Comprehensive cluster analysis with Transitivity Clustering

Tobias Wittkop[1], Dorothea Emig[2], Anke Truss[3], Mario Albrecht[2], Sebastian Böcker[3] & Jan Baumbach[2,4,5]

[1]Buck Institute for Age Research, Novato, California, USA. [2]Max Planck Institute for Informatics, Saarbrücken, Germany. [3]Friedrich-Schiller University Jena, Jena, Germany. [4]International Computer Science Institute, University of California at Berkeley, Berkeley, California, USA. [5]Saarland University, Cluster of Excellence for Multimodal Computing and Interaction, Saarbrücken, Germany. Correspondence should be addressed to J.B. (jbaumbac@icsi.berkeley.edu).

Published online 10 February 2011; doi:10.1038/nprot.2010.197

Transitivity Clustering is a method for the partitioning of biological data into groups of similar objects, such as genes, for instance. It provides integrated access to various functions addressing each step of a typical cluster analysis. To facilitate this, Transitivity Clustering is accessible online and offers three user-friendly interfaces: a powerful stand-alone version, a web interface, and a collection of Cytoscape plug-ins. In this paper, we describe three major workflows: (i) protein (super)family detection with Cytoscape, (ii) protein homology detection with incomplete gold standards and (iii) clustering of gene expression data. This protocol guides the user through the most important features of Transitivity Clustering and takes ~1 h to complete.

© 2011 Nature America, Inc. All rights reserved.

INTRODUCTION

Partitioning biological data objects into clusters, such that the objects within the clusters are more similar to each other than to objects from different clusters, is a long-standing challenge in computational biology. We measure the similarity between two objects using a similarity function, e.g., BLAST, for biological sequences. Given a set of objects to be clustered, we compute the value of this function for each pair of objects and reach a so-called similarity matrix. This is the usual starting point for most clustering approaches. Typically, a set of method-specific parameters controls the density of the partitioning, i.e., the number and size of the resulting clusters. Choosing 'good' density parameters is difficult, as there is a strong dependency both on the input data and on the underlying real-world question. For example, we might want to partition a set of protein sequences into protein superfamilies (few but large clusters), or into protein families (many but rather small clusters). The established clustering methods typically used in bioinformatics are Markov clustering[1,2] (protein family detection), hierarchical clustering[3] (gene expression data), connected component analysis[4] (protein homology detection), *k*-means (all kinds of applications), spectral clustering[5] (protein family detection) and affinity propagation[6] (similar to *k*-means, all kinds of applications). However, the end user's ability to perform a successful cluster analysis depends on much more than an efficient algorithm. Most important is the availability of a meaningful pairwise similarity function and an appropriate way to estimate the application-specific density parameters. This is crucial for informative and interpretable clustering results. The whole analysis starts with a list of data objects and ends with a list of sets of objects and the possibility to account for typical follow-up questions.

Recently, we developed and published Transitivity Clustering[7], a new method for biological data partitioning embedded in an integrated data analysis framework. In that paper, we stress that a clustering process incorporates several data analysis steps: identification of a similarity function, computation and postprocessing of a similarity matrix, (optionally) visualization as a similarity network, estimation of meaningful density parameters, clustering of the similarity network/matrix, comparison with given gold standards, fine-tuning of clustering by varying the density parameters, visualization of clustering results and finally graphical follow-up analysis of the results regarding underlying real-world questions. Although all these steps are essential for a successful cluster analysis, most tools concentrate on the clustering algorithm itself and neglect two important functionalities: semiautomatic estimation of appropriate density parameters and visual postprocessing of the clustering results regarding specific (biological) follow-up questions. Transitivity Clustering tools are accessible online at http://transclust.cebitec.uni-bielefeld.de/.

Transitivity Clustering provides the end user with simple interfaces that facilitate each step of the data clustering workflow. **Figure 1** illustrates those parts covered by this paper. With Transitivity Clustering, we provide a powerful stand-alone application. It is capable of clustering hundreds of thousands of data objects but also offers a set of extended clustering functions (see **Box 1** for more details). In addition, Transitivity Clustering comes with a collection of Cytoscape[8] plug-ins. They offer the same basic functionality as the stand-alone version and also provide access to visualization features and several methods to answer typical biological follow-up questions (refer to **Box 2**). Note that we also provide a web interface for small data sets, which provides very basic functionality but is not further discussed in this protocol. In the following section, we outline the clustering strategy behind Transitivity Clustering and briefly discuss major advantages. For a detailed discussion of the method and its performance the reader is referred to the supplementary material in reference 7.

Weighted transitive graph projection

The strategy behind Transitivity Clustering is based on transitive graph projection. We model the given pairwise similarity function/matrix as a similarity graph, in which the nodes correspond to objects and weighted edges to similarity values. Thereafter, the user is required to set a similarity threshold, the only density parameter of Transitivity Clustering. We transform the similarity graph into another graph by subtracting the threshold from the edge weights and subsequently removing those edges with weights below zero. On the basis of this modified graph, we define a cost function for adding and removing edges: the distance of the similarity from the

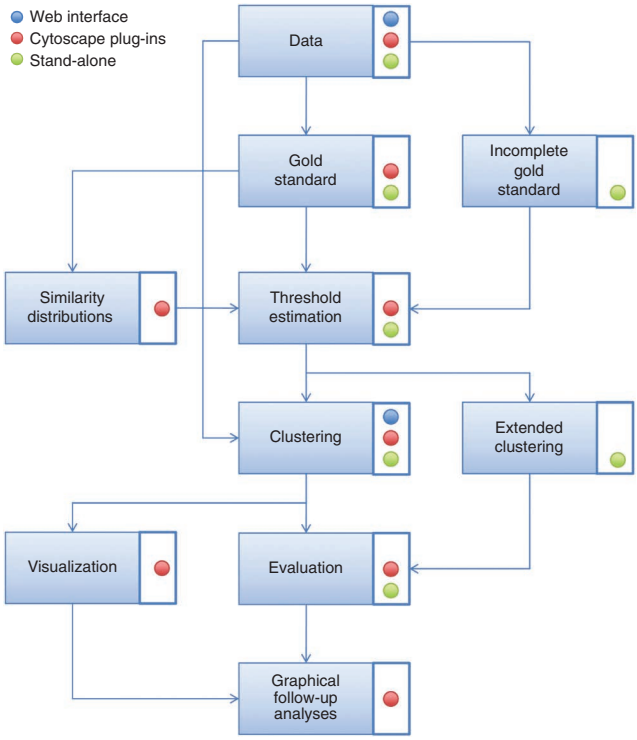

Figure 1 | Overview of the Transitivity Clustering functionalities and user interfaces. This figure outlines the organization of this protocol as well as the structure of typical clustering workflows. Steps marked with a blue dot can be performed with the web interface, steps with a red dot with the Cytoscape plug-ins and steps with green dots with the stand-alone software. In this protocol, we concentrate on the Cytoscape plug-in collection of Transitivity Clustering and the stand-alone version, accessed via Java Web Start.

user-given threshold. The aim now is to transform the modified potentially intransitive graph into a transitive graph with minimal costs for edge additions and removals. (A graph is called transitive if, for each triplet of nodes, the following condition holds: whenever we see an edge from node A to B and an edge from B to C, we require an edge from node A to C as well.) This problem is known as weighted transitive graph projection or weighted graph cluster editing. It is known to be non-deterministic polynomial-time hard (NP-hard) and hard to approximate (APX-hard)[9]. We solve this problem by combining a set of heuristic[10,11] and exact[12] algorithms in such a way that we usually find the optimal solution. For more details about the algorithms and more formal descriptions, see our previous Transitivity Clustering paper[7].

Transitivity Clustering—an overview

We highlight the following major advantages of Transitivity Clustering:

1. Weighted graph cluster editing directly attacks the problem of unraveling hidden transitive substructures in a given similarity matrix or graph. Therefore, we implicitly allow a certain level of noise in the similarity function used. Depending on the specific application case, Transitivity Clustering may be more robust than other approaches; e.g., for protein-protein interaction network clustering, as demonstrated in reference 9.
2. Transitivity Clustering needs only one density parameter to adjust the number and size of clusters: the similarity threshold. It is similarly intuitive and interpretable as the k of k-means (k-means returns exactly k clusters). By solving the weighted transitive graph projection problem this method guarantees the provision of partitioning in which (i) the average similarity between objects within one cluster is above the threshold and (ii) the average similarity between objects from different clusters is below the threshold.
3. As we demonstrate with this protocol, Transitivity Clustering provides several easy-to-use interfaces to walk through all kinds of typical cluster analysis workflows. It is available as a web tool, a stand-alone Java application (with Java Web Start), and integrated into Cytoscape by means of a set of plug-ins.

BOX 1 | EXTENDED FUNCTIONS OF THE STAND-ALONE SOFTWARE

The stand-alone software of Transitivity Clustering offers extended functionalities, compared with the Cytoscape plug-ins. The following list summarizes these additional features; more detailed descriptions can be found in reference 9.

Hierarchical clustering

Two options for performing a hierarchical clustering are available. In a top-down approach, a list of ascending thresholds is used to first detect large clusters, which are subsequently split into smaller groups as the threshold increases. The second option is a bottom-up method that processes a list of descending thresholds and thus first detects small strongly connected groups that are merged together in later iterations.

Overlapping clustering

Transitivity Clustering offers two approaches to achieve an overlapping cluster. Both methods use the results of a previously performed partitioning. The first approach calculates similarities between each element and each cluster, and then transforms these values into percentage values that reflect the cluster in which the element fits best. Subsequently, these values are used for fuzzy assignments of the elements to multiple clusters. The second approach applies a complex modification of the underlying graph model that is described in reference 9.

Integration of existing knowledge

To improve the clustering performance and to take advantage of existing knowledge, Transitivity Clustering allows forbidden and fixed element groupings. One can specify groups of elements that must be clustered together. The user can also directly set or prohibit pairwise assignments, as demonstrated in our example in Step 2B. Alternatively, you may set upper/lower bounds for the similarity such that pairs of elements with similarities above/below the bounds are set/prohibited to be in the same cluster.

© 2011 Nature America, Inc. All rights reserved.

BOX 2 | EXTENDED FUNCTIONS OF THE CLUSTEREXPLORER

The ClusterExplorer plug-in allows users to investigate their clustering results in order to answer typical biological questions. For a given clustering, you can choose to analyze single elements or clusters (corresponding to a protein or a protein family respectively). Analyzing one element can answer questions such as 'Which protein in the network is the most similar to my query protein?' or 'Which other protein families could my query protein also belong to?' The analysis of one cluster may help in finding a protein that best represents the underlying protein family. Furthermore, a single cluster analysis can be used to group protein families into superfamilies by computing the similarity of the cluster to all other clusters, as well as to merge the best putative matching families to superfamilies. With the plotting options, users can investigate the inter-/intra-cluster similarity distributions of a given clustering result and visualize the quality of the clustering result or estimate the best clustering threshold parameter from a gold standard distribution. The cluster size distribution may also provide additional insights into the quality of the clustering result, for instance, if a biologically reasonable size of the protein families is known. Finally, a clustering result can be compared with a gold standard, providing the user with a set of quality measures (**Box 4**) that can directly determine how well the elements were assigned to the respective clusters (i.e., the proteins to their protein families) in our application example.

© 2011 Nature America, Inc. All rights reserved.

npg

It is free, needs only minimal user input and offers comprehensive aid with the most crucial data clustering steps.

4. One crucial data clustering step is finding a meaningful density parameter, i.e., similarity threshold. In this protocol, we outline how a small 'gold standard' data set can be used for the semiautomatic estimation of this application- and problem-specific parameter. We also demonstrate how further background knowledge may be incorporated to increase the clustering performance.
5. Our Transitivity Clustering software offers a variety of additional functionalities: The Cytoscape plug-ins implement typical visual cluster analysis methods. The stand-alone version provides several functions to improve space and running time efficiency, as well as clustering accuracy. The applicability of upper bounds for node merging and hierarchical clustering, as well as overlapping clustering capabilities, may serve as examples here (see **Box 1** for more details).

Overview of the procedure

In the following section we give an overview of key steps in the protocol.

Step 2A: Integrated protein sequence cluster analysis with the Cytoscape plug-ins

The general aim here is to show how proteins can be accurately assigned to protein families, given their amino acid sequences and a small gold standard data set.

Researchers are provided with large amounts of proteins, which they would like to group into families and superfamilies to better understand their biological functions. However, little information on the functional relationships is known for most cases. Our clustering framework consisting of three Cytoscape plug-ins allows researchers to easily estimate optimal clustering parameters based on a small standard set.

We introduce the Cytoscape plug-ins, Blast2SimilarityGraph, TransClust and ClusterExplorer, and demonstrate how they can be used to group proteins into families. We use a data set by Brown *et al.*[13] consisting of 866 proteins that are manually assigned to protein families and superfamilies. We demonstrate that, by taking a subset consisting of one superfamily, we can estimate a reasonable clustering threshold. This gold standard set consists of 133 proteins of the vicinal oxygen chelate (VOC) superfamily, grouped into 15 protein families. We show that the parameters estimated from the small gold standard set result in very precise family assignments for the complete data set (i.e., the full set of 866 proteins). Furthermore, we demonstrate the power of the plug-ins to answer typical biological follow-up questions arising from protein sequence clustering results (see **Box 2**).

Step 2B: Semisupervised protein family detection with the Transitivity Clustering stand-alone software

The general aim here is to show that an incomplete gold standard may directly be incorporated with a clustering task to further improve accuracy.

Incomplete previous knowledge of the best cluster assignment can be used with transitivity clustering, for instance, to predict a meaningful threshold for the whole data set. In Step 2A, we used a well-structured gold standard: one superfamily. However, we may have further, but less structured, information about the expected clustering. If we know the cluster assignments for some of the objects from our data set, we can incorporate this additional knowledge. Transitivity Clustering is capable of partitioning a given data set in such a way that it ensures that objects that are known to belong to the same cluster are not separated into different groups during clustering. Note that using this knowledge affects and refines the accuracy of the cluster assignments of the other objects as well.

Here, we describe such a procedure for the stand-alone Transitivity Clustering software. The full Brown *et al.*[13] gold standard data set of 91 protein families is used. We assume to know 10% of the cluster (family) assignments (3,400 out of 34,000 pairwise assignments were chosen randomly). Thereafter, the optimal threshold that was obtained in the previous analysis is used, together with the randomly picked, incomplete gold standard, for integrated clustering. For this task, one of the extended functionalities of the Transitivity Clustering stand-alone software (see **Box 1**) is introduced. Finally, we show that by directly incorporating this background knowledge, even though it was unstructured and random, clustering performance can be increased.

Step 2C: Clustering of gene expression data

The general aim here is to partition gene expression data sets. In our example, we are given a list of microarray data sets for different cell samples to be identified into groups of samples with similar gene expression patterns.

BOX 3 | SIMILARITY FUNCTIONS

Amino acid sequences

The Basic Local Alignment Search Tool (BLAST)[16] is a commonly used tool to compare biological sequences. For each local alignment between subsequences of two proteins, a High Scoring Pair (HSP) is reported together with the start and end position of each subsequence within the protein, a score for the alignment and the *E*-value corresponding to the score. Note that the alignment may differ for the two possible directions because of the heuristic nature of BLAST. Further note that multiple HSPs may occur for the same pair of proteins for the same direction. Transitivity Clustering offers three main similarity functions that can be used given an all-versus-all BLAST result:

- Best bidirectional hit (BeH): This widely used method concentrates on the *E*-value of a single HSP. For both directions, one seeks the best hit, i.e., the HSP with the lowest *E*-value. To obtain a symmetric similarity function, the negative logarithm of the worst (largest) of the two *E*-values is taken as similarity measure between the two sequences.
- Sum of all hits (SoH): This approach is similar to BeH, but includes each identified HSP between two sequences. The pairwise symmetric similarity is the sum of all negative logarithms of the *E*-values of all HSPs between the two proteins. This may be particularly useful if multiple unconnected subsequences are important for the protein assignment.
- Normalized score: Another common approach to define a pairwise similarity between two sequences is the bit score reported by BLAST and normalized to the length of the HSP. We recommend filtering the list of HSPs for hits with reliable *E*-values; otherwise the normalization may lead to high similarities of dissimilar objects if a common subsequence is very short. As in the previously introduced measure, BeH, the maximal score for one direction is used if multiple HSPs exist. Symmetry of the similarity function is achieved by choosing the lower score of both directions.

Gene expression data

Gene expression data are usually represented by an expression matrix. Typically, the first step in each analysis is a technology-dependent data preprocessing, i.e., normalization, log-transformation, noise reduction, etc. Many tools have been developed and published for these tasks. Hence, we ignore this step and assume appropriately preprocessed data sets.

Transitivity Clustering now offers two options to compute pairwise similarity values from gene expression data:

- Pearson's correlation coefficient: This similarity function reflects the linear dependency between two vectors. The values reach from −1 to 1. Commonly, the absolute of this value is interpreted as the similarity measure, as 1 and −1 correspond to a perfect linear dependency whereas a value of 0 means no correlation between two expression profiles.
- Negative Euclidian distance: A standard measure of distance between two vectors is the Euclidian distance. As we seek similarity rather than distance, we use the negative of the Euclidian distances as similarity values.

© 2011 Nature America, Inc. All rights reserved.

Here we study the gene expression data of 38 bone marrow samples from acute leukemia patients with 999 monitored genes, processed with a Human Genome HU6800 Affymetrix microarray by Golub *et al.*[14] (data provided by Monti *et al.*[15]). The 38 samples comprise 11 cases of acute myeloid leukemia, 8 of T-lineage acute lymphoblastic leukemia and 19 of B-lineage acute lymphoblastic leukemia. Our goal is to reconstruct these leukemia classes by means of clustering the given gene expression data.

Starting with a gene expression matrix (38 samples × 999 genes), we take the log2 of the differential expression values and subsequently calculate a similarity for each pair of samples by means of Pearson's correlation coefficient (see **Box 3**). We describe in a step-by-step manner how to process gene expression data with the Transitivity Clustering stand-alone software. Furthermore, we give advice on how to choose a promising similarity threshold if some background knowledge about the number and size of the expected clusters is available.

MATERIALS

EQUIPMENT

- *Web browser:* For the Transitivity Clustering website, at a minimum, we recommend using Mozilla FireFox 2.0 or MS Internet Explorer 6.0 in Linux, Mac OSX, SunOS or Windows (XP, ME, 2000, VISTA or 7) operating systems.
- *Screen resolution:* Although it is not necessary, we further recommend a screen resolution of at least 1,024 × 768 pixels.
- *JavaScript*: For the FAQ page at the web site, JavaScript is used. It must be enabled within the web browser.
- *Java:* Transitivity Clustering is a Java program. At a minimum, Java 6 must be installed and configured properly for use with the web browser. Java is publicly available for free download at http://www.java.com/. Java Web Start must be enabled and configured with the web browser.
- *Cytoscape:* To use the Cytoscape plug-ins of Transitivity Clustering, Cytoscape release 2.6 (or newer version), needs to be installed and configured properly. This software, as well as instructions for its installation, is available from http://www.cytoscape.org/.
- *Data*: All data sets necessary for this protocol are provided as supplementary files (**Supplementary Data 1–9**). For a detailed description of the file formats, please refer to the Transitivity Clustering web site at http://transclust.cebitec.uni-bielefeld.de/.

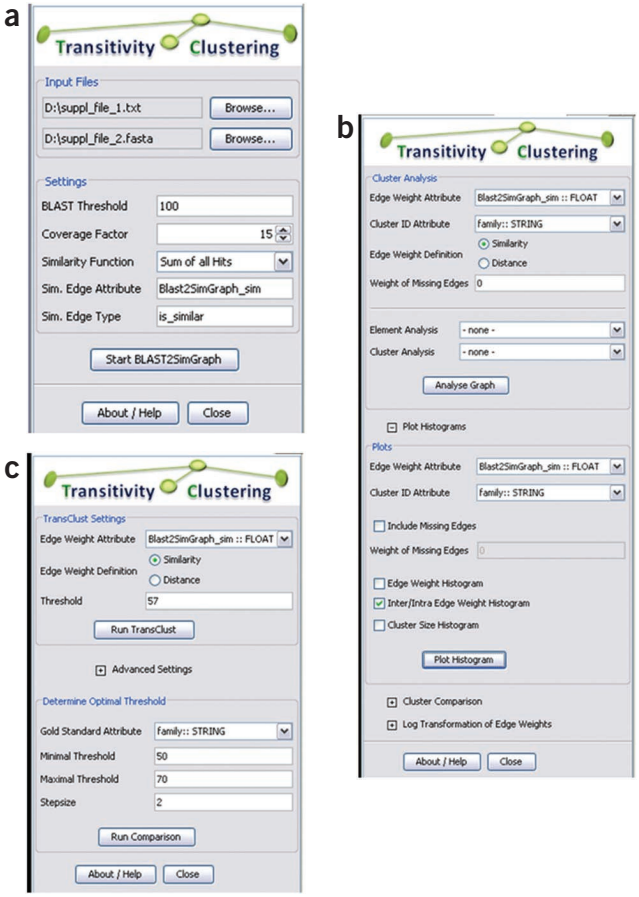

Figure 2 | Cytoscape plug-ins. The screenshots show the user interfaces of the three Cytoscape plug-ins. The parameter settings correspond to the settings applied in Step 2A. (**a**–**c**) Blast2SimilarityGraph GUI (**a**), ClusterExplorer GUI with expanded 'Plot Histograms' panel (**b**) and Transitivity Clustering GUI (**c**).

PROCEDURE

1| Download all supplementary data files (**Supplementary Data 1–9**) and save them on a hard disk.

2| Depending on the biological question in which you are interested, choose from the following three options: integrated protein sequence cluster analysis with Cytoscape (option A), semisupervised protein family detection (option B), or clustering of gene expression data (option C).

(A) Integrated protein sequence cluster analysis with Cytoscape plug-ins ● **TIMING ~30 min**

(i) Start Cytoscape.

(ii) Download and install the clustering plug-ins with the Cytoscape plug-in manager: select *Plug-ins* in the Cytoscape menu and click on *Manage Plug-ins*. A window will pop up and list the plug-ins that are available for installation. Open the *Analysis* folder in the *Available for Install* section by clicking on the '+' sign. Select *Blast2SimilarityGraph v1.01* and hit the *Install* button. Do the same for *clusterExplorerPlugin v1.01* and for *TransClust v1.01*. Close the plug-in manager.

? TROUBLESHOOTING

(iii) In this example, we analyze and cluster protein sequences. To calculate sequence similarities and import them into Cytoscape we use the Blast2SimilarityGraph plug-in. Click on the *Plug-ins* menu in Cytoscape. Select *Blast2SimilarityGraph* in the menu items and click on *Start Blast2SimilarityGraph GUI*. The graphical user interface (GUI) is opened in a panel on the left side of the Cytoscape Desktop.

(iv) We first wish to import the network of a subset of the data for which we assume to know the assignment into protein families. To load the respective BLAST file, click on the *Browse*... button in the GUI. A file chooser window will open. Navigate to the directory where you saved the Supplementary files and choose **Supplementary Data 1**. Confirm the selection using the *Open* button in the file chooser. Next, load the FASTA file by clicking the *Browse*... button in the Blast2SimilarityGraph GUI. Select **Supplementary Data 2** and confirm with the *Open* button.

(v) Set the parameters in the Blast2SimilarityGraph GUI as follows: *BLAST Threshold* = 100, *Coverage Factor* = 15, *Similarity Function* = '*Sum of all Hits*'. Maintain the other parameters unchanged (**Fig. 2a**). Thereafter click the *Start Blast2SimGraph* button to create the network. The network is automatically displayed on the right side of the Cytoscape window (**Supplementary Fig. 1**).

(vi) Import the gold standard family assignments for the VOC superfamily into Cytoscape from **Supplementary Data 3**. Select *File* → *Import* → *Attribute from Table* (*Text/MS Excel*)... An import window will open. Click on the *Select File(s)* button and navigate to the directory where the Supplementary data files are stored. Select **Supplementary Data 3** and confirm using the *Open* button. The data are now shown in the import window. Rename *Column 2* by right-clicking to '*family*' and confirm with *Ok*. Thereafter, click on the *Import* button to map the assignments to the similarity network and close the import window with *Close*.

(vii) Next, we use the ClusterExplorer plug-in to identify the best similarity threshold for the selected subset. Open the ClusterExplorer plug-in with the *Plug-ins* menu → *ClusterExplorer* → *Start ClusterExplorer GUI*. The GUI will be displayed in the panel on the left side of the Cytoscape window.

(viii) The ClusterExplorer plug-in can illustrate the distribution of similarities within (intra) and between (inter) gold standard clusters. This first step assists in identifying a range of thresholds that are suitable to cluster the sequences close to the optimal assignment. Open the *Plot Histograms* panel by clicking on the '+' sign. Set the parameters in this panel as follows: *Edge Weight Attribute* = '*Blast2SimGraph_sim*::*FLOAT*', *Cluster ID Attribute* = '*family*::*STRING*'. Then

© 2011 Nature America, Inc. All rights reserved.

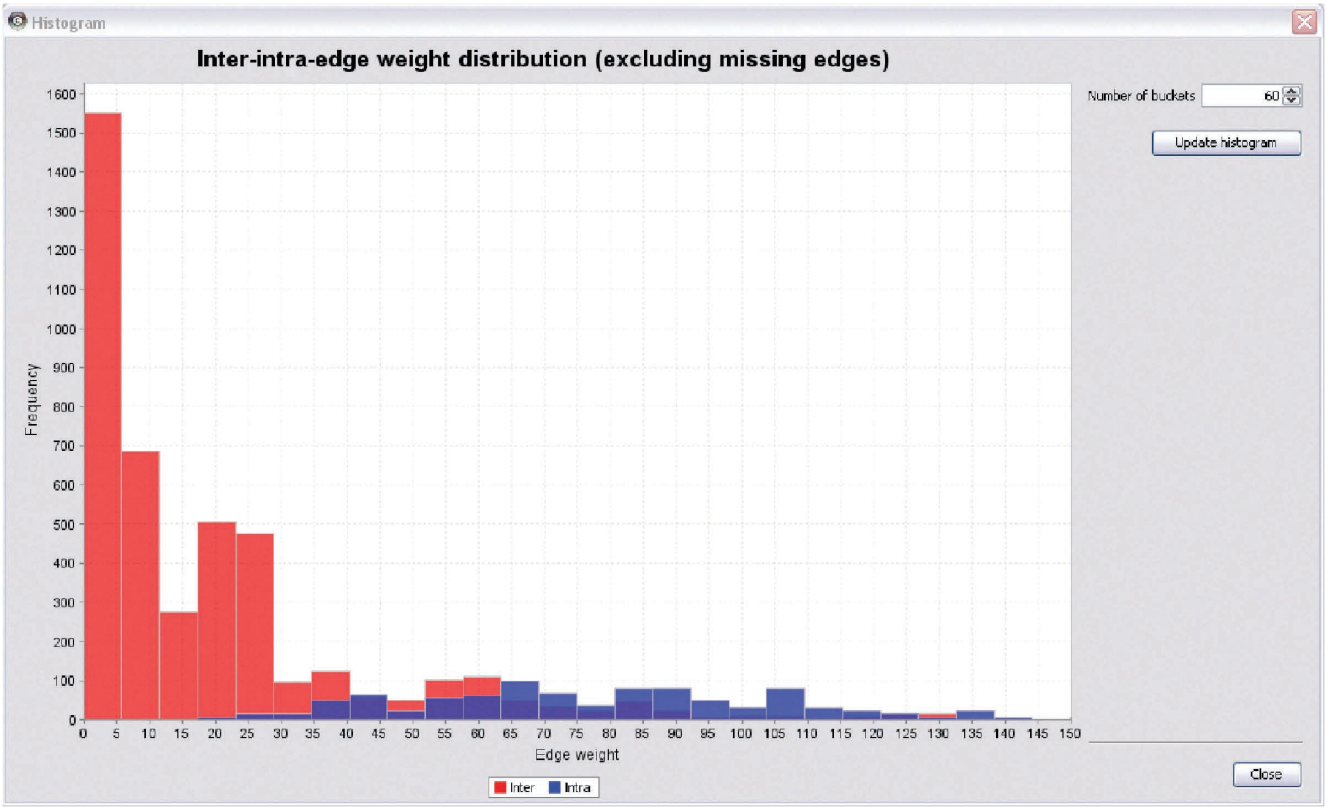

Figure 3 | Intra- versus inter-cluster similarity distributions. The histogram shows the inter-cluster similarity distribution (red) and intra-cluster similarity distribution (blue) of the gold standard clustering (family assignment). The most reasonable density threshold ranges between 50 and 70 as it separates the inter-cluster similarity distributions from the intra-cluster similarity distributions in the best way.

check the box for *Inter/Intra Edge Weight Histogram* and click on the *Plot Histogram* button (**Fig. 2b**). A new window will pop up and show the inter-edge weight distribution in red and the intra-edge weight distribution in blue (**Fig. 3** and **Supplementary Fig. 2**).

(ix) The histogram shows that the best similarity threshold is somewhere in the range of 50–70, well separating the inter- and intra-similarity distributions (**Fig. 3**). Exit this view via *Close*. The exact best threshold can be determined by the Transitivity Clustering plug-in. Open the TransClust plug-in via the *Plug-ins* menu → *TransClust* → *Start TransClust*. The GUI will be displayed in the panel on the left side of the Cytoscape window.

(x) Transitivity Clustering provides the option to determine the optimal density threshold in a given range if, as in our example, a gold standard is available. For each threshold in this range, the clustering routine is executed and the results are subsequently compared with the optimal partitioning. To do so, set the TransClust parameters as follows: *Edge Weight Attribute* = '*Blast2SimGraph_sim*::*FLOAT*', *Gold Standard Attribute* = '*family*::*STRING*', *Minimal Threshold* = 50, *Maximal Threshold* = 70, *Stepsize* = 1. Then, click on the *Run Comparison* button.

(xi) Once the comparison is made, a results panel appears on the right side of the Cytoscape window. The table inside lists for each threshold the quality of the clustering results by means of recall, precision and F-measure (see **Box 4** for details about the quality functions). The results suggest that a threshold of 57 is optimal for the gold standard network as it has the highest F-measure. Now set the threshold to 57 in the Transitivity Clustering settings and click on the *Run TransClust* button (**Fig. 2c**). The network is now clustered and visualized according to the clustering results (**Supplementary Fig. 3**).

(xii) We next aim for clustering the complete Brown *et al.*[13] data set of 866 protein sequences with the putatively best threshold of 57. To do so, open the *Blast2SimilarityGraph* GUI as in Step 2A(iii). Load **Supplementary Data 4** as a BLAST file and **Supplementary Data 5** as a FASTA file and set the parameters as described in Step 2A(iv,v).

? TROUBLESHOOTING

(xiii) To cluster the created similarity network by using the threshold of 57 obtained in Step 2A(xi), open the TransClust GUI as described in Step 2A(ix). Set the parameters *Edge Weight Attribute* = '*Blast2SimGraph_sim*::*FLOAT*', Threshold = 57 and click on the *Run TransClust* button. Wait for the clustering to complete. A progress bar will inform you about the status of the clustering process. After completion, the nodes in the network will be arranged circularly, with one ring for each cluster.

© 2011 Nature America, Inc. All rights reserved.

BOX 4 | QUALITY FUNCTIONS

To judge the quality of a clustering result we can compare the set of clusters to a set of groups from a given gold standard. For Transitivity Clustering, we offer the following functions: precision and recall are measures between 0 and 1 that correspond to fidelity and completeness of a classification; i.e., they reflect how exact and how complete the obtained clustering results are if compared with a given gold standard. Values near 1 indicate a good result, values near 0 a poor result. If the clustering is identical to the gold standard, both precision and recall equal 1. The F-measure is an equal combination of both. Hence, it is a general measure for the clustering quality. Consequently, the F-measure provides values between 0 and 1, where 1 is the best. The main problem when comparing two group assignments is to decide which clusters should be compared with which. Generally, we compare each cluster from the gold standard with the cluster from the results for which we achieve the highest F-measure. In Cytoscape, Transitivity Clustering also offers another F-measure (the 'F-measure I'), in which each cluster from the gold standard is compared with the cluster from the results with the maximal number of common elements. We generally recommend using the F-measure (notated as 'F-Measure II' within Cytoscape) as the key measure of quality. Note that the computation of all quality measures only incorporates those objects that are present in both the clustering and the gold standard.

(xiv) To evaluate how well the density threshold works on the complete data set, load the gold standard family assignments (**Supplementary Data 6**) for the complete data set into Cytoscape as described in Step 2A(vi). Rename *Column 2* by right-clicking to '*brown_family*'.

(xv) Now open the ClusterExplorer GUI as described in Step 2A(vii) and open the *Cluster Comparison* panel by clicking on the '+' sign. Select the '*brown_family*' as the *Gold Standard Attribute* and '*TransClust::Integer*' as the *Cluster ID Attribute*. Then click on the *Run Comparison* button. A panel shows up on the right side of the Cytoscape window, which contains the calculated quality function (see **Box 4**) that reflects how close the produced clusters are to the actual protein families.

(xvi) To complete this process, click *File* → *Quit*. In the windows that appear, confirm that you do not wish to save the session with '*No, just quit*'.

▲ **CRITICAL STEP** If you wish to save the Cytoscape session, which includes the results, click *Yes*, specify a directory, enter a file name and click *Save*.

(B) Semisupervised protein family detection with the Transitivity Clustering stand-alone software ● **TIMING ~20 min**

(i) Start the web browser and navigate to the homepage of Transitivity Clustering (http://transclust.cebitec.uni-bielefeld.de/).

(ii) Transitivity Clustering can be started from this site via Java Web Start after choosing the amount of RAM to be reserved for the program. The example we describe here requires only minimal memory. Hence, start the stand-alone version as Java Web Start application by clicking on the '*256MB*' link in the appropriate section. A window will pop up and ask to either save the file or open it via Java Web Start. Choose the latter and confirm with *OK*.

? TROUBLESHOOTING

(iii) As soon as the application is started, a pop-up window will appear and require that you either specify a temporary directory or use the default location. Press *Yes* to choose the default location. Transitivity Clustering will create a directory in the system-specific temporary directory and assign a name that contains the date and time of its creation. Wait for the main GUI to appear (see **Fig. 4** for an illustration).

(iv) The Transitivity Clustering software allows you to import either a precalculated file containing all pairwise similarities or to calculate the similarities of protein sequences given a BLAST and a FASTA file (see **Box 3** for a description of the available similarity functions for amino acid sequences). Here, we cluster the same set of 866 enzymes from Brown *et al.*[13] as was previously used in Step 2A. Click on *File* → *Load* → *BLAST/FASTA files* to start the import process.

(v) This will open a file chooser window. Navigate to the directory where you previously saved the Supplementary files (Step 1) and select **Supplementary Data 4**. Confirm your choice by clicking on *Choose BLAST file*. A second file chooser will appear that requires you to specify the corresponding FASTA file. Again, navigate to the directory containing the Supplementary files and select **Supplementary Data 5**. Confirm by selecting *Choose FASTA file*.

(vi) In the next pop-up window you have to choose between different similarity functions (see **Box 3** for a description of the available similarity functions). Further, you must specify the cutoff that was used for creating the BLAST file. Here, we use the same configuration as in Step 2A, which is 'Sum of Hits' with a coverage factor of 15. Check *SoHcoverage* in the upper row, set *BLAST cutoff* = 100 and *Coverage factor* = 15. Press *start* to begin the import.

(vii) A sequence of progress bars will keep you updated about the status of the import process. The FASTA and BLAST files will be read, similarities will be calculated and stored in a similarity file in the temporary directory (as specified in Step 2B(iii)). As soon as the import process is completed you will see new tabs appearing on the right side of the user interface. They provide a preview of the imported files (**Fig. 4b**).

© 2011 Nature America, Inc. All rights reserved.

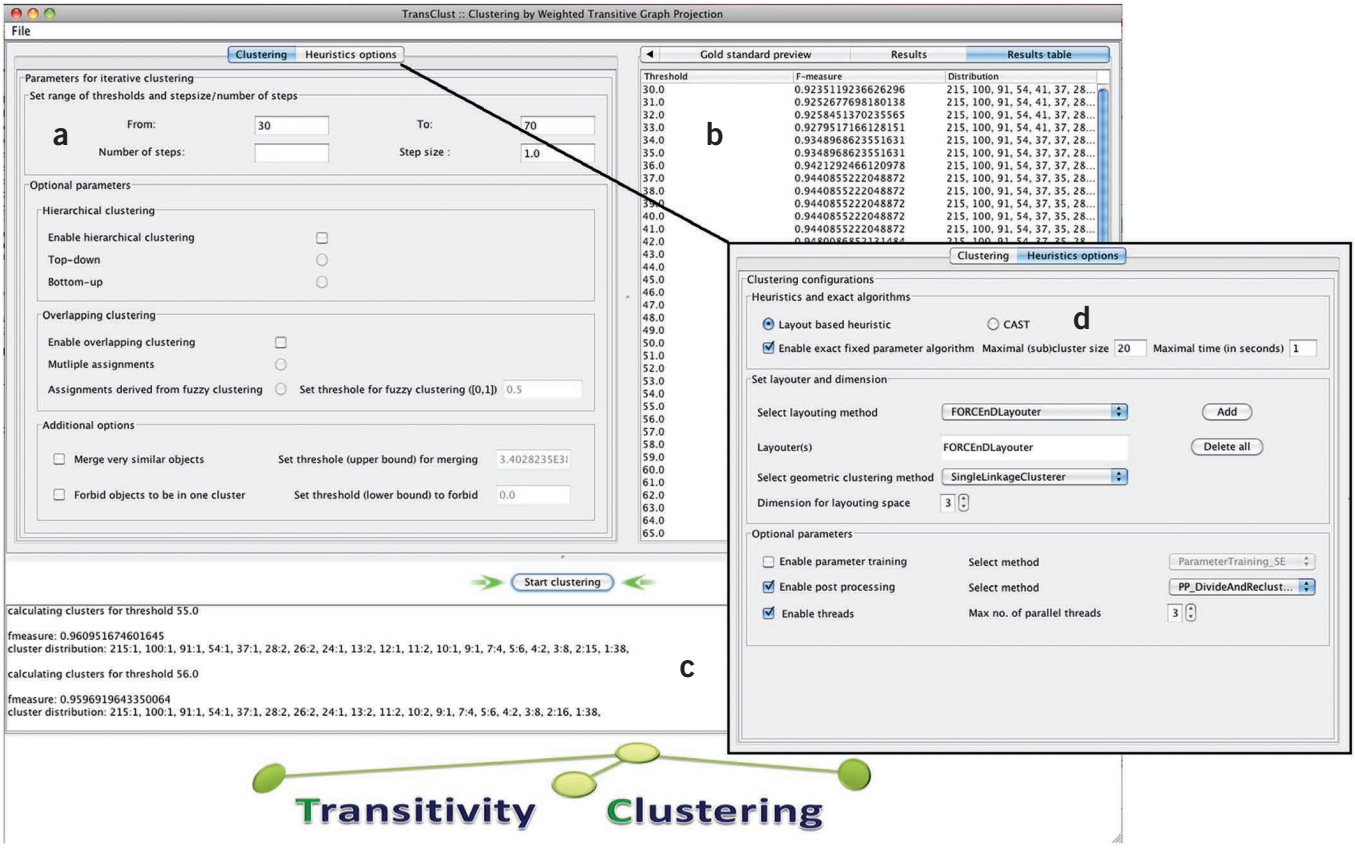

Figure 4 | Stand-alone software user interface. (**a**–**d**) The screenshot shows the stand-alone version of Transitivity Clustering, with all its components. In panel **a**, threshold as well as additional clustering features can be chosen. Panel **b** displays all files that are currently loaded into Transitivity Clustering, together with the results of the last clustering run. The progress bar (**c**) reports the status of the current process. Panel **d** allows the adjustment of heuristic and exact methods.

(viii) The Transitivity Clustering stand-alone application is capable of using known cluster assignments for parts of the input data in order to improve the clustering accuracy. To load such a file, click *File* → *Load* → *Known assignments*.

(ix) A file chooser window will open. Navigate to the directory specified in Step 1, choose **Supplementary Data 7** and confirm your choice by pressing *Choose file with known assignment*. A preview of this file will appear in a tab on the right side.

(x) *Optional step*: As with the Cytoscape plug-ins from option 2A, the stand-alone application of Transitivity Clustering is capable of comparing a previously obtained clustering result against a gold standard assignment by means of F-measure (see **Box 4**). To load gold standard family assignments, go to *File* → *Load* → *Gold standard file* and choose **Supplementary Data 6**.

(xi) The *clustering* tab on the left side of the user interface lets you specify the following: first, a range of thresholds for the subsequent clustering; second, whether to use the extended clustering options (hierarchical or overlapping); and third, whether an lower/upper bound should be used (see **Box 1**). Here, we aim for partitioning data using the threshold of 57 that was identified in Step 2A. In addition, we wish to guarantee the specified, known assignments uploaded in Step 2B(ix). Accordingly, set *From* = 57 and *To* = 57 in the corresponding text boxes and leave all remaining fields unmarked (compare with **Fig. 4**).

(xii) To execute the clustering, click on the *Start clustering* button in the center. The progress is reported in the status frame at the bottom of the interface.

? TROUBLESHOOTING

(xiii) Once the clustering process is completed, click the *Results* tab on the right side; this will provide you with a preview of the results file. The file contains the used threshold, the F-measure (if you chose to import the gold standard file in optional Step 2B(x)) and the partitioning itself. Clusters are separated by semicolon and elements within the clusters by comma. This file can also be found in the temporary directory that was created in Step 2B(iii). Now, click on the *Results table* tab. The table that appears shows the cluster assignment for each element in a table-based style.

(xiv) To complete this procedure, close the window or go to *File* → *Exit*. A window will pop up and ask whether to delete the temporary directory that has been created for this session. Confirm by clicking *Yes*.

© 2011 Nature America, Inc. All rights reserved.

▲ CRITICAL STEP If you want to save the results or the created similarity file you should either choose *No* in this step or copy the files to a different location before closing the program.

(C) Clustering of gene expression data ● TIMING ~15 min

(i) Repeat Step 2B(i–iii) to start the Transitivity Clustering stand-alone software.

? TROUBLESHOOTING

(ii) Here, we want to use the gene expression data clustering functionality. Therefore, go to *File* → *Load* → *Expression data*.

(iii) A file chooser window appears. Navigate to the directory specified in Step 1, choose **Supplementary Data 8** and confirm your choice by selecting *Choose expression matrix*.

(iv) Another pop-up window will appear. Here, you can choose a similarity function (see **Box 3** for a description of the available similarity functions for gene expression data) and whether the first row/column contains header information. In this case, the first column contains information about the samples, and the first row denotes the probe IDs. Hence, mark both check boxes, select *Pearson's correlation coefficient* and press *start*.

(v) When the computation of pairwise similarities is completed you will see a preview of the corresponding similarity file on the right side of the user interface. Transitivity Clustering illustrates the distribution of similarities. To show the respective histogram, click on the *Similarity Distribution* tab on the right side of the user interface. Note that the similarities vary between 0.3 and 1.

(vi) *Optional*: To compare the clustering results against a gold standard (in this case, the three leukemia phenotypes), go to *File* → *Load* → *Gold standard file* and navigate to the directory specified in Step 1 in the file chooser. Select **Supplementary Data 9** and confirm with *Choose gold standard file*. Note that the histogram is updated with the gold standard information and now illustrates the intra- versus inter-cluster similarity distributions. Further note the overlap of both distributions in the area between 0.5 and 0.75. Consequently, we expect to find a 'good' partitioning for similarity thresholds in this area.

(vii) To cluster data, specify the range of thresholds that should be used. In Step 2C(v) we saw that the similarities are between 0.3 and 1. Accordingly, set *From* = 0.3 and *To* = 1 in the corresponding text boxes to use the whole range of similarities. Further, set the *Number of steps* = 100.

(viii) To start the clustering process with the previously selected specification, press the *Start clustering* button.

? TROUBLESHOOTING

(ix) Wait for the iterative clustering to be completed. You can follow the progress of the clustering in the console panel (see **Fig. 4c**). Navigate to the *Results* tab to see a preview of the clustering result file. For each of the 100 thresholds, Transitivity Clustering shows the F-measure in one column (if compared with a gold standard as described in optional Step 2C(vi)), and the clustering result itself in the last column. Clusters are separated by semicolons, and elements within the clusters by commas.

(x) Navigate to the *Results table* tab, which appears right next to the *Results* tab. This table illustrates the results similarly to the previous tab, except that cluster sizes are shown instead of actual elements. If we have some knowledge about the expected number of clusters or the distribution of cluster sizes, this overview assists in identifying the most promising similarity threshold. In our case, we expected to have three almost equally sized clusters. The clustering result obtained with a threshold of 0.55 is closest to this assumption with a partitioning into three clusters of sizes 19, 11 and 8.

(xi) To complete this procedure, follow the instructions described in Step 2B(xiv).

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 1**.

TABLE 1 | Troubleshooting table.

Step	Problem	Possible reason	Solution
2A(ii)	The plug-ins are not listed in the installable plug-in directory	Cytoscape version not supported	Transitivity Clustering can be used with Cytoscape 2.5–2.7. Download and install the most current Cytoscape version. If the plug-ins are still not found, go to the Transitivity Clustering website, download the plug-ins manually and save them into the Cytoscape plug-in directory
		The plug-ins are already installed	Cytoscape stores installed plug-ins locally on your hard drive. To ensure that you have the most recent version of each plug-in, use the plug-in manager and first remove each plug-in and subsequently re-install them as described in Step 2A(ii)

(continued)

© 2011 Nature America, Inc. All rights reserved.

TABLE 1 | Troubleshooting table (continued).

Step	Problem	Possible reason	Solution
2A(xii)	The import process takes a very long time or seems to be frozen	Cytoscape runs out of memory	Assign more memory to the Cytoscape program. One way to do so is to start Cytoscape from the command line and use the –Xmx option. Therefore, open a terminal, navigate to the Cytoscape directory and type 'java –Xmx1G cytoscape.jar' to start Cytoscape with 1 GB of memory. For alternative ways to increase memory, see the Cytoscape wiki at http://cytoscape.wodaklab.org/wiki/How_to_increase_memory_for_Cytoscape
2B(ii) and 2C(i)	The Transitivity Clustering stand-alone version does not start	Java may not be installed or inactivated in your web browser	Install and configure Java from the home page at http://www.java.com/
		The certificate was not accepted	Navigate back and accept the certificate
2B(xii) and 2C(viii)	The *Start clustering* button is not visible	The screen resolution is too low	Increase the screen resolution
		The window is too small	Maximize the window and/or change the size of the different panels (see **Fig. 4**) by clicking on the line between them and moving the mouse while keeping the left mouse button pressed

© 2011 Nature America, Inc. All rights reserved.

● TIMING

The time required to execute this protocol is mainly related to the size of data sets and central processing unit (CPU) power. Download times for the software as well as for the supplementary files are negligible (<1 min). An experienced user can execute the protocol within 45 min to 1 h 15 min.

Steps 1 and 2A, integrated protein sequence cluster analysis with Cytoscape plug-ins: ~30 min

Steps 1 and 2B, semisupervised protein family detection with Transitivity Clustering stand-alone software: ~20 min

Steps 1 and 2C, clustering of gene expression data: ~15 min

ANTICIPATED RESULTS

Here, we discuss the concrete results that can be achieved by executing the three options in this protocol.

Step 2A: Integrated protein sequence cluster analysis with Cytoscape plug-ins

The general aim of this procedure is to demonstrate how proteins can be automatically and accurately assigned to protein families using the Cytoscape plug-in framework.

We assume that the only available information on the proteins of interest are the sequences, the BLAST all-versus-all results and a small hand-curated gold standard data set containing proteins and family assignments. In this case, the gold standard set consists of 133 proteins from the Brown *et al.*[13] data set, which are assigned to the VOC superfamily and have been manually assigned to 15 protein families; this data set is referred to as the GS data set. We applied the three Cytoscape plug-ins of Transitivity Clustering to demonstrate the construction and visualization of a similarity network (Blast2SimilarityGraph plug-in), the estimation of a promising region for a reasonable threshold (ClusterExplorer plug-in) and the automatic identification of the optimal threshold as well as the optimal clustering (Transitivity Clustering plug-in).

According to the intra- versus inter-protein family similarity distribution, the best threshold is found somewhere between 50 and 70. Through threshold determination we identify the optimal threshold to be ~57. When we use this parameter to the complete Brown *et al.*[13] data set, we obtain an F-measure of ~0.91, highlighting that the threshold chosen according to the GS data set is not only transferable to the complete data set but also provides results of high accuracy. As an example, the enolase superfamily consists of 286 proteins and has been manually assigned to nine protein families. Investigating the Transitivity Clustering results, we find the proteins of this superfamily to be separated into 11 protein clusters, corresponding to 11 protein families instead of 9. However, a comparison of the cluster annotations and family assignments, which are both stored in the Node Attribute Browser of Cytoscape, reveals that seven of the nine hand-curated protein family assignments perfectly match the protein clusters, which means that seven protein families have been reconstructed accurately. The two remaining protein families have not been reassigned correctly. One of the two protein families is the 'dipeptide epimerase' family. A closer investigation of these two families by utilizing the Cytoscape visualization features reveals that the family contains only two proteins that are split into singleton clusters, i.e., one cluster for each of the two proteins.

However, using the '*Element Analysis*', which is provided with the ClusterExplorer plug-in (see **Box 2** for more details), we discover that these two singleton clusters are the most similar among the complete data set, indicating that the two proteins were close to being merged into one cluster.

Step 2B: Semisupervised protein family detection with the Transitivity Clustering stand-alone software

Here the goal is to demonstrate that incomplete gold standard data may directly be incorporated with Transitivity Clustering to further improve accuracy.

We assume that we have given somewhat unstructured previous knowledge, i.e., we merely know about rather 'random' protein-cluster assignments instead of, e.g., the protein family annotations for one complete, well-researched protein superfamily. To simulate this, we randomly picked 10% of the proteins along with their family annotation from the Brown *et al.*[13] gold standard. Together with the whole gold standard data set we loaded this information into Transitivity Clustering. Subsequently, we used the previously determined optimal threshold of 57 for clustering and obtained 90 clusters of putative protein families. As we actually know the real clustering from Brown *et al.*,[13] we can evaluate this procedure. By iteratively clustering for varying thresholds between 30 and 70 with Transitivity Clustering, we find that the optimal result is now achieved with a threshold of 55 with an increased F-measure (now 0.96, before 0.91). We further observe that the F-measures for partitionings 'around' the threshold of 55 are higher than before (data not shown). These improvements in robustness and clustering accuracy are clearly due to use of the additional background knowledge. Transitivity Clustering guarantees that those 10% of the edges that are known to be true are not removed during clustering, i.e., elements known to be in the same cluster are not separated. Note that this also increases the chances that other proteins will be assigned to correct clusters.

Step 2C: Clustering of gene expression data

We aim to demonstrate the capability of Transitivity Clustering to cluster any kind of data set in which a pairwise similarity function is available. Here we partition a set of 38 different leukemia cell samples[14] into three classes with a similarity function that is based on the gene expression profiles of 999 genes.

In the first step, we calculate the pairwise Pearson correlation coefficient of the log2-transformed gene expression values. Subsequently, we use these values as similarity matrix for Transitivity Clustering. From the similarity distribution we see values between 0.3 and 1. Iterative application of Transitivity Clustering shows that we receive three clusters of sizes 19, 11 and 8 elements for a threshold of 0.55. Under the assumption that we expect three almost equally sized clusters, this is the closest result for all possible clustering thresholds. In fact, the partitioning is identical to the phenotypes associated with the samples in ref. 14, i.e., 19 samples for B-lineage acute lymphoblastic leukemia, 11 samples of acute myeloid leukemia and 8 cases of T-lineage acute lymphoblastic leukemia.

© 2011 Nature America, Inc. All rights reserved.

Note: Supplementary information is available via the HTML version of this article.

ACKNOWLEDGMENTS J.B. thanks the German Academic Exchange Service (DAAD) for funding his work at ICSI, Berkeley. J.B. and M.A. are grateful for support from the German Research Foundation (DFG)-funded Cluster of Excellence for Multimodal Computing and Interaction. D.E. and M.A. received funding from the German National Genome Research Network. T.W. gained financial support through NIH grant NIH R01 LM009722 and the Buck Trust.

AUTHOR CONTRIBUTIONS T.W. and J.B. collected data, and tested and wrote Step 2A. D.E. and M.A. prepared Step 2B. A.T. and S.B. were responsible for Step 2C. All authors contributed to the preparation and proofreading of all other parts of the manuscript.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

Published online at http://www.natureprotocols.com/.
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/.

1. Enright, A.J., Kunin, V. & Ouzounis, C.A. Protein families and TRIBES in genome sequence space. *Nucleic Acids Res.* **31**, 4632–4638 (2003).
2. Enright, A.J., Van Dongen, S. & Ouzounis, C.A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30**, 1575–1584 (2002).
3. Krause, A., Stoye, J. & Vingron, M. Large scale hierarchical clustering of protein sequences. *BMC Bioinformatics* **6**, 15 (2005).
4. Enright, A.J. & Ouzounis, C.A. GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics* **16**, 451–457 (2000).
5. Paccanaro, A., Casbon, J.A. & Saqi, M.A. Spectral clustering of protein sequences. *Nucleic Acids Res.* **34**, 1571–1580 (2006).
6. Frey, B.J. & Dueck, D. Clustering by passing messages between data points. *Science* **315**, 972–976 (2007).
7. Wittkop, T. *et al.* Partitioning biological data with transitivity clustering. *Nat. Methods* **7**, 419–420 (2010).
8. Cline, M.S. *et al.* Integration of biological networks and gene expression data using Cytoscape. *Nat. Protoc.* **2**, 2366–2382 (2007).
9. Wittkop, T. *Transitivity Clustering: Clustering Biological Data by Unraveling Hidden Transitive Substructures*, 148 (Suedwestdeutscher Verlag fuer Hochschulschriften, 2010).
10. Rahmann, S. *et al.* Exact and heuristic algorithms for weighted cluster editing. *Comput. Syst. Bioinformatics Conf.* **6**, 391–401 (2007).
11. Wittkop, T., Baumbach, J., Lobo, F.P. & Rahmann, S. Large scale clustering of protein sequences with FORCE—a layout based heuristic for weighted cluster editing. *BMC Bioinformatics* **8**, 396 (2007).
12. Böcker, S., Briesemeister, S. & Klau, G.W. Exact algorithms for cluster editing: evaluation and experiments. *Algorithmica* (in press) (2009).
13. Brown, S.D., Gerlt, J.A., Seffernick, J.L. & Babbitt, P.C. A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biol.* **7**, R8 (2006).
14. Golub, T.R. *et al.* Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–537 (1999).
15. Monti, S., Tamayo, P., Mesirov, J. & Golub, T. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning* **52**, 91–118 (2003).
16. Altschul, S.F. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.