BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

# Structuring heterogeneous biological information using fuzzy clustering of k-partite graphs

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

## Abstract

**Background:** Extensive and automated data integration in bioinformatics facilitates the construction of large, complex biological networks. However, the challenge lies in the interpretation of these networks. While most research focuses on the unipartite or bipartite case, we address the more general but common situation of *k*-partite graphs. These graphs contain *k* different node types and links are only allowed between nodes of different types. In order to reveal their structural organization and describe the contained information in a more coarse-grained fashion, we ask how to detect clusters within each node type.

**Results:** Since entities in biological networks regularly have more than one function and hence participate in more than one cluster, we developed a *k*-partite graph partitioning algorithm that allows for overlapping (fuzzy) clusters. It determines for each node a degree of membership to each cluster. Moreover, the algorithm estimates a weighted *k*-partite graph that connects the extracted clusters. Our method is fast and efficient, mimicking the multiplicative update rules commonly employed in algorithms for non-negative matrix factorization. It facilitates the decomposition of networks on a chosen scale and therefore allows for analysis and interpretation of structures on various resolution levels. Applying our algorithm to a tripartite disease-gene-protein complex network, we were able to structure this graph on a large scale into clusters that are functionally correlated and biologically meaningful. Locally, smaller clusters enabled reclassification or annotation of the clusters' elements. We exemplified this for the transcription factor MECP2.

**Conclusions:** In order to cope with the overwhelming amount of information available from biomedical literature, we need to tackle the challenge of finding structures in large networks with nodes of multiple types. To this end, we presented a novel fuzzy *k*-partite graph partitioning algorithm that allows the decomposition of these objects in a comprehensive fashion. We validated our approach both on artificial and real-world data. It is readily applicable to any further problem.

## Background

With the increasing availability of high throughput "-omics" technologies such as next generation sequencing, proteomics or metabolic profiling an enormous amount of textual data is accessible in the biomedical literature. Hence, methods able to structure these heterogeneous data and to extract new knowledge gain more and more importance. Learning approaches often focus on the analysis of homogeneous data sets that can be represented as graphs having vertices of a single type only. However, biological networks are complex and highly diverse and therefore often involve objects of multiple types, forming *k*-partite graphs consisting of different kinds of vertices. We use this representation as it provides a more comprehensive picture of the underlying structure compared to the widely used graph transformations. These so-called projections - e.g. of a bipartite network into an unipartite version - discard important information [1]. For instance, [2] shows that in the case of metabolism the use of projections leads to wrong interpretations of some of the most relevant graph attributes, whereas the bipartite view offers a cleaner interpretation of its topological features.

* Correspondence: mara.hartsperger@helmholtz-muenchen.de; fabian.theis@helmholtz-muenchen.de
† Contributed equally
[1]Institute of Bioinformatics and Systems Biology (MIPS), Helmholtz Zentrum München - German Research Center for Environmental Health, Ingolstädter Landstr. 1, 85764 Neuherberg, Germany
Full list of author information is available at the end of the article

The human disease network presented in [3] is an example for a bipartite graph having two disjoint sets of vertices. Here, structural questions need to be addressed outside of the unipartite graph setting. One set of nodes represents all known genetic disorders, the vertices of the other partition correspond to all known disease genes in the human genome. A disorder and a gene are connected if mutations in that gene are implicated in that disorder. Other examples of bipartite networks are protein complex or gene-localization, gene-function or microRNA-target networks. The integration of such network data then leads to complex $k$-partite graphs.

A key question is how to interpret the internal organization of these networks. A possible answer may be a modular decomposition, which implies the coexistence of structural subunits associated with more highly interconnected parts. We regard the identification of these a priori unknown building blocks - such as for instance functional modules in protein-protein-interaction (PPI) networks - as clustering methods. The clusters and their interconnections are essential for understanding the underlying functional properties. They structure biological data by compressing their information into a condensed form.

Most available clustering methods do not treat the single node types (partitions) separately and therefore do not represent the global cluster structure of $k$-partite networks correctly. While this has been addressed in terms of algorithms for some time now [4-6], not many applications were successfully implemented in bioinformatics yet, although the field commonly deals with such networks [1]. A particular issue that may hamper application to biological data is that most existing algorithms identify separated, disjoint clusters by assigning each point to exactly one cluster [7,8]. This is unrealistic for biological systems as e.g. genes or proteins commonly participate in multiple processes or pathways [9]. So far, only a few approaches exist that allow the detection of overlapping clusters. These either assign each data point to several hard clusters [10] or determine a degree of membership to each cluster [11,12]. Such methods are known as fuzzy clustering, but have not been applied to the common biological case of $k$-partite graphs.

To overcome these difficulties we developed a novel fuzzy clustering algorithm based on a non-negative matrix factorization (NMF) model [13]. Our algorithm extends a hard clustering algorithm recently put forward in [14]. This algorithm clusters each node type of the graph separately and then connects clusters via a smaller, weighted $k$-partite graph in an alternating minimization procedure. Thereby, the cluster assignment in the first step is made in a binary fashion. This disjoint clustering is a feature that is often achieved by soft clustering algorithms when not forcing explicit cluster overlap [11]. However, it can be easily seen that the cost function proposed is not fully minimized. Our computationally efficient and scalable algorithm avoids this problem. It is similar in structure to multiplicative algorithms for NMF, with the difference that we address a three-matrix factorization problem (see e.g. [15]), and have to deal with a multi-summand cost function. As our cost function is monotonous with respect to the number of clusters, our algorithm allows detection of clusters on different scales. Hence, we are able to decompose the network on different resolution levels.

The manuscript is organized as follows. In the first part, we develop the fuzzy clustering algorithm and validate it on a toy example and graphs with known modular structure. Then, we apply it to a tripartite disease-gene-protein complex graph representing an expanded view of the human disease network from [3] extended by protein complexes [16]. By integrating functional annotation we demonstrate that we are able to structure this complex graph into biologically meaningful clusters on a large scale. Finally, focusing on the small-scale architecture, we identify overlapping clusters that give a more comprehensive picture about gene-disease connections rather than looking at disjoint clusters alone. We exemplify how this clustering allows for reclassification, annotation or even detection of misclassified elements on a local level.

## Results and Discussion

A $k$-partite graph is a graph $G = (V, E)$ of edges $E$ between a set of vertices $V$ together with a partition of the vertices into $k$ disjoint subsets $V_i$ such that no two vertices in the same subset are adjacent. For $k = 1$ this reduces to the standard graph, where we do not take into account different node types. Graphs with two partitions are called bipartite. Let $n_i := |V_i|$ be the number of vertices in the partition $i$, $i = 1 \dots k$. We represent the graph as a set of $n_i \times n_j$-dimensional adjacency matrices $\mathbf{A}^{(ij)}$ for all $i, j$ with $1 \le i < j \le k$. Typically, 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.

### Approach

We want to approximate $G$ by a smaller $k$-partite cluster network $H$ which we call *backbone network*. It is defined on the fuzzy clusters of each $G$-partition $V_i$. We fix the number of clusters in partition $i$ to $m_i$. We denote a non-negative $n_i \times m_i$-dimensional matrix $\mathbf{C}^{(i)}$ to be a *fuzzy clustering* of $V_i$, if it is right-stochastic, i.e. $\sum_{l=1}^{m_i} \mathbf{C}_{kl}^{(i)} = 1$ for all $k$. Its $(k, l)$-th element $\mathbf{C}_{kl}^{(i)}$ gives the degree of membership of the original node $k$ to the backbone node $l$.

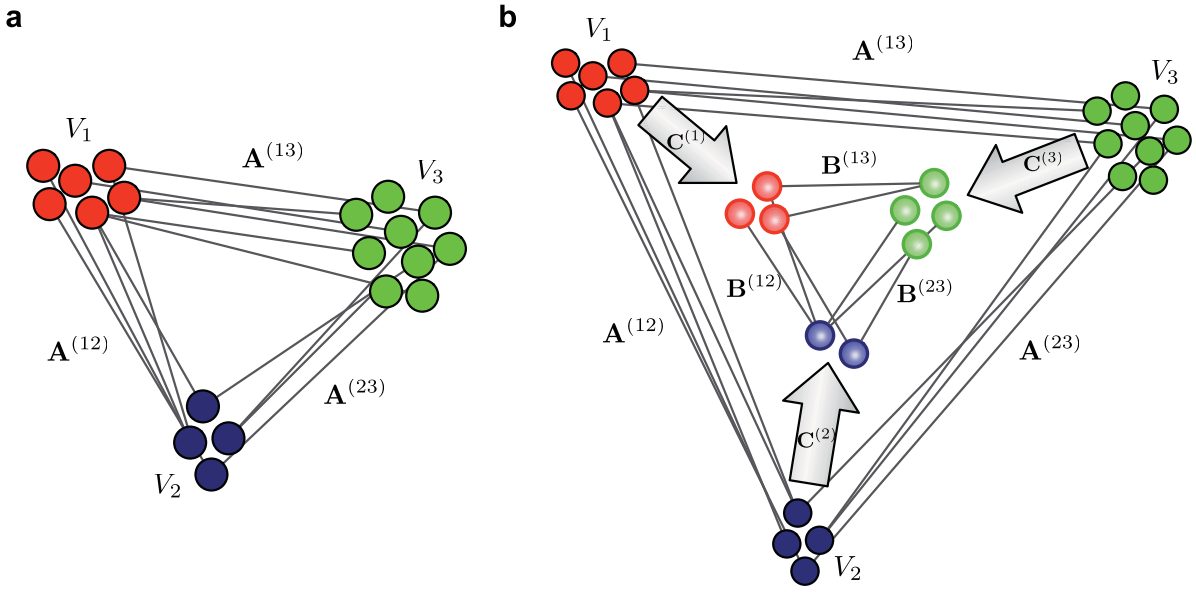


**Figure 1 Illustration of the fuzzy clustering approach**. We want to approximate the tripartite example graph $G$ in **(a)** by a smaller tripartite cluster network $H$, the so-called *backbone graph* **(b)**. The decomposition into fuzzy clusters connected by this backbone must explain the original connectivity as good as possible. The edges of $G$ are collected in adjacency matrices $\mathbf{A}^{(ij)}$ connecting the elements of the partitions $i$ and $j$. The approximation of $G$ by the backbone graph is encoded in the adjacency matrices $\mathbf{B}^{(ij)}$ connecting the fuzzy node clusters $\mathbf{C}^{(i)}$. These matrices $\mathbf{C}^{(i)}$ collect the degrees of membership of each node of partition $V_i$ to each cluster of this type. Its $(k, l)$-th element $\mathbf{C}^{(i)}_{kl}$ specifies how much node $k$ belongs to the backbone node $l$.

Then we search for a $k$-partite graph $H$ with $m_i \times m_j$ adjacency matrices $\mathbf{B}^{(ij)}$ and a fuzzy clustering $C := (\mathbf{C}^{(i)})_{i=1,\dots,k}$ such that the connectivity explained by $H$ is as close as possible to $G$ after clustering according to $C$. Figure 1 shows an example graph and its approximation by a backbone network. From the approximation, we can easily reconstruct an edge $\mathbf{A}^{(ij)}_{uv}$ between two nodes $u$ and $v$ from partitions $i$ and $j$ in the original graph. To this end, we have to sum up all edge weights $\mathbf{B}^{(ij)}$ in the backbone graph that connect the communities $u$ and $v$ are assigned to. Of course, in a fuzzy environment these contributions have to be weighted by the nodes' degrees of membership $\mathbf{C}^{(i)}$ and $\mathbf{C}^{(j)}$, respectively. Taken together, the entry of the adjacency matrix can be reconstructed as the double sum

$$\mathbf{A}^{(ij)}_{uv} \approx \sum_{x=1}^{m_i}\sum_{y=1}^{m_j} \mathbf{C}^{(i)}_{ux}\mathbf{B}^{(ij)}_{xy}\mathbf{C}^{(j)}_{vy}.$$

Writing this in matrix notation, we see that the requirement of explaining maximum possible connectivity means that the adjacency matrices $\mathbf{A}^{(ij)}$ are best possible approximated by factorizations of the form

$$\mathbf{A}^{(ij)} \approx \mathbf{C}^{(i)}\mathbf{B}^{(ij)}(\mathbf{C}^{(j)})^{\top}.$$

We can measure the difference between the two graphs $H$ and $G$ in a variety of ways. In [14], this choice has been circumvented by focusing on arbitrary Bregman divergences, see e.g. [17], 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 by [15]. Here, we choose the minimum square distance as the cost function. This implies minimization of

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

where $\|.\|_F^2$ denotes the squared Frobenius norm i.e. the square sum of the matrix elements. This cost function is obviously monotonous with respect to the number of clusters in each partition.

### *Algorithm formulation and relation to other work*

We aim at minimizing the cost function $f(H, C)$ using a local algorithm extending gradient descent. In order to avoid the choice of update rates and to ensure positivity of both the backbone network and the degrees of membership of all nodes, we employ multiplicative update rules. This strategy is widely used in algorithms for non-negative matrix factorization (NMF) [18]. We find the

following update rules (see Methods for the detailed derivation):

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

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

We note that these update rules do not increase the cost function (1). This can be shown via auxiliary functions similar to NMF [18] and multi-factor NMF [15], which has been applied in a related model for co-clustering of microarray data [19]. This theoretical result implies convergences of the update rules. However, in contrast to early statements in NMF [18], it does not necessarily imply convergence 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. This is one of the reasons, why sometimes alternating least-squares algorithms are chosen [20].

We have not yet taken into account the constraint that the fuzzy clusterings $\mathbf{C}^{(i)}$ are required to be right-stochastic. We force this constraint by regularly projecting each row of $\mathbf{C}^{(i)}$ onto the sphere of the 1-norm. The final fuzzy $k$-partite clustering algorithm is summarized in Figure 2. Implementations are available as Additional Files 1 and 2.

Our algorithm has two nested loops over the number of partitions $k$, hence its runtime depends quadratically on this number. The update rules for $\mathbf{C}^{(i)}$ and $\mathbf{B}^{(ij)}$ however are fully vectorized and contain only matrix operations with non-square matrices. Their time complexity is dominated by the occurring matrix products: multiplying two matrices of sizes $s_1 \times s_2$ and $s_2 \times s_3$ is of complexity $\mathcal{O}(s_1 s_2 s_3)$. Assuming that the cluster numbers $m_i$ are smaller than the largest two partition sizes, the total time complexity of the fuzzy clustering algorithm can then be estimated as

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

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 within any partition. Hence, the runtime grows only quadratically in the total number of nodes in the case of graphs with similarly large partitions. In general, the runtime is linear in each partition's size $n_i$ and cluster number $m_i$. We additionally confirmed this theoretical analysis by simulations shown in Additional File 3.

#### *Algorithm evaluation - Toy example*

For illustration, we applied our algorithm to a bipartite graph having several vertices connected with all vertices of the other partition (e.g. nodes 1 and 10). Figure 3 shows that these vertices are assigned to two clusters with distinct degree of membership, whereas vertices partially connected are element of a single cluster only (e.g. 3). This demonstrates the idea and importance of using a fuzzy that allows for overlapping clusters.

#### *Algorithm evaluation - Performance analysis*

Before applying our algorithm to real-world data, we tested its behavior on simulated data with controlled cluster structure. In particular, we compared it to the hard clustering algorithm from [14]. We used exactly the same stopping criteria for both algorithms. We generated random, modularly structured $k$-partite networks as described in the Methods section. In order to compare algorithm performance, we determined runtime, final cost function value and the quality of cluster estimation in four different settings. We restricted ourselves to bipartite and layered tripartite graphs with two different noise settings because Long *et al.* provided code for analyzing these special cases only.

We found that while the method of Long *et al.* performed around two times faster, our algorithm produced around 10% lower cost function and was able to estimate the cluster structure better (see Figure 4). This difference in algorithm runtime originates from the much more fine-tuning of the continuous degrees of membership compared to hard cluster assignments. These require less update steps until convergence.

#### *Algorithm evaluation - Stability of clusters*

In contrast to deterministic methods like for instance singular value decomposition (SVD), NMF-based methods have problems concerning robust computation. Even for standard unipartite NMF there is no unique global minimum of the cost function [21]. Our algorithm aims to minimize the cost function using a local optimization strategy extending gradient descent. This implies that the algorithm only converges to a local minimum. The algorithm is indeterministic, it does not converge to the same solution on each run due to the stochastic nature of initial conditions. Thus, following the general proceeding in literature on NMF [21,22], we compare the local minima from several different starting points (multiple restarts), using the results of the best local minimum found.

In order to illustrate the stability of the fuzzy clustering algorithm we applied it to a toy network with well

**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 element-wise multiplication and division, respectively.*

**Figure 2 Fuzzy clustering algorithm**. Summarization of the final fuzzy $k$-partite clustering algorithm.

defined cluster structure using multiple restarts. We compared the clustering results of 100 runs and quantified the cluster stability using a fuzzy variant of the rand index recently proposed in [23]. As we show in Additional File 4, our algorithm is able to reproduce the true clustering results in more than 70% of the runs. Hence, we can not guarantee that the local optimization finds a global minimum of the cost function, and with this the cluster structure of a graph. This illustrates the critical need for multiple restarts.

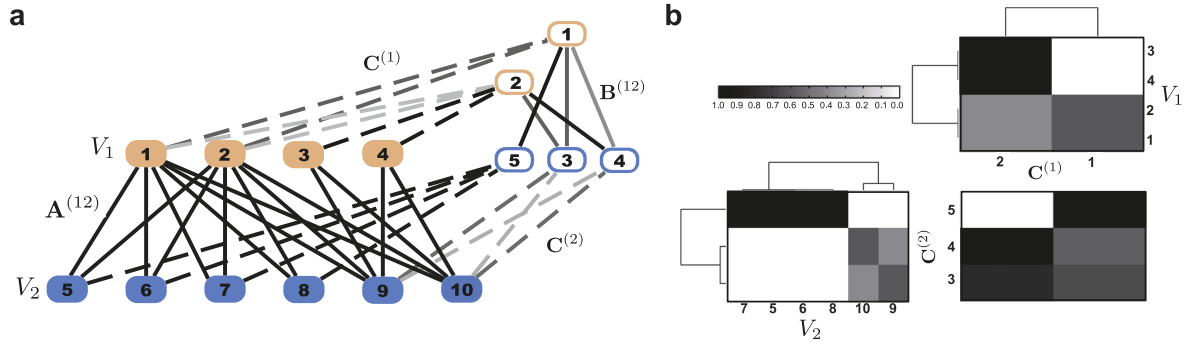


**Figure 3 Illustration of the cluster decomposition of a bipartite toy example**. **(a)** We demonstrate the graph decomposition with our algorithm on a small bipartite graph with overlapping cluster structure. The original graph consists of partitions $V_1 = \{1 \ldots 4\}$ (red filled nodes) and $V_2 = \{5 \ldots 10\}$ (blue filled nodes) connected by edges $\mathbf{A}^{(12)}$ colored in black. We decomposed it into two clusters for partition $V_1$ and three clusters for partition $V_2$. The resulting fuzzy clustering is illustrated as a weighted graph connecting original nodes to cluster nodes (framed red and blue). The cluster assignments $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are indicated by dashed lines, where the coloring corresponds to the degree of cluster membership. The interconnections of the clusters form the *backbone graph*, encoded in the adjacency matrix $\mathbf{B}^{(12)}$ which we denote by continous lines where color indicates the edge weight. Another way of illustrating the graph decomposition is shown in **(b)**. It is clearer especially for larger graphs. First, we plot hierarchical clusterings of the nodes' degrees of membership in partitions $V_1$ and $V_2$ (encoded by $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$). This facilitates the identification of overlapping clusters (e.g. nodes 1 and 10 are assigned to more than one cluster) or hard cluster assignments (e.g. node 5). The backbone graph $\mathbf{B}^{(12)}$ is shown bottom right. This backbone graph is densely connected in our example.

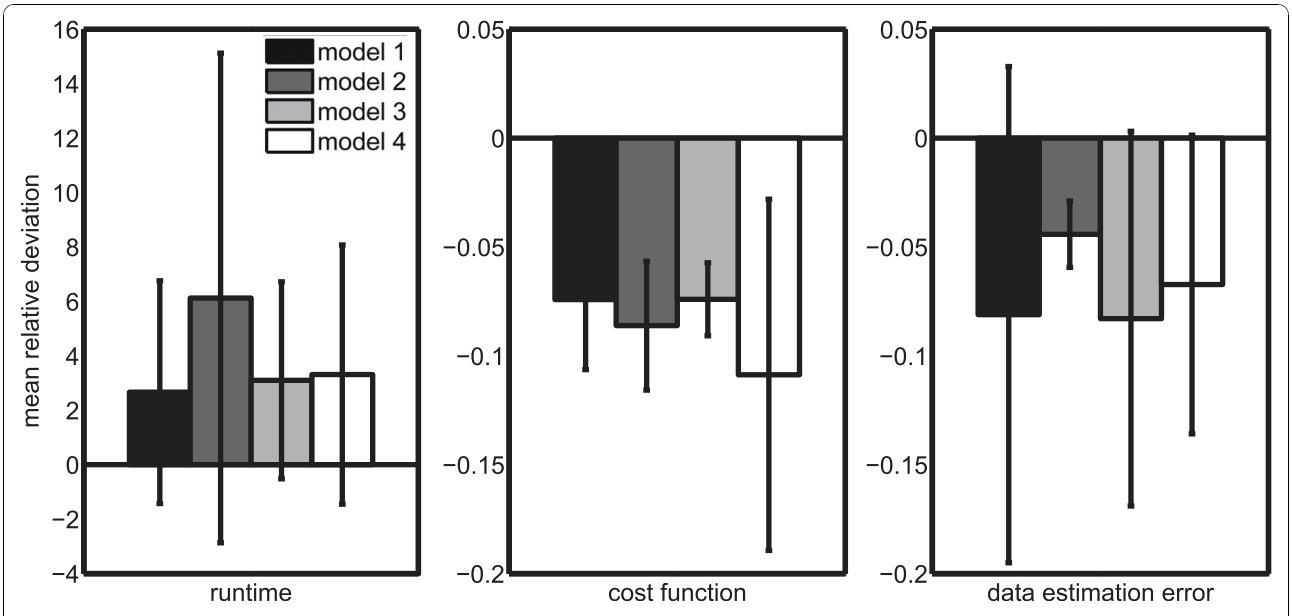


**Figure 4 Performance on toy models**. We validated our algorithm on graphs with predefined cluster structure. To this end, we compared it with the hard clustering method by Long *et al.* on four different random toy models, see Table 1. The plot shows the mean relative deviation between the two algorithms relative to the results of the hard clustering. Error bars denote standard deviations over 1000 runs. We see that the fuzzy cluster assignments of our method require much more runtime, but both cost function and data estimation error (see Methods) are significantly smaller. The large standard deviations show the dependency of the decomposition on the random initial conditions. Therefore, by default we perform multiple restarts with different initializations.

### Structuring biological data

In order to exemplify the analysis of biological networks, we applied our algorithm to a layered tripartite disease-gene-protein complex network, see Figure 5a for an illustration. In this graph, a disorder and a gene are connected if mutations in that gene are implicated in that disorder. A complex and a gene are linked if the gene is coding for a protein part of the complex. We constructed this graph by integrating the human gene-disease network from [3] and protein complexes from the CORUM database, as explained in the Methods section.

An important feature of many biological networks is their hierarchical organization, where higher-level structure is composed of multiple instances of a lower-level structures of different types [24]. This implies that small groups of nodes organize in a hierarchical manner to increasingly large groups on many different scales [25,26]. To account for this topological characteristic we have to be able to extract relevant information on an appropriate, pre-defined resolution level.

We addressed this issue by analyzing the very global structure and a detailed local level of the disease-gene-protein complex network. In the following, we first present the results of a decomposition into large clusters which demonstrates that our method is generally applicable to biological data. Then, we discuss smaller clusters that allowed for a precise interpretation of single elements.

As discussed before, due to its random initialization our algorithm is inherently indeterministic. Different clustering results have of course a significant impact on the interpretation of the biological meaning of the results. We show in Additional File 4 that our algorithm is quite stable on graphs with well defined cluster structure. To avoid analyzing a local minimum, we discuss performance over 10 runs and verify that the disease-gene-protein complex network has indeed a defined cluster structure in Additional File 5.

Dealing with a theoretically monotonous cost function, it is hard to determine the optimal numbers of clusters for each node type in which the graph has to be partitioned. Appropriate values are not apparent from prior knowledge about our data set. We therefore chose desired approximate resolutions $m_g$ for the gene partition. The number of clusters $m_c$ and $m_d$ in the protein complex and disease partitions were then scaled according to their partitions' sizes (see Methods). We use this heuristics, since a brute-force sampling of the three-dimensional parameter space is computationally out of reach. Then, we looked for plateaus and steep drops in the cost function within a certain range around this value $m_g$ and chose a local optimum of the algorithmically found decompositions. In Additional File 5 we performed simulations showing that the profile of the cost function may indeed indicate for a proper number of clusters in graphs with known cluster structure.

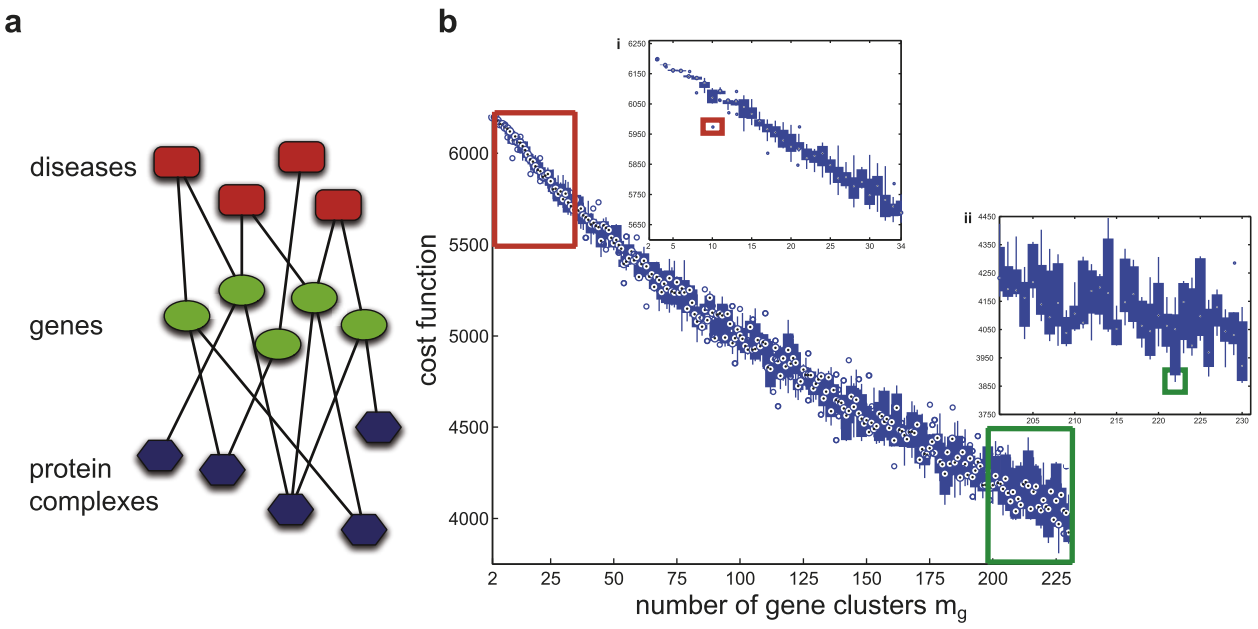


**Figure 5 Decomposition of a gene-disease-protein complex network**. We integrated the gene-disease network from [3] with human protein complexes from the CORUM database [16]. This resulted in a layered tripartite graph, which is schematically drawn in **(a)**. We performed a 10-fold approximation of this graph to estimate appropriate numbers of clusters. The boxplot curve **(b)** shows how the cost function $f(H, C)$ from equation (1) depends on the number of gene clusters $m_g$. The true minima of the cost function are decreasing with $m_g$, and this is also visible in the approximated minima using our proposed algorithm. Therefore, we are able to identify structures on various resolution levels. The details represent the cost function course for large-scale clustering **(i)** and a decomposition on small scale **(ii)**, respectively. For our detailed analyses, we used the decompositions showing steep drops in the cost function marked by the red and green boxes.

### *Large-scale clustering*

First, we focused on the identification of large clusters. Figure 5b shows the distribution of the cost function values after algorithm convergence for each parameter setting. In the following discussion, we used ($m_g$, $m_c$, $m_d$) = (10, 5, 6) as it showed the first steep drop in the cost function. Moreover, here we observed a significant local minimum of the cost function values of the algorithmically determined decompositions. From the illustration of the decomposition in Figure 6 we see that the resulting clusters vary strongly in size. For all partitions, the majority of elements was assigned to a single cluster with degree of membership $\mu >= 0.9$. This demonstrates that the analyzed graph has a well defined cluster structure at the desired resolution level. The corresponding histograms are given in Additional File 5. Therein we also discuss an example illustrating that such large degrees of membership are rarely found in graphs lacking any cluster structure. However, there exists also a considerable amount of elements assigned to several clusters simultaneously, e.g. in complex clusters 3 and 5, gene clusters 1 and 3 or disease clusters 3 and 4. This confirms the usefulness of our fuzzy approach.

***Cluster evaluation*** To determine whether the resulting clusters are biologically reasonable, we applied GO enrichment analysis (see Methods) to the clusters of the gene partition. We found, for instance, that for the genes in the two overlapping clusters 1 and 3 significantly enriched GO-terms are *cell cycle* and *cellular response to stimulus/stress*. Genes in cluster 4 can be related to e.g. *death*, *cell proliferation* and *developmental processes*, whereas cluster 6 represents *translation*. *Gene expression*-associated GO-terms such as *RNA processing* and *splicing* were detected in cluster 7. This shows that our method was able to identify biologically meaningful functionally enriched clusters. The complete tables for GO enrichments in all clusters are shown in Additional File 6.

The interconnectivity of the in total 21 clusters is sparse (see Figure 6). The skeleton for the global cluster structure for both underlying bipartite graphs (gene-complex and gene-disease) demonstrates that locally overlapping clusters also tend to interconnect with the same clusters of the other partition; for instance, disease clusters 3 and 4 are both connected with gene cluster 9. To evaluate the extracted backbone graph, in the following we tested the hypothesis that interconnected clusters of different partitions are also functionally correlated.

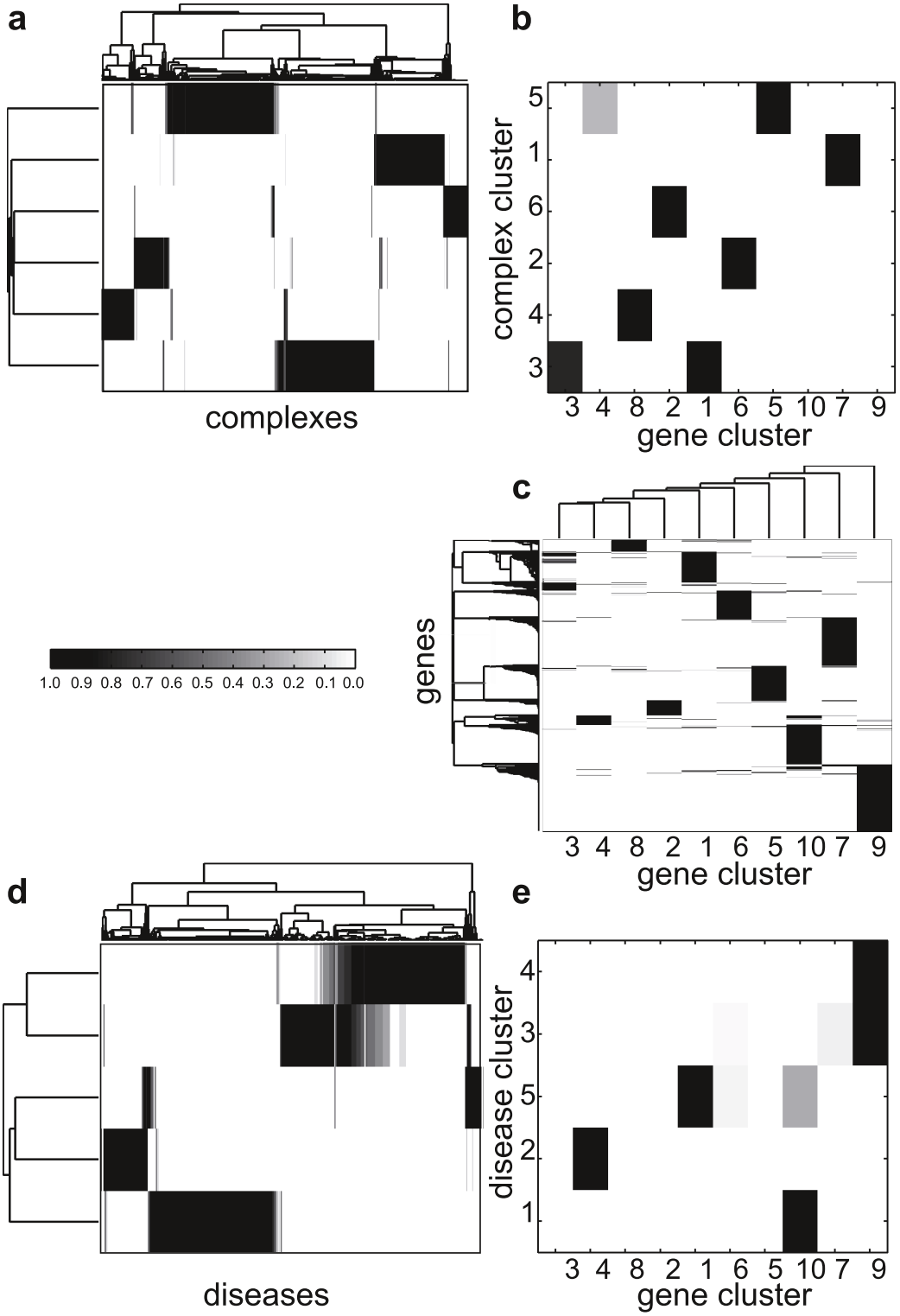


**Figure 6 Illustration of large-scale cluster structures in the gene-disease-protein complex network**. The large-scale decomposition of the gene-disease-protein complex network is illustrated as described in Figure 2b. The hierarchical clustering of the nodes' degrees of membership of the **(a)** complex, **(c)** gene and the **(d)** disease partition show that the majority of elements was assigned to single clusters. However, a considerable amount of cluster overlaps exists, e.g. for the disease clusters 3 and 4. The backbones for gene-complex **(b)** and for gene-disease **(e)** are sparsely connected, but show that locally overlapping clusters tend to interconnect with the same clusters of the other partition; e.g. disease cluster 3 and 4 are both connected to gene cluster 9 with large weights.

***Gene-complex interconnections*** Assuming that the resulting interconnected gene and complex clusters are functionally related, one expects to see a similar profile for FunCat annotation and backbone interconnectivity of each cluster. This hypothesis was verified in Figure 7, where for instance complex cluster 3 and the interconnected gene clusters 1 and 3 show a high binary FunCat correlation. The difference score (as defined in Methods) between backbone interconnectivity and annotation correlation is 2.48, resulting in a $p$-value < $10^{-5}$. To compare the results of the fuzzy clustering approach with the results for the disjoint clustering method from [14] we applied the algorithm with the same parameter settings and identical annotation and randomization procedure to the obtained clusters. For hard clustering we achieved a larger difference score of 2.99 which corresponds to a significant $p$-value of 0.0015.

***Gene-disease interconnections*** To ascertain that our method is able to detect biological feasible clusters in all partitions, we determined for each gene and disease cluster disorder class profiles. Again, we observed a high similarity between backbone interconnectivity and disorder correlation having a difference score of 1.09 ($p$-value < $10^{-5}$). For instance, gene cluster 1 and 10 and the interconnected disease clusters 1 and 5 show a high disorder correlation (see Figure 7).

### *Small-scale clustering*

We showed that our method is able to both detect and interconnect biologically meaningful clusters. However, due to their size of about 279 genes on average the single clusters are hard-to-interpret. The detection of smaller clusters representing biological units enables a precise biological interpretation. In the following, we describe results for ($m_g$, $m_c$, $m_d$) = (222, 135, 112),

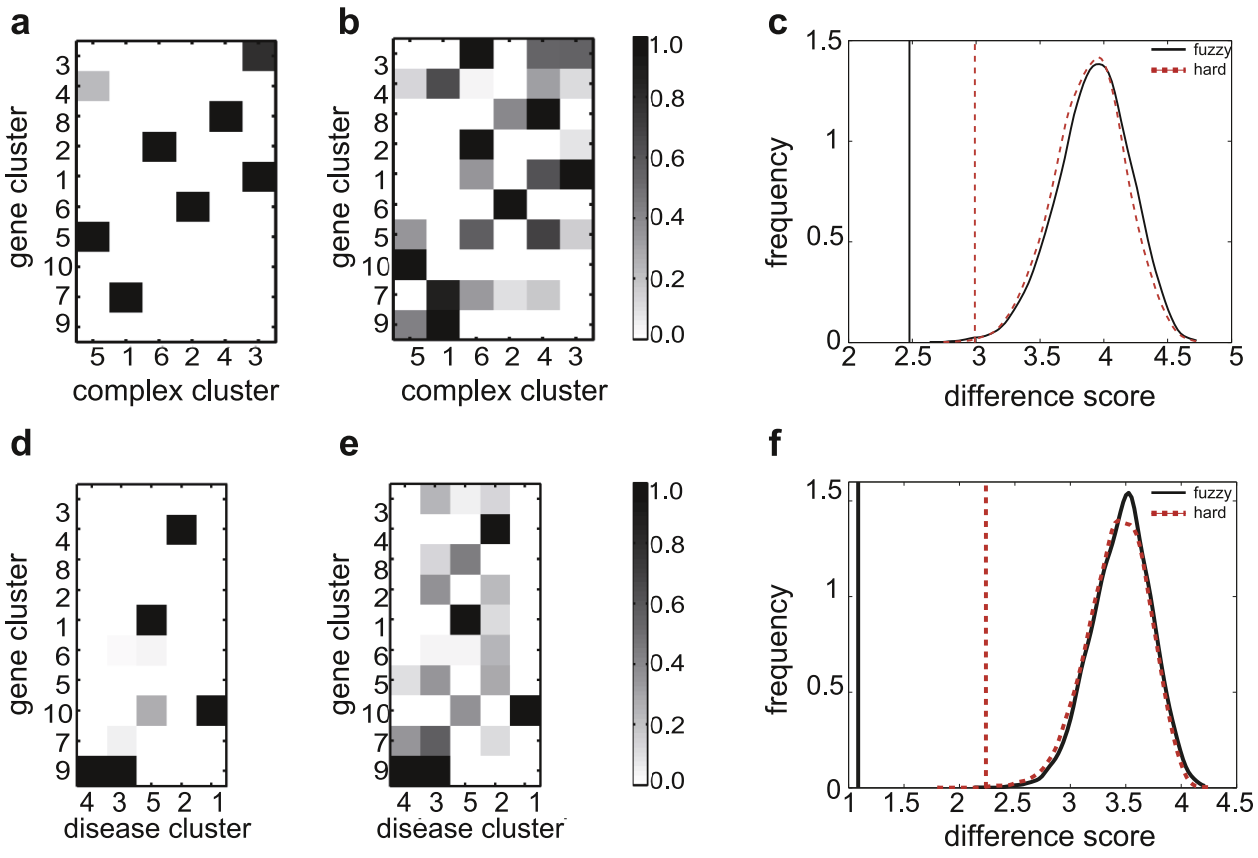


**Figure 7 Evaluation of the backbone of the gene-disease-protein complex network**. To evaluate the large-scale clustering we additionally included functional annotations. **(a)** and **(b)** compare the gene-complex backbone graph with the functional correlations of the extracted clusters according to FunCat annotation. Similarly, **(d)** and **(e)** show the gene-disease backbone and the clusters' disorder class correlations (see Methods). We see that interconnected clusters also seem to correlate in their annotations. To test this hypothesis rigorously, we calculated difference scores as defined in Methods in order to quantify the correlation of the backbones and their annotations, respectively. Vertical lines in **(c)** and **(f)** correspond to these difference scores for the fuzzy (black) and the hard (red) clustering. Comparing these values to the difference scores for $10^5$ randomized cluster assignments we obtain significant $p$-values, both < $10^{-5}$. The correlations between annotations of connected clusters of the backbone is higher when applying the fuzzy approach.

where we found the lowest value of the cost function (see Figure 5b). This setting accounts for an average cluster size of 10 genes.

In order to make use of the cluster overlaps, we looked for genes assigned to more than one cluster with a degree of membership of $\mu > 0.2$. We considered this threshold as significant as it is 50-fold higher than assigning each gene uniformly to all 222 gene clusters with equal degree of membership $\mu = 0.0045$.

As a showcase we chose *MECP2*, a protein that functions as a key factor in epigenetic transcriptional regulation. It is known to be involved in neurodevelopmental and psychiatric disorders such as *Autism*, *Mental retardation* and *Angelman syndrome* [3,27,28], and was assigned to three distinct gene clusters: 25 ($\mu = 0.42$), 32 ($\mu = 0.31$), 200 ($\mu = 0.24$). These clusters mainly cover neurological (23%), psychiatric (81%) and pleiotropic (7%) genes having a degree of membership $\mu > 0.2$. This is illustrated in Figure 8, where we visualized the backbone interconnectivity and the fuzzy clustering of the nodes in the neighborhood of *MECP2*.

We then analyzed the nine disease clusters interconnected with the three gene clusters in the backbone network. In total, 45 disorders representing mainly psychiatric (66%) and neurological (20%) disorders were assigned to eight disease clusters with a degree of membership of $\mu > 0.2$; 6 out of 9 psychiatric disorders available in the network analyzed are present in three disease clusters.

Another large fraction of these disease clusters are disorders classified as *multiple*. Most of them (*Shprintzen-Goldberg syndrome* or *Aarskog Scott syndrome*) show also neurological diseases such as mental retardation [29,30]. We also identified the *ophthamological* disorder *Blepharospasm*, an adult-onset focal dystonia that causes involuntary blinking and eyelid spasms [31] for that a known polymorphism in the dopamine receptor *DRD5* is associated with [32]. This is a subform of *Dystonia* and classified as a *neurological* disorder (ICD-10 G24.5) by the WHO [33].

Furthermore, we found *Anorexia nervosa* to be present in the analyzed clusters. It is annotated as a *nutritional* disorder by [3], however it represents a life-threatening complex psychiatric disorder [34]. Another so far unclassified disease *Alcohol dependence* was assigned to the interconnected cluster. It is classified as a mental and behavioral disorder (ICD-10 F10.2) and in a broader sense can be considered as psychiatric disorder.

In contrast, applying the hard clustering algorithm, *MECP2* was assigned to a single gene cluster which is connected to two disease clusters. Although all associated disorders were identified correctly, no further information could be obtained from the clusters. However, [27] reports an epigenetic overlap in autism-spectrum neurodevelopmental disorders as *MECP2* affects the regulation of *UBE3A* expression. These relations became immediately apparent in the cluster result of our fuzzy approach: Both genes were mutually assigned to gene cluster 25 that identifies the phenotypic and genotypic overlaps, whereas direct links to known connected genes are missing in the hard clustering (see Figure 8).

## Conclusions

The widespread application of high-throughput methods such as microarrays or next generation sequencing has considerably increased the amount of experimental data and the information available in biomedical literature that is accessible to text-mining approaches [35]. These data can usually be represented in terms of networks. Over the last years, networks have emerged as an invaluable tool for describing and analyzing complex systems. However, we need to take into account that network information is commonly available for various types of nodes. Especially integrative biological networks are $k$-partite [3,36].

Another important feature of biological networks is their hierarchical organization, implying that small groups of nodes organize in a hierarchical manner to increasingly larger groups on many different scales [24-26]. This necessitates the analysis of these objects on various resolution levels. Furthermore, many proteins or genes are pleiotropic, and often associated with many functions. Hence, clustering algorithms that assign elements into several functional modules are essential [10,12,37].

We presented a novel computationally efficient and scalable graph clustering algorithm that is capable to deal with all these described issues. Further, it does not require any *a priori* knowledge about the data set. Results on a tripartite network, constructed by integrating the human disease network and protein complexes, demonstrated that we could identify and interconnect biologically meaningful clusters on different scales. Overlapping modules gave a more comprehensive picture of e.g. gene-disease connections than looking at disjoint clusters alone. Summarizing, the proposed fuzzy clustering algorithm is suitable to compress and approximate the underlying topology of heterogeneous biological networks, which facilitates the understanding of such networks on multiple scales. It is freely available and readily applicable to many further problems.

## Methods

### Derivation of the update rules

We want to minimize $f(H, C)$ in equation (1) using a local algorithm extending gradient descent. Let $\mathbf{D}^{(ij)}$: =

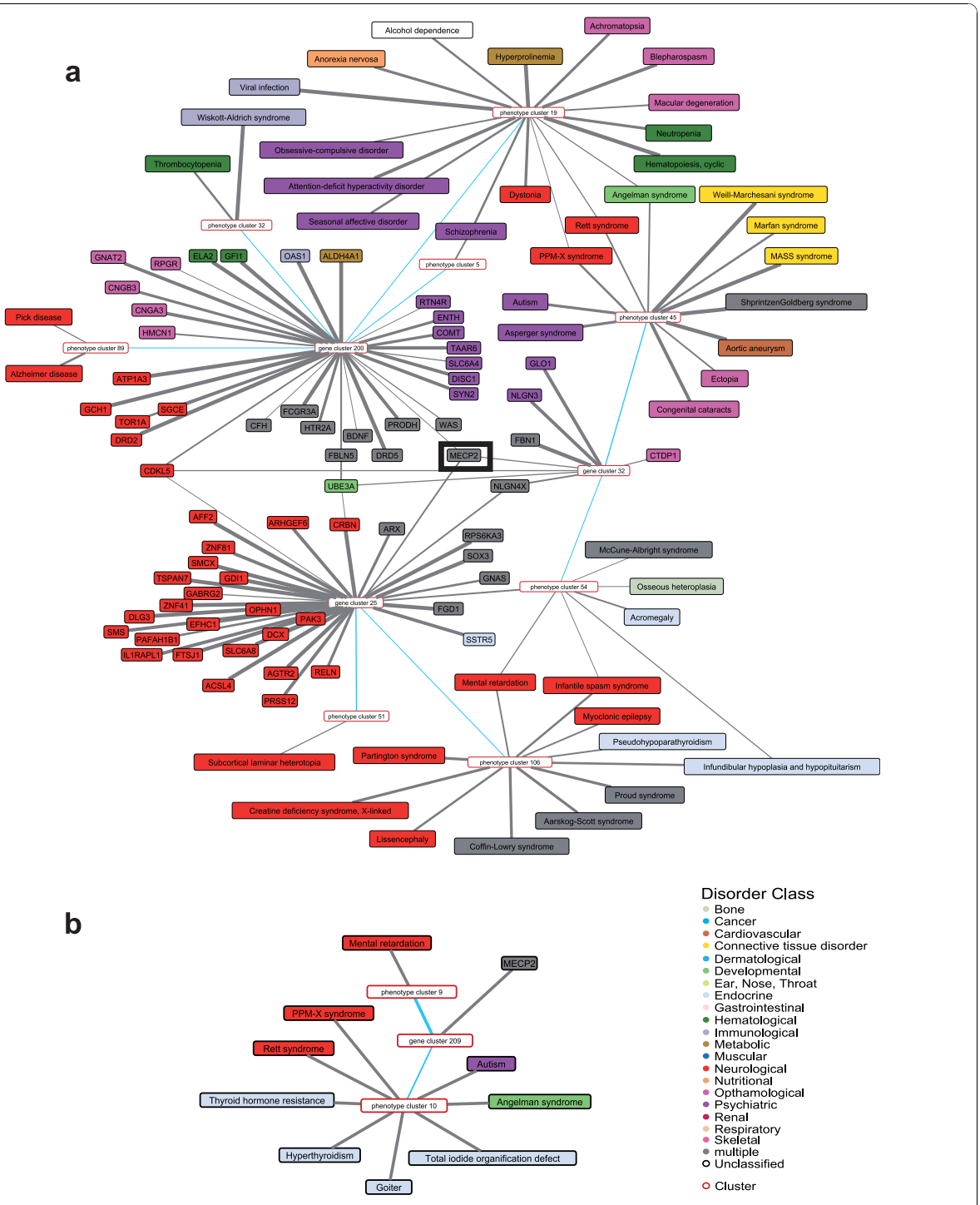


**Figure 8 The small-scale clustering in the neighborhood of MECP2**. We draw the results - the backbone network and the nodes' degrees of membership to clusters, thresholded by $\mu > 0.2$ - of the small-scale clustering in the neighborhood of *MECP2* using the fuzzy **(a)** and the hard clustering **(b)**. Nodes are colored according to their disorder class annotations. Blue edges indicate backbone interconnectivity, grey edges cluster assignment. Edge thickness indicates the degree of membership. *MECP2* is connected to three gene clusters mainly covering neurological (red) and psychiatric (purple) genes. The seven interconnected disease clusters also represent mainly psychiatric and neurological disorders. Also unclassified disorders are present such as e.g. *Alcohol dependence* (white), which is classified as a mental and behavioral disorder. In a broader sense, however, it can be considered as psychiatric disorder. Applying the hard clustering **(b)**, *MECP2* is assigned to gene cluster 209 which is connected to two disease clusters only. Although all associated disorders are identified correctly, in contrast to the fuzzy clustering no further information can be obtained from the decomposition.

$\mathbf{A}^{(ij)} - \mathbf{C}^{(i)}\mathbf{B}^{(ij)}(\mathbf{C}^{(j)})^\top$ denote the residuals, then $f = \sum_{i<j,k,l}(d_{kl}^{(ij)})^2$. Hence

$$\begin{aligned}\frac{\partial f}{\partial b_{rs}^{(ij)}} &= -2\sum_{kl} d_{kl}^{(ij)} c_{kr}^{(i)} c_{ls}^{(j)} \\ &= -2\left((\mathbf{C}^{(i)})^\top \mathbf{D}^{(ij)} \mathbf{C}^{(j)}\right)_{rs} \\ \frac{\partial f}{\partial c_{rs}^{(i)}} &= -2\sum_{j>i,k,l} d_{rl}^{(ij)} b_{sk}^{(ij)} c_{lk}^{(j)} - 2\sum_{j<i,kl} d_{kr}^{(ji)} c_{kl}^{(j)} b_{ls}^{(ji)} \\ &= -2\sum_{j>i}(\mathbf{D}^{(ij)}\mathbf{C}^{(j)}(\mathbf{B}^{(ij)})^\top)_{rs} \\ &\quad - 2\sum_{j<i}\left((\mathbf{D}^{(ji)})^\top \mathbf{C}^{(j)}\mathbf{B}^{(ji)}\right)_{rs}.\end{aligned}$$

We assume an undirected $k$-partite graph, so $\mathbf{A}^{(ij)}$ is undefined for $i > j$. For simplicity of notation, we now set $\mathbf{A}^{(ij)} := (\mathbf{A}^{(ji)})^\top$ for $i > j$ (and similarly for the $k$-partite graph $H$). Then $\mathbf{D}^{(ij)} = (\mathbf{D}^{(ji)})^\top$, and the differential simplifies to

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

Altogether, by replacing the residuals, we have shown

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

If we are to minimize $f$ by alternating gradient descent, we start from an initial guess of $\mathbf{B}^{(ij)}$, $\mathbf{C}^{(i)}$. Then, we alternate between updates of the $\mathbf{B}^{(ij)}$ and the $\mathbf{C}^{(i)}$ with learning rates $\eta_{rs}^{(ij)}$ and $\eta_{rs}^{(i)}$, respectively:

$$b_{rs}^{(ij)} \leftarrow b_{rs}^{(ij)} - \eta_{rs}^{(ij)} \frac{\partial f}{\partial b_{rs}^{(ij)}} \qquad \forall i, j : i < j$$

$$c_{rs}^{(i)} \leftarrow c_{rs}^{(i)} - \eta_{rs}^{(i)} \frac{\partial f}{\partial c_{rs}^{(i)}} \qquad \forall i$$

These update rules have two disadvantages: first, the choice of update rate $\eta$ (possibly different for $\mathbf{B}$, $\mathbf{C}$ and $i$, $j$) is unclear; in particular, for too small $\eta$ convergence may take too long or may not be achieved at all, whereas for too large $\eta$ we may easily overshoot the minimum. Moreover, the resulting matrices may become negative. Hence we follow Lee and Seung's idea for NMF [13] and rewrite this into multiplicative update rules. We therefore choose update rates

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

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

Plugging this into the gradient descent equations, we finally get:

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

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

Commonly, in order to extend cost functions in (unipartite) data clustering to include fuzzy clusters, a so-called *fuzzification factor* is introduced [11,38]]. Instead of squared norm minimization of the residuals $\mathbf{D}^{(ij)}$, a higher residual power is minimized, which results in overlapping non-trivial cluster assignments. However, we see that in our examples, already the standard case is sufficient. 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.

### Evaluation on simulated data

We built a random, modularly structured $k$-partite network as follows: We fix the number of clusters $m_i$ of nodes with color $i$, $i = 1, \ldots k$. The backbone graph is initialized by $m_i \times m_j$-matrices $\mathbf{B}^{(ij)}$ filled with zeros. We added uniformly random ones in each column according to a set percentage $\alpha$ (here on average $\alpha \geq 1$ ones in each column) such that each row has at least a single non-zero entry. In order to construct the actual network $\mathbf{A}$, we split up $\mathbf{A}^{(ij)}$ into $m_i \cdot m_j$ blocks of a fixed chosen cluster-size (here 10). We fixed a cluster connectivity $\beta$ and a random connectivity $\gamma < \beta$. Now, for each non-zero entry in $\mathbf{B}^{(ij)}$, we set the corresponding block of $\mathbf{A}^{(ij)}$ to a random Erdös-Rényi graph [39] with density $\beta$. Finally the clusters are connected by replacing each zero block of $\mathbf{A}^{(ij)}$ with an Erdös-Rényi graph of the lower connectivity $\gamma$. We analyzed 1000 realizations of four network prototypes with increasing complexity (parameters are given in Table 1). In order to compare algorithm performance, we

**Table 1 Random data models for evaluation of the fuzzy clustering algorithm**

| model | $k$ | m | $\alpha$ | $\beta$ | $\gamma$ | description |
|---|---|---|---|---|---|---|
| 1 | 2 | (3, 3) | 1 | 0.7 | 0.2 | equal-sized, no overlap |
| 2 | 2 | (3, 4) | 1 | 0.7 | 0.2 | no cluster overlap |
| 3 | 3 | (3, 4, 5) | 1.2 | 0.6 | 0.1 | 3-partite, low-noise |
| 4 | 3 | (3, 4, 5) | 1.2 | 0.8 | 0.2 | 3-partite, noisy |

Parameters for the simulated data models. $k$ denotes the number of partitions of the network, m is a vector with the number of clusters in each partition, $\alpha$ the backbone connectivity, $\beta$ the cluster and $\gamma$ the noise connectivity.

determined algorithm runtime, final cost function value and quality of cluster estimation. Cluster estimation quality was measured by the summed up Frobenius norms of the difference between the true $\mathbf{C}^{(i)}$ and the estimated $\hat{C}^{(i)}$, where clusters have been permuted such as to give minimal difference (permutation indeterminacy).

### Construction of a disease-gene-protein complex graph

We constructed a layered, tripartite graph by enlarging the human disease network [3] by all human protein complexes from the CORUM core set (as of July 2009) [16]. Integrating both data sets resulted in a graph of 5672 nodes and 7795 edges with all genetic disorders, all known disease genes and human protein complexes. We extracted the largest connected component resulting in a network with $|V|$ = 3737 and $|E|$ = 6219. It consists of 854 complexes ($V_c$), 590 diseases ($V_d$) and 2293 genes ($V_g$) (see Additional File 7).

### Parameter determination

We determined parameters for clustering on different scales. For large-scale clustering, we approximated the number of clusters to be found for each node type by limiting the maximal number of gene clusters $m_g$ for $V_g$ to $m_g = m_g = \left\lfloor \sqrt{|V_g|/2} \right\rfloor$ as suggested in [40]. The number of complex clusters $m_c$ for $V_c$ and disease clusters $m_d$ for $V_d$ were then scaled according to $m_g$ by $_i$ = $m_i = \left\lceil m_g \sqrt{|V_i|/|V_g|} \right\rceil$, where $i \in \{c, d\}$. To detect smaller clusters, we set the maximum number of gene clusters to $m_g$ for $V_g$ according to $m_g = m_g = \left\lceil \frac{|V_g|}{10} \right\rceil$. This resulted in a minimum average cluster size of 10 genes. Parameters $m_c$ and $m_d$ for $V_c$ and $V_d$ were scaled as previously.

### Cluster evaluation

We validated the gene clusters using Gene Ontology (GO) enrichment analysis. To this end, the genes used in the analysis (degree of membership $\mu > 0.2$) were tagged with their respective GO categories and analyzed within each cluster for overrepresentation of certain categories versus the "background" level of the population (in this case, all genes in the tripartite graph). We used Ontologizer [41] with the setting "Parent-Child-Intersection" restricting the analysis to the *biological process* category. For multiple testing correction we employed Bonferroni correction. To assign GO terms to gene sets, a $p$-value cutoff of 0.05 was used.

For evaluating the cluster interconnectivity we employed FunCat [42] classifications for all genes and protein complexes. We used FunCat, as Gene Ontology associations for genes could be mapped to their according FunCat categories, but not vice versa. A subset of 13 main categories was used, subcategory annotations were mapped to corresponding main category terms. Disorder classifications for genes and diseases were taken from [3], where classification classes *grey* and *multiple* were combined for pleiotropic genes (see Additional File 8). We calculated Pearson's correlation coefficients between cluster FunCat/disorder annotations by weighting a cluster element's classification by its degree of membership to the particular cluster. The difference score between normalized backbone interconnectivity and annotation correlation was determined using the Frobenius norm of their difference.

### Null model

Null models for the evaluation of the backbone graph in the large-scale clustering were generated by applying a weighted bipartite randomization procedure to each partition-cluster subgraph $\mathbf{C}^{(i)}$. To this end, we generalized the degree preserving rewiring of complex networks first introduced by [43]. In the weighted case, one has to decide between preserving either the number of neighbors of all nodes, or the total weight of their adjacent edges. We chose to maintain the first quantity: In every randomization step we randomly picked two edges and exchange their endpoints of the partition type, thereby keeping the weights attached to the edges. With this we also conserved the weighted degree of the partition nodes which reflects the right-stochasticity of the fuzzy clusterings. The degree of randomization can be monitored by a loss of degree-correlations between first and second neighbors. In practice, correlations vanish after about one randomization step per edge. So, for our analyses we used five times this number as suggested in [44]. The $p$-values were calculated over 10000 runs.

## Availability and requirements

**Project name:** Fuzzy clustering of k-partite graphs.

**Project home page:** http://cmb.helmholtz-muenchen.de/fuzzyclustering.

**Operating system(s):** Platform independent.

**Programming language:** MATLAB/Octave.

**Other requirements:** MATLAB 7.1 or higher (no additional toolboxes required) or Octave.

**License:** Free for non-commercial purposes.

## Additional material

**Additional file 1: MATLAB source code**. Fuzzy k-partite graph clustering algorithm for MATLAB.

**Additional file 2: Octave source code**. Fuzzy k-partite graph clustering algorithm for Octave.

**Additional file 3: Simulations on algorithm runtime**. Verification of the estimation of the algorithm's time complexity by simulations.

**Additional file 4: Simulation on cluster stability**. Analysis of the algorithm's stability towards the random initialization.

**Additional file 5: The chosen number of clusters**. Analysis of the cost function as an indicator for determining the number of clusters. We study the stability of the clusterings with respect to this choice and give evidence that the gene-disease-complex graph is modularly structured.

**Additional file 6: GO enrichment analysis for the gene clusters from the large-scale clustering**. Tables 1-10 show the GO (Gene Ontology) enrichment using Ontologizer [41] for the ten gene clusters in the large-scale clustering. We used only genes having a degree of membership $\mu > 0.2$ (see Methods).

**Additional file 7: Integrated tripartite network**. Illustration of the largest connected component of the layered, tripartite graph gene-disease-protein complex network. It consists of 2293 gene (green), 590 disease (red) and 854 complex (blue) nodes connected by 6219 edges.

**Additional 8: FunCat and disorder class annotation tables**. Table 1 shows the FunCat classes used for evaluating the gene and protein complex clusters. A subset of 13 FunCat main categories was taken from CORUM. Table 2 represents the 20 primary disorder classes retrieved from Goh et al. (2007). Additional classes are *multiple, grey and unclassfied*.

#### Acknowledgements
The authors thank A. Ruepp for discussions on CORUM, H.W. Mewes and P. Wong for critical reading of the manuscript and helpful comments. They also thank M. Münsterkötter for the FunCat-GO mapping, A. Kowarsch for support with the enrichment analysis and B. Long and M. Zhang for providing their hard clustering algorithm. This work was supported by the Helmholtz Alliance on Systems Biology (project CoReNe), the Federal Ministry of Education and Research (BMBF) in its MedSys initiative (project SysMBo, FKZ: 0315494A) and the TUM Graduate School for Information Science in Health (GSISH).

#### Author details
[1]Institute of Bioinformatics and Systems Biology (MIPS), Helmholtz Zentrum München - German Research Center for Environmental Health, Ingolstädter Landstr. 1, 85764 Neuherberg, Germany. [2]Department of Mathematical Science, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany.

#### Authors' contributions
The first two authors, MLH and FB, should be regarded as joint first authors. Conceived and designed the study: MLH, FB, VS, FJT. Performed the experiments: MLH, FB. Analyzed the data: MLH. Developed methods/analysis tools: MLH, FB, FJT. Wrote the paper: MLH, FB, FJT. All authors read and approved the final version of the manuscript.

Received: 12 May 2010 Accepted: 20 October 2010
Published: 20 October 2010

#### References
1. Klamt S, Haus UU, Theis F: **Hypergraphs and cellular networks.** *PLoS Comput Biol* 2009, **5(5)**:e1000385.
2. Montanez R, Medina MA, Solé RV, Rodríguez-Caso C: **When metabolism meets topology: Reconciling metabolite and reaction networks.** *Bioessays* 2010, **32(3)**:246-256.
3. Goh KI, Cusick ME, Valle D, Childs B, Vidal M, Barabási AL: **The human disease network.** *Proc Natl Acad Sci USA* 2007, **104(21)**:8685-8690.
4. Barber M: **Modularity and community detectionin bipartite networks.** *Phys Rev E Stat Nonlin Soft Matter Phys* 2007, **76(6 Pt 2)**:066102.
5. Karypis G, Aggarwal R, Kumar V, Shekhar S: **Multilevel hypergraph partitioning: application in VLSI domain.** *Proc. DAC '97* ACM Press; 1997, 526-529.
6. Zhou D, Huang J, Schoelkopf B: **Learning with Hypergraphs: Clustering, Classification, and Embedding.** *Advances in Neural Information Processing Systems 19* Cambridge, MA: MIT Press; 2007.
7. MacQueen JB: **Some Methods for Classification and Analysis of MultiVariate Observations.** In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume 1*. Edited by: Cam LML, Neyman J. University of California Press; 1967:281-297.
8. Jain AK, Dubes R: *Algorithms for Clustering Data* Prentice Hall; 1988.
9. Pereira-Leal JB, Enright AJ, Ouzounis CA: **Detection of functional modules from protein interaction networks.** *Proteins* 2004, **54**:49-57.
10. Palla G, Derenyi I, Farkas I, Vicsek T: **Uncovering the overlapping community structure of complex networks in nature and society.** *Nature* 2005, **435(7043)**:814-818.
11. Bezdek J: *Pattern Recognition with Fuzzy Objective Function Algoritms* New York Plenum Press; 1981.
12. Gasch AP, Eisen MB: **Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering.** *Genome Biol* 2002, **3(11)**, RESEARCH0059.
13. Lee D, Seung H: **Learning the parts of objects by non-negative Matrix Factorization.** *Nature* 1999, **40**:788-791.
14. Long B, Wu X, Zhang Z, Yu P: **Unsupervised Learning on K-partite Graphs.** *Proc. SIGKDD 2006* 2006, 317-326.
15. Dhillon I, Sra S: **Generalized Nonnegative Matrix Approximations with Bregman Divergences.** *Proc. NIPS 2005* 2006.
16. Ruepp A, Brauner B, Dunger-Kaltenbach I, Frishman G, Montrone C, Stransky M, Waegele B, Schmidt T, Doudieu ON, Stümpflen V, Mewes H: **CORUM: the comprehensive resource of mammalian protein complexes.** *Nucleic Acids Res* 2008, , **36 Database:** D646-D650.
17. Banerjee A, Merugu S, Dhillon I, Ghosh J: **Clustering with Bregman Divergences.** *Journal of Machine Learning Research* 2005, **6**:1705-1749.
18. Lee D, Seung H: **Algorithms for non-negative matrix factorization.** In *Proc. NIPS 2000. Volume 13*. MIT Press; 2001:556-562.
19. Cho H, Dhillon I, Guan Y, Sra S: **Minimum Sum Squared Residue based Co-clustering of Gene Expression data.** *Proc. SIAM International Conference on Data Mining* 2004, 114-125.
20. Paatero P, Tapper U: **Positive matrix factorization: A non negative factor model with optimal utilization of error estimates of data values.** *Environmetrics* 1994, **5**:111-126.
21. Langville AN, Meyer CD, Albright R: **Initializations for the Nonnegative Matrix Factorization.** *KDD 2006 Philadelphia, PA USA* 2006.
22. Devarajan K: **Nonnegative matrix factorization: an analytical and interpretive tool in computational biology.** *PLoS Comput Biol* 2008, **4(7)**: e1000029.
23. Hüllermeier E, Rifqi M: **A Fuzzy Variant of the Rand Index for Comparing Clustering Structures.** In *IFSA/EUSFLAT Conf* Edited by: Carvalho JP, Dubois D, Kaymak U, da Costa Sousa JM 2009, 1294-1298.
24. Clauset A, Moore C, Newman MEJ: **Hierarchical structure and the prediction of missing links in networks.** *Nature* 2008, **453(7191)**:98-101.
25. Ravasz E, Barabási AL: **Hierarchical organization in complex networks.** *Phys Rev E Stat Nonlin Soft Matter Phys* 2003, **67(2 Pt 2)**:026112.
26. Barabási AL, Oltvai ZN: **Network biology: understanding the cell's functional organization.** *Nat Rev Genet* 2004, **5(2)**:101-113.
27. Samaco RC, Hogart A, LaSalle JM: **Epigenetic overlap in autism-spectrum neurodevelopmental disorders: MECP2 deficiency causes reduced expression of UBE3A and GABRB3.** *Hum Mol Genet* 2005, **14(4)**:483-492.
28. Campos M, Abdalla CB, dos Santos AV, Pestana CP, dos Santos JM, Santos-Reboucas CB, Pimentel MMG: **A MECP2 mutation in a highly conserved**

aminoacid causing mental retardation in a male. *Brain Dev* 2009, **31(2)**:176-178.
29. Shprintzen RJ, Goldberg RB: **A recurrent pattern syndrome of craniosynostosis associated with arachnodactyly and abdominal hernias.** *J Craniofac Genet Dev Biol* 1982, **2**:65-74.
30. Lebel RR, May M, Pouls S, Lubs HA, Stevenson RE, Schwartz CE: **Non-syndromic X-linked mental retardation associated with a missense mutation (P312L) in the FGD1 gene.** *Clin Genet* 2002, **61(2)**:139-145.
31. Fiorio M, Tinazzi M, Scontrini A, Stanzani C, Gambarin M, Fiaschi A, Moretto G, Fabbrini G, Berardelli A: **Tactile temporal discrimination in patients with blepharospasm.** *J Neurol Neurosurg Psychiatry* 2008, **79(7)**:796-798.
32. Misbahuddin A, Placzek MR, Chaudhuri KR, Wood NW, Bhatia KP, Warner TT: **A polymorphism in the dopamine receptor DRD5 is associated with blepharospasm.** *Neurology* 2002, **58**:124-126.
33. Isaac M, Janca A, Sartorius N: *ICD-10 Symptom Glossary for Mental Disorders* World Health Organization, Division of Mental Health, Geneva 1994.
34. Sylvester CJ, Forman SF: **Clinical practice guidelines for treating restrictive eating disorder patients during medical hospitalization.** *Curr Opin Pediatr* 2008, **20(4)**:390-397.
35. Barnickel T, Weston J, Collobert R, Mewes HW, Stümpflen V: **Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts.** *PLoS One* 2009, **4(7)**:e6393.
36. Yildirim MA, Goh KI, Cusick ME, Barabási AL, Vidal M: **Drug-target network.** *Nat Biotechnol* 2007, **25(10)**:1119-1126.
37. Gulbahce N, Lehmann S: **The art of community detection.** *Bioessays* 2008, **30(10)**:934-938.
38. Dunn J: **A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters.** *Journal of Cybernetics* 1973, **3**:32-57.
39. Erdös AP, Rényi : **On Random Graphs. I.** *Publicationes Mathematicae* 1959, **6**:290-297.
40. Mardia KV, Bibby JM, Kent JT: *Multivariate analysis* Academic Press; 1979.
41. Bauer S, Grossmann S, Vingron M, Robinson PN: **Ontologizer 2.0-a multifunctional tool for GO term enrichment analysis and data exploration.** *Bioinformatics* 2008, **24(14)**:1650-1651.
42. Ruepp A, Zollner A, Maier D, Albermann K, Hani J, Mokrejs M, Tetko I, Güldener U, Mannhaupt G, Münsterkötter M, Mewes H: **The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes.** *Nucleic Acids Res* 2004, **32(18)**:5539-5545.
43. Maslov S, Sneppen K: **Specificity and stability in topology of protein networks.** *Science* 2002, **296(5569)**:910-913.
44. Wong P, Althammer S, Hildebrand A, Kirschner A, Pagel P, Geissler B, Smialowski P, Bloechl F, Oesterheld M, Schmidt T, Strack N, Theis F, Ruepp A, Frishman D: **An evolutionary and structural characterization of mammalian protein complex organization.** *BMC Genomics* 2008, **9**:629.

doi:10.1186/1471-2105-11-522
**Cite this article as:** Hartsperger *et al.*: Structuring heterogeneous biological information using fuzzy clustering of k-partite graphs. *BMC Bioinformatics* 2010 **11**:522.