Internet Mathematics Vol. 7, No. 4: 255–273

Extension and Robustness of Transitivity Clustering for Protein–Protein Interaction Network Analysis

Tobias Wittkop, Sven Rahmann, Richard Röttger, Sebastian Böcker, and Jan Baumbach

Abstract. Partitioning biological data objects into groups such that the objects within the groups share common traits is a longstanding challenge in computational biology.

Recently, we developed and established transitivity clustering, a partitioning approach based on weighted transitive graph projection that utilizes a single similarity threshold as density parameter. In previous publications, we concentrated on the graphical user interface and on concrete biomedical application protocols. Here, we contribute the following theoretical considerations: (1) We provide proofs that the average similarity between objects from the same cluster is above the user-given threshold and that the average similarity between objects from different clusters is below the threshold. (2) We extend transitivity clustering to an overlapping clustering tool by integrating two new approaches. (3) We demonstrate the power of transitivity clustering for protein-complex detection. We evaluate our approaches against others by utilizing gold-standard data that was previously used by Brohée et al. for reviewing existing bioinformatics clustering tools.

The extended version of this article is available online at http://transclust.mpi-inf.mpg.de.

ISSN: 1542-7951 print

1. Background

Partitioning biomedical data objects into clusters such that the objects within the clusters share common traits is a longstanding challenge in computational biology. Typical examples are the identification of functionally related proteins, the detection of clusters of coregulated genes in gene-expression studies, and the prediction of protein complexes in protein–protein interaction networks. The usual starting point of a cluster analysis is a pairwise similarity matrix calculated using a function that assigns a similarity value to each pair of objects. One can transform such a matrix into a weighted undirected graph whose nodes correspond to the objects, and weighted edges to the similarities. Based on this matrix/graph, clustering strategies aim to identify groups of densely connected elements. A set of user-defined density parameters controls the size and the number of the resulting clusters.

The selection of "good" density parameters is crucial and difficult, since it strongly depends on the similarity function used and the real-world question behind the cluster analysis. For example, we might want to cluster a set of proteins into protein superfamilies (large clusters, weak density parameters) or into protein families (small clusters, restrictive parameters). Typical bioinformatics clustering tools include connected component analysis, k-means, Markov clustering, restricted neighborhood search clustering, spectral clustering, affinity propagation, and superparametric clustering [Enright and Ouzounis 00, Enright et al. 02, Frey and Dueck 07, Paccanaro et al. 06, Wittkop et al. 07, Wittkop et al. 10, Blatt et al. 96, King et al. 04, Blatt et al. 96].

Previously, we developed transitivity clustering (TC) [Wittkop et al. 10], a data-partitioning tool for the integrated clustering of biological data based on solving the so-called weighted transitive graph projection problem (WTGPP). This problem has been given many names in the literature, most prominently "weighted cluster editing" [Böcker et al. 08, Rahmann et al. 07, Wittkop et al. 07]. We proposed a combination of heuristic and exact approaches to tackle this problem. We stressed that the biologist's success in typical cluster analysis depends on more than an efficient algorithm. We discussed the availability of meaningful pairwise similarity functions for specific biological problems as well as appropriate ways to estimate application-specific density parameters. We introduced several software toolkits for an easy-to-use graphical analysis of protein similarity networks for the detection of functionally related proteins. In this paper we address important properties of the transitivity clustering approach as well as interesting features of the underlying WTGPP that were neglected in [Wittkop et al. 10].

1.1. Contributions

While in our previous publication, we concentrated on the graphical user interface and on concrete application protocols for protein homology detection, we contribute the following theoretical considerations here:

1. Proofs: We show that the average similarity between all elements within the same cluster is above the threshold, and that the average similarity between elements from different clusters is below the threshold.
2. Extensions: We extend transitivity clustering with two overlapping clustering strategies: fuzzy associations and cost reduction. While the first requires the user to set an additional threshold, the second approach does not need any further input parameters.
3. Robustness: We evaluate transitivity clustering and the two overlapping clustering extensions on a typical bioinformatics problem: the identification of protein complexes in protein–protein interaction networks. In particular, we evaluate the robustness of our approaches in comparison to other standard bioinformatics clustering approaches.

1.2. Definitions

Throughout this article, we use several definitions. First, we provide some basic graph-theoretic definitions. Afterward, we define clustering and the clustering paradigm behind TC.

Definition 1.1. (Undirected simple graph.) An *undirected simple graph* $G = (V, E)$ consists of a set of nodes V and a set of edges $E \subseteq \binom{V}{2}$, where $\binom{V}{2}$ denotes the set of all two-element subsets of V. Following this definition, the edges are undirected, and the graph contains no self-loops or multiple edges between two nodes.

We will write uv for an unordered pair $\{u, v\} \in \binom{V}{2}$.

Definition 1.2. (Induced subgraph.) An *induced subgraph* $G' = (V', E')$ of a graph $G = (V, E)$ consists of a chosen set of nodes $V' \subset V$ and the set of edges $E' := E \cap \binom{V'}{2} \subset E$, so E' consists of exactly those edges that connect elements of V' and are present in E.

Definition 1.3. (Path.) In a graph $G = (V, E)$, a *path* between two nodes $u, v \in V$ is a sequence of nodes $u = v_1, \ldots, v_n = v$ for which every connecting edge exists, i.e., for $1 \leq i \leq n-1$, $\{v_i, v_{i+1}\} \in E$.

Definition 1.4. (Connected component.) A *connected component* of an undirected simple graph $G = (V, E)$ is an induced subgraph $G' = (V', E')$ of G, where V' is a maximal subset of V, such that there exists a path between every two nodes in G'.

Clustering is defined as the assignment of objects into groups such that objects within the same group are more similar to each other than to objects from different groups. Here, we distinguish two types of clusterings: (1) Partitional clustering divides the set of objects into disjoint groups, and (2) overlapping clustering allows assignments of objects to multiple groups.

Definition 1.5. (Partitional clustering.) A *partitional clustering* of a set of objects S is a subset $S' = \{s_1, \ldots, s_n\}$ of the power set $\mathcal{P}(S)$ such that $S = \bigcup_{i=1}^{n} s_i$ and $s_i \cap s_j = \varnothing$, $1 \leq i \neq j \leq n$.

In the remainder of this article, by "clustering" we shall mean "partitional clustering" if not stated otherwise.

Definition 1.6. (Overlapping clustering.) An *overlapping clustering* of a set of objects S is a subset $S' = \{s_1, \ldots, s_n\}$ of $\mathcal{P}(S)$ such that $S = \bigcup_{i=1}^{n} s_i$.

The clustering model underlying transitivity clustering is based on solving the so-called weighted transitive graph projection problem (WTGPP). We need to define transitivity first.

Definition 1.7. (Transitive graph.) An undirected simple graph $G = (V, E)$ is called *transitive* if

$$\text{for all triples } uvw \in \binom{V}{3}, \quad uv \in E \text{ and } vw \in E \text{ implies } uw \in E.$$

The WTGPP is now defined as follows.

Problem 1.8. (Weighted transitive graph projection problem.) Given a set of objects V, a threshold $t \in \mathbb{R}$, and a pairwise similarity function $\mathrm{sim}\colon \binom{V}{2} \to \mathbb{R}$, the graph G is defined as

$$G = (V, E);\ E = \left\{ uv \in \binom{V}{2} : \mathrm{sim}(uv) > t \right\}.$$

The weighted transitive graph projection problem is the determination of a transitive graph $G' = (V, E')$ such that there exists no other transitive graph $G'' = (V, E'')$ with $\mathrm{cost}\,(G \to G'') < \mathrm{cost}\,(G \to G')$. Here the modification costs

are defined as

$$\operatorname{cost}(G \to G') := \underbrace{\sum_{uv \in E \setminus E'} |\operatorname{sim}(uv) - t|}_{\text{deletion cost}} + \underbrace{\sum_{uv \in E' \setminus E} |\operatorname{sim}(uv) - t|}_{\text{addition cost}}.$$

Note that more than one solution for a given problem instance may exist, but this case almost never occurs in practice if the similarity function is diverse and real-valued. Further, note that this problem is NP-hard [Křivánek and Morávek 86] as well as APX-hard [Charikar et al. 03].

1.3. Brief Overview of Transitivity Clustering

We report the resulting connected components of G', i.e., the solution of the WTGPP of G, as clusters (partitional clustering). To compute the solution of this hard problem, we developed and implemented transitivity clustering (TC). We utilize a combination of heuristic [Rahmann et al. 07, Wittkop et al. 07] and exact [Böcker et al. 08] algorithms; see the TC paper [Wittkop et al. 10] for more details about the algorithms, the strategy, the software, and its application to genetic sequence clustering. We briefly highlight the main advantages of transitivity clustering.

1. As we will demonstrate for protein–protein interaction network clustering, transitivity clustering is very robust against a certain noise level in the similarity function.
2. To adjust the number and size of the clusters, transitivity clustering needs only one density parameter: the similarity threshold. It directly corresponds to the similarity function used and thereby ensures easy interpretability of the clustering results. In this article, we will show (1) that the average similarity between elements within one cluster is above the chosen threshold and (2) that the average similarity between elements from different clusters is below the threshold.
3. Transitivity clustering comes with a set of user-friendly implementations and interfaces: (1) A web server provides easy access for small-scale data sets. (2) Powerful stand-alone software provides command-line as well as GUI-based access to all TC features. (3) Three Cytoscape [Cline et al. 07] plug-ins allow for integrated clustering and visualization of many kinds of biological similarity networks. TC is free to use, needs only minimal user input, and offers comprehensive aid with the most important steps in a typical biomedical cluster analysis workflow.

4. The most important of these steps is the identification of a meaningful similarity threshold that controls the size and granularity of the clusters. To aid the end user with this important step, we deliver functions with transitivity clustering that allow for incorporating background knowledge, for instance small gold standards, into the clustering procedure in an easy-to-use fashion.

5. Several extended functions for eased data handling and visualization, increased accuracy, and processing speedup are directly accessible. The applicability of upper bounds for node merging may serve as an example here.

2. Properties of the Weighted Transitive Graph Projection Problem

The first property of the weighted transitive graph projection problem (WTGPP) that we prove may considerably reduce the running time of any algorithm that solves this problem. We show that it is sufficient to solve the WTGPP for each connected component of a similarity graph individually. Splitting the problem into several smaller problems allows one to solve this NP-complete problem even for very large data sets with hundreds of thousands of nodes. This fact has been mentioned several times in the literature [Rahmann et al. 07, Böcker et al. 08] and lies at the core of many algorithms for the problem, but no formal proof has been provided.

Theorem 2.1. *Given a set of objects V, a threshold $t \in \mathbb{R}$, and a pairwise similarity function* $\mathrm{sim}\colon \binom{V}{2} \to \mathbb{R}$ *and a graph $G = (V, E)$ as defined in the WTGPP, to solve the WTGPP of G it is sufficient to solve the WTGPP of the connected components $G_1, \ldots, G_m$ of G, i.e., if $G'_1, \ldots, G'_m$ are solutions for the WTGPP of the connected components of G, then $G' = \bigcup_{i=1}^{m} G'_i$ is a solution for the WTGPP of G.*

Proof. To prove this theorem, it is sufficient to show that there exists no solution of a WTGPP with cliques that intersect with multiple connected components of the similarity graph G. This will be done by assuming that such a solution exists and deriving a contradiction.

Let G' be a solution of a WTGPP with cliques that contain objects from different connected components of G. Edges in G' between objects of different connected components of G correspond to a similarity smaller than the threshold t due to the definition of the similarity graph. Hence, deleting them results in a decrease in costs. Furthermore, deleting all edges between a subset of nodes

of a clique and all other nodes of that clique leads to two disjoint cliques, since all nodes within the two sets are still connected. Consequently, splitting the cliques in G' into cliques that have no intersection with two different connected components of G reduces the costs and still respects the transitivity rule. It is, in turn, a transitive graph with lower costs than G'. This is a contradiction to the minimal-cost criterion of a solution of a WTGPP. □

One of the advantages of transitivity clustering is its simplicity. Only one density parameter has to be chosen to determine the number and size of the resulting clusters. In the following, we will prove properties of a clustering that allow a user to interpret the clustering in the context of the chosen similarity function and threshold. Furthermore, these properties assist in the detection of an appropriate threshold.

Theorem 2.2. *Let $C = \{C_1, \ldots, C_m\}$ be the clusters of a solution $G' = (V, E')$ for a given WTGPP with threshold t and similarity function* sim.

(i) *The mean similarity between an element u and all other elements of its clique C_u is greater than or equal to the threshold t for all elements $u \in V$.*

(ii) *The mean similarity between all elements of one cluster C_i is greater than or equal to t for all cliques $C_i \in C$.*

Proof. Statement (ii) is a direct consequence of (i). To prove (i), the negative proposition is assumed and a contradiction derived. Let u be an element of the cluster C_i of size $|C_i| \geq 2$. Assume that the mean similarity between u and all other elements of C_i is below t:

$$\begin{aligned} \operatorname{mean}\operatorname{sim}(u, C_i) = \frac{1}{|C_i| - 1} \sum_{v \in C_i \setminus \{u\}} \operatorname{sim}(uv) < t \\ \iff \sum_{v \in C_i \setminus \{u\}} \operatorname{sim}(uv) < t \cdot (|C_i| - 1). \end{aligned}$$

Now, $C' = \{C_1, \ldots, C_i \setminus \{u\}, \ldots, C_m, \{u\}\}$ is a decomposition of the elements into cliques and hence a putative solution for the underlying WTGPP. The costs for C' can be calculated using the costs that appear to build C and adding all costs to remove edges between u and C_i. Note that these additional costs may be negative for edges that did not exist in the initial graph and had to be added to create C_i. Using the assumption that the mean similarity between u and all

elements of C_i is below t, the cost difference between C and C' is consequently

$$\sum_{v \in C_i \setminus \{u\}} (\mathrm{sim}(uv) - t) = \Big(\sum_{v \in C_i \setminus \{u\}} \mathrm{sim}(uv) \Big) - (t \cdot (|C_i| - 1)) < 0.$$

This is a contradiction to the assumptions that C is a solution of the WTGPP, since there exists a decomposition into cliques with lower costs. □

A statement about the average similarity between an object and all the objects of a foreign cluster is not possible. In the following example we illustrate the case where the average similarity of one element to a different cluster then its own can be above the threshold that was used to obtain the clustering.

Example 2.3. Let $V = \{a, b, c\}$ be the elements of interest. Let the similarities between these elements be $\mathrm{sim}(ab) = 0.5$, $\mathrm{sim}(ac) = 0$, and $\mathrm{sim}(bc) = 1$. For a threshold $t = 0.4$, the clustering obtained by solving the corresponding WTGPP is $C = \{C_1, C_2\} = \{\{a\}, \{b, c\}\}$. The mean similarity between objects within one cluster is obviously above the threshold, and the mean similarity between these clusters is below the threshold:

$$\frac{\mathrm{sim}(ab) + \mathrm{sim}(ac)}{2} = 0.25 < t.$$

The mean similarity between b and a, which is one element from one cluster and all elements from the other, is 0.5 and hence above the threshold.

It is possible, though, to make a statement about the average similarity between two clusters.

Theorem 2.4. *Let $C = \{C_1, \ldots, C_m\}$ be the cliques of a solution to a given WTGPP with threshold t and similarity function* sim. *The mean similarity between two cliques C_i and C_j is below the threshold for all $1 \leq i < j \leq m$.*

Proof. Again the proof for this theorem is done by assuming the negated proposition and deriving a contradiction. Let C_i and C_j with $i \neq j$ be cliques with average similarity above the threshold t. The decomposition of the objects into cliques $C' = (C \setminus \{C_i, C_j\}) \cup \{C_i \cup C_j\}$ is a putative solution for the WTGPP. The costs for C' can again be calculated using the costs for C and adding all costs for adding the connective edges between C_i and C_j:

$$\mathrm{costs}(C') = \mathrm{costs}(C) + \sum_{u \in C_i} \sum_{v \in C_j} (-\mathrm{sim}(uv) + t).$$

Consequently, edges that were initially removed between C_i and C_j to obtain C contribute negatively to the additional costs, while edges that were not present in the similarity graph of this problem increase the costs of C'.

In order to establish a contradiction to the assumption that C is a solution for the WTGPP, all that remains is to show that the second term is below zero. This can easily be derived from the initial assumption that the average similarity between C_i and C_j is above the threshold, since

$$\begin{aligned} &\operatorname{mean}\operatorname{sim}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{u \in C_i} \sum_{v \in C_j} \operatorname{sim}(uv) > t \\ \iff &\sum_{u \in C_i} \sum_{v \in C_j} (\operatorname{sim}(uv) - t) > 0. \end{aligned}$$

□

3. Extension to Overlapping Clustering

Up to now, we have described a partitioning clustering approach in which each object is assigned to exactly one cluster. This may be problematic for some applications in which we might wish to assign some objects to multiple clusters. In our robustness analysis, we will present such an example: Some proteins might act as parts of more than one protein complex. To account for this, we integrated two new methods for creating an overlapping clustering using WTGP. The first approach creates a fuzzy assignment. Based on the similarity function, each element–cluster pair is assigned a value between zero and one, which is subsequently used to assign those objects to additional clusters whose value exceeds a second threshold. The second method is more closely related to the initial WTGPP. Here, the WTGPP is solved first for the user-given threshold. Afterward, single elements are co-assigned to the remaining clusters if this reduces the overall costs. No second threshold is required. Note that the emerging assignment of objects to multiple clusters contradicts the transitivity rule. Formal descriptions follow.

3.1. Fuzzy Association

In the following, let $C = \{C_1, \ldots, C_m\}$ be a clustering obtained for a specific threshold by solving the corresponding WTGPP, let $V = \{v_1, \ldots, v_n\}$ be the objects that were clustered, and let $\operatorname{sim}\colon \binom{V}{2} \to \mathbb{R}$ be the pairwise similarity function. Without loss of generality, we can assume that $\operatorname{sim}\colon \binom{V}{2} \to \mathbb{R}^+$, since a similarity function that has negative values can be transformed into a strictly

positive function by adding the lowest value to all similarities and the chosen threshold. This will not change the outcome of the clustering, since the objective function of the WTGPP works only on the difference between similarity and threshold, which does not change.

Our method assigns to each object v_i and each cluster C_j a value $f_{i,j} = f(v_i, C_j) \in [0, 1]$ such that

$$\sum_j f(v_i, C_j) = 1 \text{ for all } v_i \in V.$$

To obtain f, first the mean similarity $m_{i,j}$ between every pair of object v_i and cluster C_j is calculated and stored in the matrix $M = (m_{i,j}) \in \mathbb{R}^{n \times m}$:

$$m_{i,j} = \frac{1}{|C_j \setminus v_i|} \sum_{v \in C_j, v \neq v_i} \text{sim}(vv_i).$$

The matrix is transformed into a column stochastic matrix $F = (f_{i,j}) \in [0, 1]^{n \times m}$ to fulfill the criteria as defined above. The value $f_{i,j}$ determines how well an element v_i fits into a cluster C_j. To get an overlapping clustering, a second threshold is needed. This value $t_2 \in [0, 1]$ influences the number of allowed overlaps. The overlapping clustering is created by adding an element v_i to an additional cluster C_j if $f_{i,j}$ exceeds t_2.

Choosing a low value t_2 would lead to many overlaps, while a value $t_2 \approx 1$ would assign only a few elements to multiple clusters. We exclude singletons from fuzzy assignments due to two factors that would increase the false positive assignment of nodes to clusters. Firstly, a singleton node does not have an average similarity to its own cluster, and thus it is likely to be assigned to an additional cluster, although the initial clustering suggests otherwise. Secondly, the average similarity of a node with a singleton cluster is the actual similarity between the two nodes. Transitivity clustering is a robust clustering technique, since erroneous similarities can be compensated by their neighboring edges. This advantage would be lost if an assignment were based on a single edge and thus would increase the chance of false-positive groupings.

3.2. Cost Reduction

The basic concept of cost reduction (CR) is to assign objects to an additional cluster if the internal costs are below zero. By adding a node to an additional cluster and hence all edges between the node and the cluster, the overall costs may be reduced if initially removed edges are added again.

Starting with a partitional clustering $C = \{C_1, \ldots, C_m\}$, each object u can be assigned to an additional cluster C_j if

$$\text{costs}(u, C_j) = \sum_{v \in C_j} (\text{sim}(uv) - t) < 0.$$

We now proceed in a greedy fashion: For the combination u, C_j with smallest costs and $\text{costs}(u, C_j) < 0$, the altered clusters $C' = \{C_1, \ldots, C_j \cup \{u\}, \ldots, C_m\}$ replace the previous clusters, and the process starts again. These operations are executed until no further improvement is possible, i.e., no assignment of an object to an additional cluster would reduce the internal costs of any cluster. As before, singletons are excluded as well as assignments to singletons. Note that Theorem 2.2 still holds for this overlapping clustering, since elements can be added to one cluster only if this reduces the overall cost, i.e., the average similarity between the newly added element and all other elements of the new cluster is above the threshold.

4. Robustness Analysis

In the following robustness analysis, we investigate the effect of noise within the utilized similarity function. We simulate this noise by randomly modifying edges in the input data. Subsequently, we compare the ability of transitivity clustering and other clustering approaches to reconstruct a given gold standard from such perturbed data.

Previously, [Brohée and van Helden 06] presented an evaluation of bioinformatics clustering tools for the task of reconstructing protein complexes in protein–protein interaction (PPI) networks. The authors compared Markov clustering (MCL), restricted neighborhood search clustering (RNSC), MCODE, and superparametric clustering (SPC). In order not to replicate existing results, we use only the best tools from Brohée et al.'s evaluation study, namely MCL and RNSC.

In what follows, we use different quality functions to measure the clustering accuracy. For a summary and formal description, the reader is referred to the appendix, Section 6.

4.1. Data

We use the same data as in [Brohée and van Helden 06]. The data set consists of 1095 yeast proteins obtained from the MIPS database [Mewes et al. 04]. As in [Brohée and van Helden 06], the protein complex annotations of these proteins are used as a gold standard. Note that proteins may belong to multiple

protein complexes. Hence, the data set is also suited to evaluating the overlapping clustering functionality of transitivity clustering. We start with an initial graph $G = (V, E)$ in which the proteins are represented as nodes. We draw edges between all proteins known to interact as a protein complex.

Brohée et al. modified this graph by randomly adding and deleting a certain number of edges.[1] In what follows, let $A_{i,j}$ denote the graph where $i\%$ of edges are added and $j\%$ of edges are deleted.

4.2. Evaluation Method

The techniques MCL and RNSC performed best in the review of [Brohée and van Helden 06]. We proceed with TC in the same way and compare the results with MCL and RNSC. For all input graphs $A_{i,j}$, we set all edge weights to 1. Now we add edges between all unconnected nodes and set the edge weight to 0. This results in a *similarity graph*. Note that a protein may contribute to multiple complexes, which makes the graph intransitive for similarity thresholds greater than 0. Afterward, we iteratively cluster the input graphs $A_{i,j}$ for varying thresholds between 0 and 1. For each clustering result, we compare against the gold standards and calculate the F-measure as a measure of quality.

We evaluate the robustness of our approach in accordance with [Brohée and van Helden 06]. Therefore, the presented results are created with one fixed density parameter for all modified graphs $A_{i,j}$. We used a series of different density parameters for MCL and RNSC. For both methods, we chose the density parameter that achieves the highest average F-measure over all graphs $A_{i,j}$, since it reflects the most robust scenario. Note that we play fairly, since we apply a single density parameter (the one that provides the best average results) to each graph individually, similarly to what is done in [Brohée and van Helden 06]. Subsequently, we performed analogous analyses with the two previously defined overlapping methods of TC.

In contrast to the original study, we use the F-measure to assess the clustering quality. We argue that the separation, which serves as the main quality measure in the study from Brohée et al., is a biased measure, since it strongly depends on the number of clusters. A conservative tool, for instance, that produces a comparably high number of singletons cannot reach a good score. The F-measure compares a gold-standard cluster with the best cluster in the produced clustering and hence ignores this drawback. A drastic illustration of the problem with the separation is a comparison of the gold standard with itself: For the Brohée et al.

[1] Modified graphs are available at http://rsat.bigre.ulb.ac.be/rsat/data/published_data/brohee_2006_clustering_evaluation/.

data this leads to a separation of 0.62, which is smaller than the results achieved by any clustering methods. In contrast, the F-measure is 1 (the highest possible value) for this comparison, as for any other clustering compared to itself.

4.3. Evaluation Results

Figure 1 illustrates the results for all altered graphs by means of a heat map whose colors represent the achieved F-measures. TC and RNSC produce similar results for the tested graphs. Both methods are robust against edge additions and quite robust against edge deletions. For the original graph with no edge additions or deletions, these two methods achieve a high F-measure of 0.85. After doubling the number of edges and deleting 40% of the original edges, RNSC still achieves an F-measure of 0.69 and TC one of 0.73. MCL generally has lower F-measures and seems to be less robust against edge additions. Starting with an F-measure of 0.79 for the unaltered graph, the same scenario as above leads to an F-measure of only 0.46. On the other hand, it seems very robust against edge removals, since the F-measure drops by only 0.01 if 40% of the initial edges are deleted and no edges are added.

Table 1 presents the results of the two overlapping methods fuzzy association (FA) and cost reduction (CR) in comparison to the partitioning method

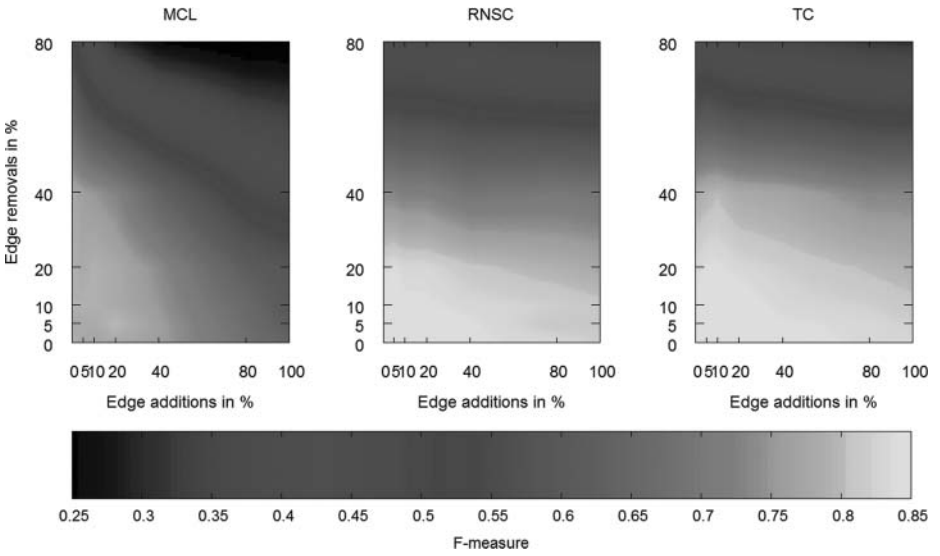

Figure 1. Results of robustness analysis. The F-measure serves as quality measure (color figure available online).

	0	05	10	20
0	0.85/0.87/0.87	0.85/0.86/0.86	0.85/0.85/0.86	0.85/0.85/0.86
5	0.85/0.86/0.87	0.85/0.86/0.86	0.85/0.85/0.86	0.85/0.85/0.85
10	0.84/0.85/0.86	0.85/0.85/0.86	0.85/0.85/0.86	0.85/0.85/0.85
20	0.83/0.83/0.85	0.84/0.84/0.85	0.84/0.84/0.84	0.83/0.83/0.84
40	0.78/0.79/0.81	0.79/0.8/0.81	0.8/0.8/0.81	0.78/0.79/0.79
80	0.4/0.42/0.44	0.42/0.43/0.45	0.39/0.43/0.41	0.37/0.39/0.4

	40	80	100
0	0.85/0.85/0.85	0.83/0.84 /0.83	0.83/0.82/0.82
5	0.84/0.84/0.84	0.83/0.83/0.82	0.82/0.82/0.81
10	0.83/0.83/0.84	0.82/0.82/0.82	0.81/0.81/0.81
20	0.82/0.81/0.83	0.8/0.8/0.8	0.79/0.79/0.79
40	0.77/0.78/0.77	0.73/0.74/0.72	0.73/0.73/0.7
80	0.36/0.36/0.37	0.32/0.31/0.32	0.31/0.3/0.31

Table 1. Comparison of the overlapping methods of TC. The F-measures are for partitional transitivity clustering, the fuzzy associations method, and the cost reduction method. The columns represent the percentage of added edges, while the rows represent the percentage of removed edges.

of transitivity clustering. Fuzzy association requires a second threshold, which is set here to a conservative value of 0.8. This means that only elements with high similarity to an additional cluster are assigned to multiple groups. Both overlapping methods FA and CR are as robust as the partitioning method and even achieve higher F-measures in most cases.

5. Conclusion

In previous publications we presented transitivity clustering, a clustering tool that can be applied to various kinds of biological data sets to identify tightly connected substructures. Here, we have contributed theoretical foundations regarding the robustness of TC. The similarity threshold, the only density parameter of TC, determines the cluster associations in such a way that the average similarity within the clusters is above the threshold, and that between the clusters is below the threshold.

We believe that this property of TC plays an important role in its robustness against a certain noise level in the similarity function, because mere noise is unlikely to change the average similarity of a certain cluster such that the resulting clusters would change in a significant way. In addition, we have presented

two extensions of TC that allow for the more realistic clustering scenario of overlapping clustering. The first, fuzzy association, lets the user adjust the amount of desired overlap, while the second, cost reduction, has no parameters and still guarantees the above-mentioned properties regarding the average similarity. We argue that particularly the overlapping clustering methods will prove useful in further biomedical applications.

6. Appendix: Quality measures

We evaluate clustering quality by comparing it with a gold-standard assignment. This external quality evaluation allows one to compare different approaches that optimize different internal optimization functions. For this purpose, various measures have been developed. Here we briefly introduce all of these measures.

In the following, let $C = \{C_1, \ldots, C_n\}$ be the clustering obtained from the algorithm, and $K = \{K_1, \ldots, K_m\}$ the reference clustering. Furthermore, let $T = (t_{i,j}) \in \mathbb{N}^{m \times n}$ denote the matrix each entry of which is the number of common objects between K_i and C_j,

$$t_{i,j} := |\{K_i \cap C_j\}|, \quad 1 \leq i \leq m,\ 1 \leq j \leq n.$$

This section uses also the standard abbreviations for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Another notation is $|T|$ for the sum of all entries in T and $|T_{\cdot,j}|$ and $|T_{i,\cdot}|$ for the sums of the entries of the ith row and jth column respectively.

According to the evaluation of clustering algorithms for PPI networks in [Brohée and van Helden 06], the following definitions are given.

Definition 6.1. (Positive predictive value.) The *positive predictive value* (PPV),

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \in [0, 1],$$

describes the ratio of correct predictions to all predictions. It can reach the value 1 only if no false positive predictions occur, i.e., in the case of clustering, if no pair of objects that belong to different clusters in the gold standard are assigned to the same cluster. Since the reference clustering may be overlapping, the PPV for each pair of clusters C_j, K_i is defined as

$$\mathrm{PPV}(C_j, K_i) = \frac{t_{i,j}}{|T_{\cdot,j}|}.$$

A clusterwise PPV can then be defined for each cluster C_j as

$$\mathrm{PPV}(C_j) = \max_{1 \le i \le m} \mathrm{PPV}(C_j, K_i).$$

To get an overall PPV between two clusterings, the clusterwise PPVs are incorporated as follows:

$$\mathrm{PPV}(C, K) = \frac{\sum_{j=1}^{n} \mathrm{PPV}(C_j) \cdot |T_{\cdot,j}|}{|T|} = \frac{1}{|T|} \cdot \sum_{j=1}^{n} \left(\max_{1 \le i \le m} \frac{t_{i,j}}{|T_{\cdot,j}|} \right) \cdot |T_{\cdot,j}|.$$

Definition 6.2. (Sensitivity.) The *sensitivity*

$$\mathrm{Sen} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

reflects the quantity of the correct predictions in relation to all true class members. In the case of clustering, this is the ratio between objects that are in the same cluster, both in the reference clustering and in the obtained clustering, against all objects in the reference clustering.

A *reference-cluster-wise sensitivity* is defined as

$$\mathrm{Sen}(K_i) = \max_{1 \le j \le n} \frac{t_{i,j}}{|K_i|}.$$

A general sensitivity can then be calculated as

$$\mathrm{Sen}(C, K) = \frac{\sum_{i=1}^{m} |K_i| \cdot \mathrm{Sen}(K_i)}{|K|} = \frac{\sum_{i=1}^{m} |K_i| \cdot \max_{1 \le j \le n} \frac{t_{i,j}}{|K_i|}}{|K|}.$$

Definition 6.3. (Accuracy.) The *accuracy* (ACC) is a trade-off between PPV and sensitivity. Both values on their own can be high even if the clustering is not perfect. A clustering with only singletons would lead to a high PPV, since no false positive prediction occurs, while building one big cluster containing all elements would have the maximal sensitivity value. Neither of these examples is necessarily desirable. Hence, a combination of these values, the accuracy, evaluates the quality better.

The *arithmetic accuracy* is the arithmetic mean of sensitivity and PPV:

$$\mathrm{ACC}_{\mathrm{arithmetic}}(C, K) = \frac{\mathrm{Sen}(C, K) + \mathrm{PPV}(C, K)}{2}.$$

The *geometric accuracy* takes the geometric mean of sensitivity and PPV:

$$\mathrm{ACC}_{\mathrm{geometric}} = \sqrt{\mathrm{Sen}(C, K) \cdot \mathrm{PPV}(C, K)}.$$

A novel distance measure was introduced in [Brohée and van Helden 06] called separation. In contrast to the above-described values, it takes all pairwise relations between the clusters obtained from the algorithm and the reference clusters into account, and does not concentrate on the best-matching cluster.

Definition 6.4. (Separation.) The *separation* for each pair of clusters C_j and K_i is defined as

$$\mathrm{Sep}(C_j, K_i) = \frac{t_{i,j}^2}{|T_{i,\cdot}| \cdot |T_{\cdot,j}|}.$$

Now it is possible to define the separation for each cluster C_j and K_i as the sum of pairwise separations

$$\mathrm{Sep}(C_j) = \sum_{i=1}^{m} \mathrm{Sep}(C_j, K_i)$$

and

$$\mathrm{Sep}(K_i) = \sum_{j=1}^{n} \mathrm{Sep}(C_j, K_i).$$

To obtain an overall value for the two clusterings C and K, one takes the mean of separations for all clusters in C (clusterwise) and K (reference-cluster-wise) and subsequently calculates the geometric mean:

$$\begin{aligned}\mathrm{Sep}(C, K) &= \sqrt{\frac{\sum_{j=1}^{n} \mathrm{Sep}(C_j)}{|C|} \cdot \frac{\sum_{i=1}^{m} \mathrm{Sep}(K_i)}{|K|}} \\ &= \frac{\sum_{j=1}^{n} \sum_{i=1}^{m} \mathrm{Sep}(C_j, K_i)}{\sqrt{m \cdot n}} \\ &= \frac{\sum_{j=1}^{n} \sum_{i=1}^{m} \frac{t_{i,j}^2}{|T_{i,\cdot}| \cdot |T_{\cdot,j}|}}{\sqrt{m \cdot n}}.\end{aligned}$$

Definition 6.5. (F-measure.) The last quality measure is based on the general definition of recall (sensitivity), precision (PPV), and the F-measure:

Recall or sensitivity: $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$.

Precision or PPV: $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$.

F-measure: $2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} = \frac{2 \cdot \mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})+(\mathrm{TP}+\mathrm{FN})}$.

In [Paccanaro et al. 06], the F-measure for comparing a clustering C against a reference clustering K was modified in the following way. First, the best cluster

C_j for each cluster K_i of the reference is found with respect to the standard definition of F-measure,

$$\text{F-measure}(K_i) = \max_{1 \le j \le n} \frac{2 \cdot t_{i,j}}{|C_j| + |K_i|}.$$

The overall F-measure is then defined as

$$\begin{aligned}\text{F-measure}(C, K) &= \frac{1}{\sum_{i=1}^{m} |K_i|} \sum_{i=1}^{m} (|K_i| \cdot \text{F-measure}(K_i)) \\ &= \frac{1}{\sum_{i=1}^{m} |K_i|} \sum_{i=1}^{m} \left(|K_i| \cdot \max_{1 \le j \le n} \frac{2 \cdot t_{i,j}}{|C_j| + |K_i|} \right).\end{aligned}$$

Acknowledgments. The work of JB was supported by the Cluster of Excellence for Multimodal Computing (MMCI). TW received support from NIH grant R01 LM009722 and the Buck Trust. RR is grateful for financial support of the International Max Planck Research School (IMPRS).

References

[Blatt et al. 96] M. Blatt, S. Wiseman, and E. Domany. "Superparamagnetic Clustering of Data." *Physical Review Letters* 76:18 (1996), 3251–3254.

[Böcker et al. 08] S. Böcker, S. Briesemeister, and G. W. Klau. "Exact Algorithms for Cluster Editing: Evaluation and Experiments." In *Workshop on Experiments with Algorithms*, edited by C. C. McGeoch, Lecture Notes in Computer Science 5038, pp. 289–302. New York: Springer, 2008.

[Brohée and van Helden 06] S. Brohée and J. van Helden. "Evaluation of Clustering Algorithms for Protein–Protein Interaction Networks." *BMC Bioinformatics* 7 (2006), 488.

[Charikar et al. 03] M. Charikar, V. Guruswami, and A. Wirth. "Clustering with Qualitative Information." *Journal of Computer and System Sciences* 71 (2003), 360–383.

[Cline et al. 07] M. S. Cline, M. Smoot, E. Cerami, et al. "Integration of Biological Networks and Gene Expression Data Using Cytoscape." *Nature Protocols* 2:10 (2007), 2366–2382.

[Enright and Ouzounis 00] A. J. Enright and C. A. Ouzounis. "GeneRAGE: A Robust Algorithm for Sequence Clustering and Domain Detection." *Bioinformatics* 16:5 (2000), 451–457.

[Enright et al. 02] A. J. Enright, S. V. Dongen, and C. A. Ouzounis. "An Efficient Algorithm for Large-Scale Detection of Protein Families." *Nucleic Acids Research* 30:7 (2002), 1575–1584.

[Frey and Dueck 07] B. J. Frey and D. Dueck. "Clustering by Passing Messages between Data Points." *Science* 315:5814 (2007), 972–976.

[King et al. 04] A. D. King, N. Przulj, and I. Jurisica. "Protein Complex Prediction via Cost-Based Clustering." *Bioinformatics* 20:17 (2004), 3013–3020.

[Křivánek and Morávek 86] M. Křivánek and J. Morávek. "NP-Hard Problems in Hierarchical-Tree Clustering." *Acta Informatica* 23:3 (1986), 311–323.

[Mewes et al. 04] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, et al. "MIPS: Analysis and Annotation of Proteins from Whole Genomes." *Nucleic Acids Research* 32 (2004) (database issue), D41–D44.

[Paccanaro et al. 06] A. Paccanaro, J. A. Casbon, and M. A. Saqi. "Spectral Clustering of Protein Sequences." *Nucleic Acids Research* 34:5 (2006), 1571–1580.

[Rahmann et al. 07] S. Rahmann, T. Wittkop, J. Baumbach, M. Martin, A. Truss, and S. Boecker. "Exact and Heuristic Algorithms for Weighted Cluster Editing." *Computational Systems Bioinformatics Conference* 6 (2007), 391–401.

[Wittkop et al. 07] T. Wittkop, J. Baumbach, F. Lobo, and S. Rahmann. "Large Scale Clustering of Protein Sequences with FORCE:A Layout Based Heuristic for Weighted Cluster Editing." *BMC Bioinformatics* 8:1 (2007), 396.

[Wittkop et al. 10] T. Wittkop, D. Emig, S. Lange, S. Rahmann, M. Albrecht et al. "Partitioning Biological Data with Transitivity Clustering." *Nature Methods* 7:6 (2010), 419–420.

Tobias Wittkop, Buck Institute for Age Research, 8001 Redwood Blvd, Novato, CA 94945, USA (twittkop@buckinstitute.org)

Sven Rahmann, TU Dortmund, Otto-Hahn-Str. 14, Raum 2.14, 44221 Dortmund, Germany (sven.rahmann@tu-dortmund.de)

Richard Röttger, Max Planck Institute for Informatics, Campus E2.1, Raum 202, 66123 Saarbrücken, Germany (roettger@mpi-inf.mpg.de)

Sebastian Böcker, Friedrich-Schiller-Universität-Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany (sebastian.boecker@uni-jena.de)

Jan Baumbach, Max Planck Institute for Informatics, Campus E2.1, Raum 202, 66123 Saarbrücken, Germany (jbaumbac@mpi-inf.mpg.de)