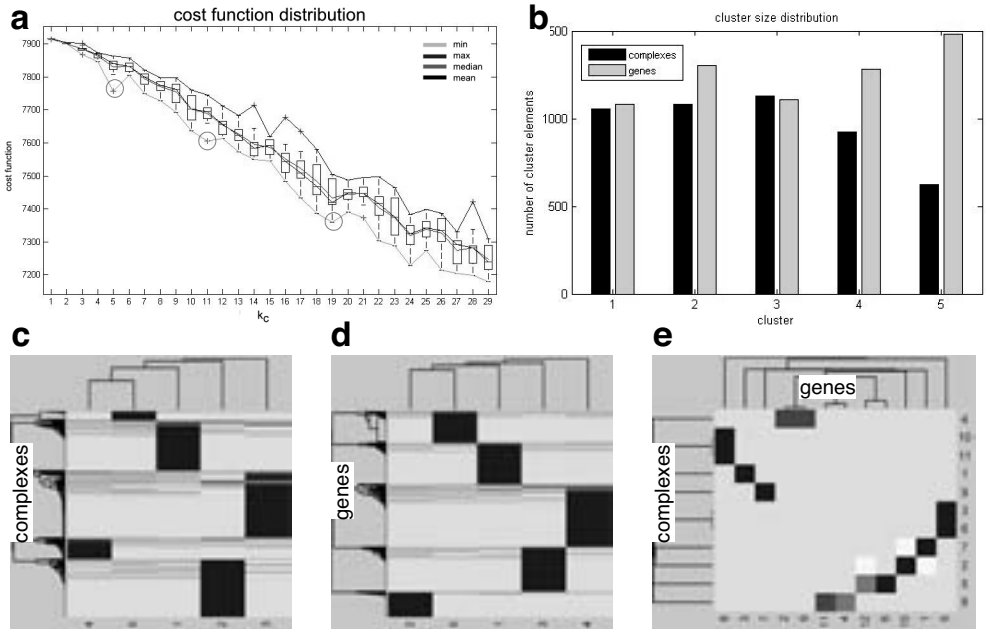

Figure 3: (a) Approximation of cluster numbers k_c, k_g. (b) Distribution of cluster sizes for $k_c = 5, k_g = 5$. Hierarchical clustering of (c) complex and (d) gene clusters (see fig 4(c) for backbone network for $k_c, k_g = 5$). The clustered backbone for $k_c = 11, k_g = 12$ is shown in (e).

tion values for the particular parameter settings. As final parameters k_c and k_g we chose $(k_c, k_g) \in \{(5,5), (11,12), (19,22)\}$, where we observe significant drops of the cost function. With this, we detect organizational structures on different levels of resolution. In the following we will mostly discuss the smallest graph with 5 clusters each (see figure 3(b)).

Figure 3 shows that our method is able to identify overlapping clusters. In the resulting five clusters, the majority of elements is assigned to a single cluster. However, there exists a considerable amount of nodes assigned to several clusters simultaneously, see figures 3(c,d). Almost ten percent of complexes (193) and genes (187) are assigned to two clusters with $p >= 0.3$. For instance, the genes *ITGB2* and *MCRS1* are even part of threes clusters with $p >= 0.3$. This clearly demonstrates the need for a fuzzy approach. The clusters vary strongly in size (figure 3(b)). and their interconnectivity is sparse, see figure 4(c). However, in the case of k_c=11 and k_g=12 we already have a resolution level that is fine enough to see details, and several binary clusters become apparent (figure 3(e)).

In order to evaluate whether both the extracted clusters and their interconnections given by the backbone graph are biologically feasible, we employed FunCat classifications. For all genes we mapped Gene Ontology associations to their according FunCat categories to achieve comparability between the node types (http://mips.gsf.de/proj/funcatDB/). Usually, complexes and genes are annotated with the lowest FunCat category or GO term respectively. In our analysis we took a subset of 13 FunCat main categories. Subcategory annotations were mapped to the according main category terms for consistency reasons.

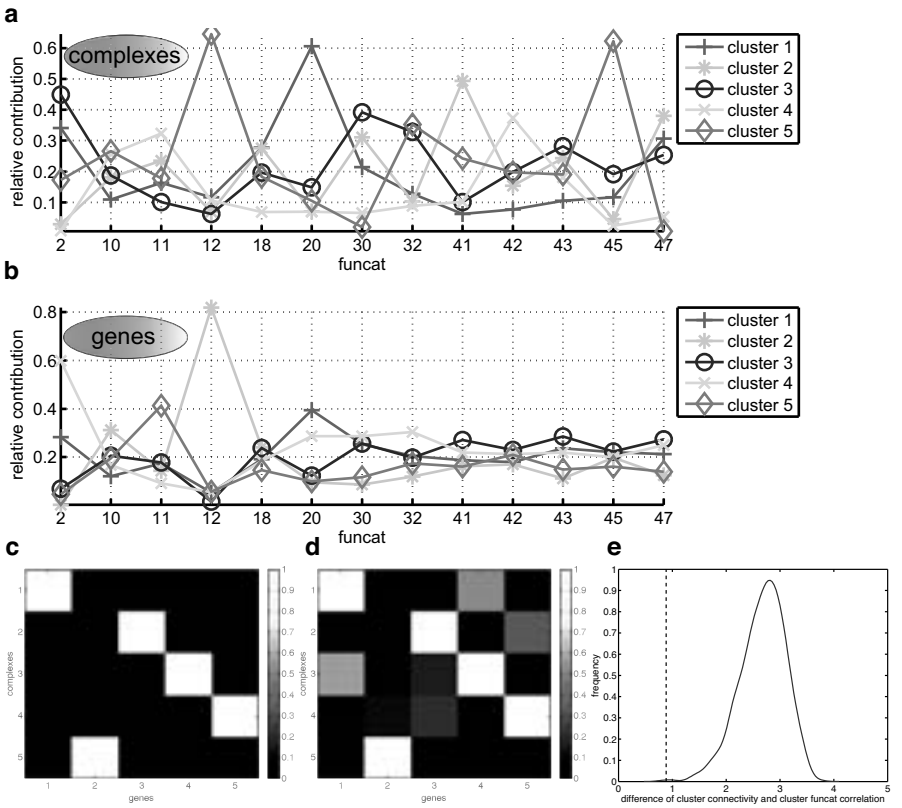

Figure 4: (a,b) FunCat annotation profile for complex and gene clusters. (c) shows the normalized backbone connectivity, and (d) the normalized positive crosscorrelations of the FunCat profiles from (a) and (b). (e) Shows statistics over 1000 random networks, proving significance of the clusters (dashed line) with a p-value of $p < 10^{-3}$.

From figure 4(a) and (b) we see that the extracted clusters can be easily interpreted biologically, as most of them have a high fraction of functional annotations with a certain FunCat term. Moreover, from visual comparison, see figure 4, we see that interconnected clusters also seem to be functionally correlated. In order to quantify this, we determined for each cluster how it is associated with each of the 13 FunCat categories by weighting a cluster elements FunCat classification by its degree of membershipto the particular cluster and calculated Pearson correlation of FunCat annotations of the complex and gene clusters. As expected, we find a high similarity between the clusters interconnectivity and their functional correlation. This shows that our fuzzy partitioning approach yields biologically meaningful results by identifying functionally related clusters.

To evaluate the significance of these results we compared our findings with the results of a random model. Assuming that a random network does not form functionally related clusters, we applied a bipartite randomization procedure to our original network. We generalized the degree-preserving rewiring for complex networks, first introduced by Maslov and Sneppen [MS02]: In every randomization step we randomly picked two edges and ex-

changed their endpoints of one type (either proteins or complexes) without creating multiple edges or self-loops. This rewiring procedure leads to a loss of degree-correlations between first and second neighbors. Hence, one can observe the degree of randomization by the course of these quantities over the process. This also tells us how many randomization steps are needed. In practice, degree-correlations vanished after around one randomization step per edge. So, for our analyses we used five times this number as in [WAH$^+$08].

We determined the clusters' FunCat profiles and calculated normalized positive correlations. To have a distance measure, we calculated the difference between the normalized backbone connectivity and the normalized positive cross-correlation matrix. Comparing these distances to clusterings using the hard approach from [LWZY06], we found much smaller values. As an example, a histogram is shown in figure 4(e), which illustrates that out of 1000 iterations only a single random entry is smaller than the 0.89, resulting in a p-value $< 10^{-3}$. This shows the significance of our results.

4 Conclusion

In this contribution, we presented a novel computationally efficient and scalable graph partitioning algorithm. Unlike other methods in the field it allows for the identification of overlapping clusters in k-partite graphs of heterogeneous data. It is based on an efficient minimization procedure, mimicking the multiplicative update rules employed in algorithms for non-negative matrix factorization. We verified our approach on a bipartite network of protein complexes where we demonstrated that we successfully identified functionally correlated clusters.

Partitioning on a local level, i.e. aiming at detecting quite small clusters, our algorithm will enable reclassification, annotation or even detection of misclassified elements in heterogeneus data sets. Partitioning into large-scale clusters, we focus on understanding their global organization. For instance, simple bipartite graph analysis has recently brought insights into the organization of microRNA interactions [RKS$^+$10]. At the moment, we extend this work by integrating predictions of microRNA target sites with protein complexes, disease information and different types of annotations.

Acknowledgements. This work was partially supported by the Helmholtz Alliance on Systems Biology (project 'CoReNe') and the Federal Ministry of Education and Research (BMBF) in its MedSys initiative (project 'SysMBO').

References

[Bar07] M.J. Barber. Modularity and community detection in bipartite networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 76(6 Pt 2):066102, Dec 2007.

[Bez81] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algoritms*. Plenum Press, New York, 1981.

[CDGS04] H. Cho, I.S. Dhillon, Y. Guan, and S. Sra. Minimum Sum Squared Residue based Co-clustering of Gene Expression data. In *Proc. SIAM International Conference on Data Mining*, pages 114–125, 2004.

[DS06] I.S. Dhillon and S. Sra. Generalized Nonnegative Matrix Approximations with Bregman Divergences. In *Proc. NIPS 2005*, 2006.

[Dun73] J.C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3:32–57, 1973.

[GL04] J. Guillaume and M. Latapy. Bipartite Structure of All Complex Networks. *Information Processing Letters*, 90(5):215–221, 2004.

[JD88] Anil K Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[KAKS97] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: application in VLSI domain. In *Proc. DAC '97*, pages 526–529. ACM Press, 1997.

[KHT09] S. Klamt, U. Haus, and F.J. Theis. Hypergraphs and cellular networks. *PLoS Computational Biology*, 5(5), 2009.

[LS99] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative Matrix Factorization. *Nature*, 40:788–791, 1999.

[LS01] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *Proc. NIPS 2000*, volume 13, pages 556–562. MIT Press, 2001.

[LWZY06] B. Long, X. Wu, Z. Zhang, and P.S. Yu. Unsupervised Learning on K-partite Graphs. In *Proc. SIGKDD 2006*, pages 317–326, 2006.

[Mac67] J. B. MacQueen. Some Methods for Classification and Analysis of MultiVariate Observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[MS02] Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, May 2002.

[PDFV05] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, Jun 2005.

[RBDK+08] A. Ruepp, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, M. Stransky, B. Waegele, T. Schmidt, O. Noubibou Doudieu, V. Stümpflen, and H.W. Mewes. CORUM: the comprehensive resource of mammalian protein complexes. *Nucleic Acids Res*, 36(Database issue):D646–D650, Jan 2008.

[RKS+10] Andreas Ruepp, Andreas Kowarsch, Daniel Schmidl, Felix Buggenthin, Barbara Brauner, Irmtraud Dunger, Gisela Fobo, Goar Frishman, Corinna Montrone, and Fabian J. Theis. PhenomiR: a knowledgebase for microRNA expression in diseases and biological processes. *Genome biology*, 11(1):R6+, January 2010.

[WAH+08] P Wong, S Althammer, A Hildebrand, A Kirschner, P Pagel, B Geissler, P Smialowski, F Bloechl, M Oesterheld, T Schmidt, N Strack, FJ Theis, A Ruepp, and D Frishman. An evolutionary and structural characterization of mammalian protein complex organization. *BMC Genomics*, 9(1):629, Dec 2008.

[ZHS07] D. Zhou, J. Huang, and B. Schoelkopf. Learning with Hypergraphs: Clustering, Classification, and Embedding. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.