Bioinformatics Advance Access published August 2, 2010

Multi-netclust: an efficient tool for finding connected clusters in multi-parametric networks

Arnold Kuzniar[1,2,†], Somdutta Dhir[3,†], Harm Nijveen[1,2], Sándor Pongor[3,4] and Jack A.M. Leunissen[1,2,*]

[1]Laboratory of Bioinformatics, Wageningen University and Research Centre, PO Box 569, 6700 AN Wageningen, The Netherlands;

[2]The Netherlands Bioinformatics Centre, PO Box 9101, 6500 HB Nijmegen, The Netherlands;

[3]Protein Structure and Bioinformatics, International Centre for Genetic Engineering and Biotechnology, Padriciano 99, I-34012 Trieste, Italy;

[4]Bioinformatics Group, Biological Research Centre, Hungarian Academy of Sciences, Temesvári krt. 62, H-6701 Szeged, Hungary

Associate Editor: Dr. Jonathan Wren

ABSTRACT

Summary: Multi-netclust is a simple tool that allows users to extract connected clusters of data represented by different networks given in the form of matrices. The tool uses user-defined threshold values to combine the matrices, and uses a straightforward, memory-efficient graph algorithm to find clusters that are connected in all or in either of the networks. The tool is written in C/C++ and is available either as a form-based or as a command-line based program running on Linux platforms. The algorithm is fast, processing a network of more than 10^6 nodes and 10^8 edges takes only a few minutes on an ordinary computer.

Supplementary Materials: http://www.bioinformatics.nl/netclust/

Contact: jack.leunissen@wur.nl

1 INTRODUCTION

Finding tightly connected clusters in large data sets is a frequent task in many areas of bioinformatics such as the analysis of protein similarity networks, microarray or protein-protein interaction data. Classical clustering algorithms have difficulties in handling large data sets used in bioinformatics owing to high demands on computer resources. Fast heuristic algorithms have been developed for specific tasks, for example BLASTClust from the NCBI-BLAST package (Altschul, et al., 1990), Tribe-MCL (Enright, et al., 2002) or the CD-HIT (Li and Godzik, 2006) can delineate protein or gene families in a large network of sequence similarities (e.g. BLAST E-values). However, there are no apparent tools that could efficiently handle large multiple networks, such as those necessary to group proteins using more than one similarity criterion (e.g. based on sequence, structure or function) (Fig. 1A).

We developed an efficient, semi-supervised tool that takes the users' empirical knowledge of cutoff values into account (below which interactions or similarities can be neglected) to combine multiple data networks using an averaging or kernel fusion method (Kittler, et al., 1998). The resulting combined network can then be queried for connected components (clusters) using an efficient implementation of the union-find algorithm (Tarjan, 1975). The clusters correspond to groups of nodes that are connected either by any or by all of the constituent networks, depending on the aggregation rule used (Fig. 1, B and C, respectively). In order to adapt this method to large heterogeneous data sets, we combined the thresholding, aggregation as well as the connected component search into a single, memory- and time-efficient tool, Multi-netclust. Multi-netclust is not a new clustering method but an optimized implementation of existing graph algorithms suitable for handling large networks of more than 10^6 nodes and 10^8 edges.

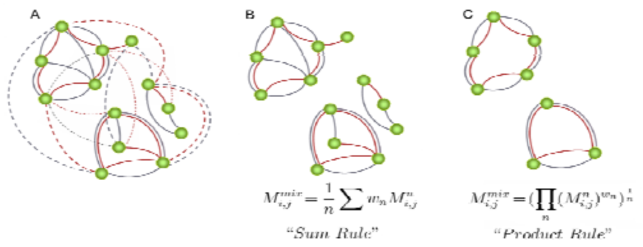
A
B
C
$M_{i,j}^{mix} = \frac{1}{n} \sum w_n M_{i,j}^n$
"Sum Rule"
$M_{i,j}^{mix} = (\prod_n (M_{i,j}^n)^{w_n})^{\frac{1}{n}}$
"Product Rule"

Fig. 1. The principle of Multi-netclust is illustrated on a two-parameter network. Red and grey edges correspond to distinct similarity data (A). Dotted lines denote edges that are below the respective threshold and hence can be omitted from the networks. Two different aggregation rules are implemented: the weighted arithmetic averaging ("sum rule") gives clusters that are connected within either of the two networks (B); the weighted geometric averaging ("product rule") gives clusters that are connected within both networks (C). M_{ij} denotes the value assigned to the edges, w is the weighting factor ("alpha") of the two matrices (hence n=2), and M^{mix} refers to the aggregated matrix.

[*]To whom correspondence should be addressed.

[†] The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

2 MULTI-NETCLUST INPUT AND OUTPUT

Multi-netclust uses external-memory rather than the in-core approach (Chiang, 1995) for matrix manipulations so that the size of the data sets is not a limiting factor. The input to Multi-netclust are networks given in sparse matrix format, as well as the aggregation rule, “alpha” weighting factor, and similarity (or distance) cutoff value(s) associated with a processing step(s). Generally, the product rule results in more reliable connections confirmed by multiple data sources whereas the sum rule expands the network with new (not necessarily reliable) connections. Setting the “alpha” value for each matrix provides means, for example, to weight the reliability of different data sources or to decide which data set is more likely to contribute with new (additional) information. A permissive cutoff value usually results in a few large clusters while a strict cutoff value tend to produce many small (singleton) clusters. The data can be entered either via a CGI interface, or from the command line. The output of Multi-netclust is a list of the connected clusters given in a structured text format.

Multi-netclust is written in the C/C++ language, and the CGI interface is a Perl script. The source code, sample data, explanations and benchmark results are available on the website http://www.bioinformatics.nl/netclust/. There is also a web-based application suitable to run smaller test-sets.

Table 1. Protein classification results obtained for the individual and combined similarity networks.

Data set	Correct	Incorrect	Singletons
SW × DALI^{1} (251)	910	0	447
BLAST (0.1) × DALI^{2} (0.4)	888	0	469
BLAST (0.4) + DALI^{2} (0.4)	803	469	85
SW (251)	316	0	1041
DALI^{1} (251)	56	1266	35
DALI^{2} (0.4)	790	475	92
BLAST (0.4)	36	0	1321
BLAST (0.1)	66	1101	190

Numbers in parentheses denote the threshold used. Symbols '×' and '+' refer to the product and sum aggregation rules, respectively. DALI^{1} = matrix of raw scores, DALI^{2} = matrix of diagonally normalized scores. Correct = proteins connected only to members of the same SCOP superfamily, Incorrect = proteins connected to members of other SCOP superfamilies. The results were obtained for “alpha” weighting factor 0.5.

3 PERFORMANCE

The run-time of Multi-netclust subsumes (i) the time needed for reading-in the data, thresholding and aggregation (>99.9%), and (ii) finding the connected components and writing the results (<0.1%). A benchmark data set of 1357 proteins, taken from the Protein Classification Benchmark database (Sonego, et al., 2007) was used to combine sequence similarities calculated by the BLAST and Smith-Waterman (Smith and Waterman, 1981) algorithms, and DALI 3D structure similarities (Holm and Sander, 1995). The analysis took 4 seconds on a 2 GHz processor, the influence of parameter settings on the purity of connected clusters is apparent from the results (Table 1). An interesting example is the immunoglobulin superfamily (SCOP b.1.1) which has 125 members in the benchmark data set. Using DALI alone, the b.1.1 proteins clustered together with the “E set domains” (SCOP b.1.18), grouping proteins related to immunoglobulin and/or fibronectin type III superfamilies. Using BLAST alone, the b.1.1. proteins clustered together with a number of other superfamilies. Surprisingly, the combination of both DALI and BLAST data sets made 94% of the group b.1.1 cluster correctly.

The external memory-based, connected component search algorithm is fast and memory-efficient compared to single-linkage based clustering methods and in-memory graph algorithms used for similar purposes within the bioinformatics community (see supplementary material on the website). The strength of Multi-netclust becomes more apparent when we deal with large data sets that can not be handled with other algorithms. For example, a network of 2,713,908 nodes and 781,328,458 edges took less than 5 minutes on an ordinary computer. Of the other algorithms tested (see case studies on the website), only BLASTClust was able to handle a data set of similar size, however its use is limited to BLAST similarity networks (and at greater expense of CPU time and memory required), whereas Multi-netclust is generally applicable. To conclude, Multi-netclust is an efficient tool that can aid exploratory analyzes of large biological networks using an ordinary computer. Specifically, the potential applications include any task where network data of heterogeneous sources, such as sequence and structure similarities, gene expression or protein-protein interaction data, are to be combined together, resulting in new and/or improved biologically relevant predictions.

ACKNOWLEDGEMENTS

The authors would like to thank Pieter van Beek (SARA Computing and Networking Services) for help in providing additional BLAST data, Anand Gavai for his assistance in R programming, Stijn van Dongen for the help with the MCL utilities, and the NBIC initiative for using the Dutch Life Science Grid platform.

Funding: Graduate School Experimental Plant Sciences (to A.K.); BioAssist programme of the Netherlands Bioinformatics Centre (to H.N.).

REFERENCES

Altschul SF, et al. 1990. Basic local alignment search tool. *J. Mol. Biol.* **215**:403-410.

Chiang YJ, et al. 1995. External-Memory Graph Algorithms. In: *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95). Society for Industrial and Applied Mathematics.* 139-149.

Enright AJ, et al. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30**:1575-1584.

Holm L and Sander C. 1995. Dali: a network tool for protein structure comparison. *Trends Biochem Sci.* **20**:478-480.

Kittler J, et al. On combining classifiers. 1998. *IEEE Trans Pattern Anal Mach Intell.* **20**: 226-239.

Li W and Godzik A. 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**:1658-1659.

Sonego P, et al. 2007. A Protein Classification Benchmark collection for machine learning, *Nucleic Acids Res.* **35**:D232-236.

Smith TF and Waterman MS. 1981. Identification of common molecular subsequences. *J Mol Biol.* **141**:195-197.

Tarjan RE. 1975. Efficiency of a Good But Not Linear Set Union Algorithm. *Journal of the ACM.* **22**:215 – 225.

BLASTClust ftp://ftp.ncbi.nih.gov/blast/documents/blastclust.html.