Multi-netclust use cases

Example 1

Purpose:
To show the difference between two matrix aggregation methods, namely the sum rule and product rule, when combining two different data networks represented by sparse matrices.

Input:
Here we use two (artificial) networks represented by the `Grey.matrix` and `Red.matrix` matrix files. Both networks have the same number of nodes but differ in the number of edges as well as in the values assigned to the edges (edge weights). In Figure 1 the networks are superimposed graphically into a single, multi-parametric network with grey and red edges, which correspond to the different data sets. The bold edges indicate "strong" similarities (weight=1.0) whereas the dashed edges indicate "weak" or negligible similarities (weight=0.01).

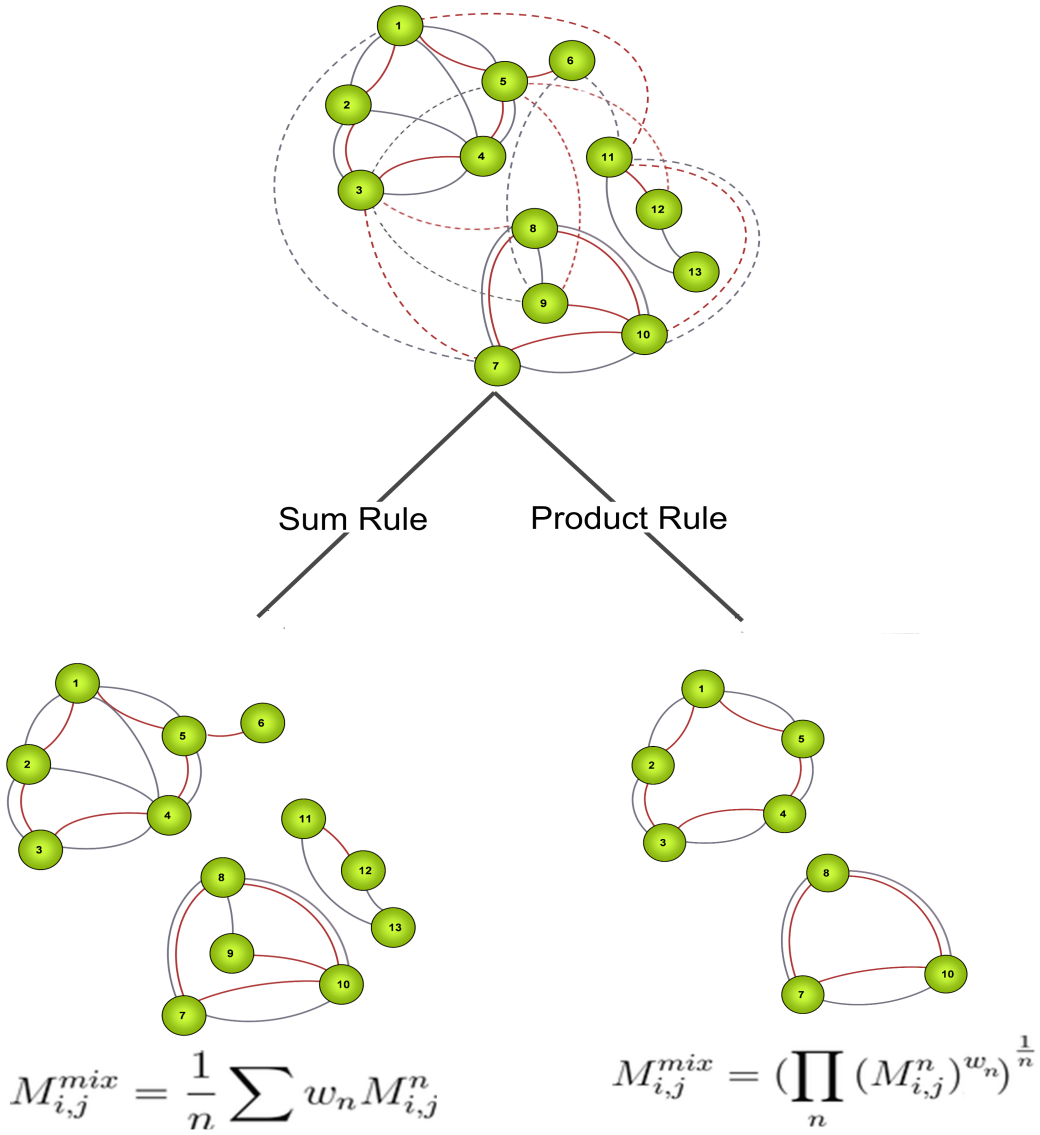

Figure 1

Data to the Multi-netclust can be entered either via the web interface or from the command-line. The web interface consists of two input sections where the user can adjust (a) the preprocessing and (b) cluster detection settings (Figure 2). The only mandatory field(s) in the form is the 'Matrix file'. Make sure that the input files are in the following format:

```
[nodeA]  [nodeB]  [weightAB]
A        B        1
B        C        0.01
```

(Note: the columns must be separated by a space ' ' or tab '\t')

To obtain the clustering results shown in Figure 1, the Multi-netclust web server (or standalone program) was executed with either sum or product rule, and with the clustering cutoff set to 0.1 so that “weak” similarities below this value were ignored during the cluster detection phase.

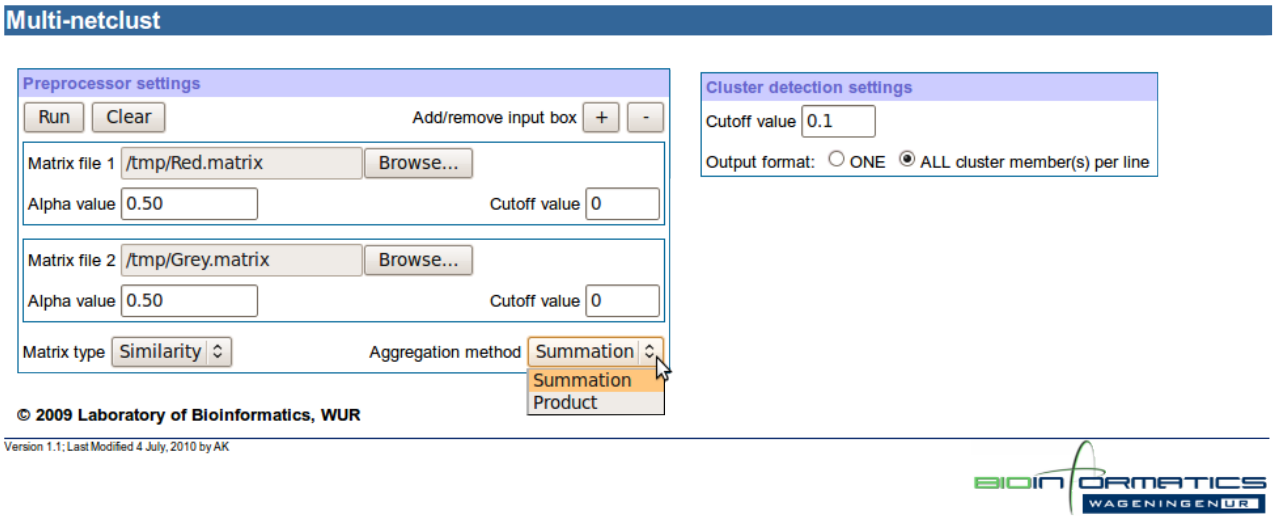

Figure 2

Here are the command-line equivalents for the standalone Multi-netclust program:

a) Sum rule

```
multi-netclust -n 2 -m 0 -C 0.1 -o Sum.matrix -f 3 Grey.matrix
Red.matrix
```

b) Product rule

```
multi-netclust -n 2 -m 1 -C 0.1 -o Product.matrix -f 3
Grey.matrix Red.matrix
```

Output:
The output of the Multi-netclust program are a list of the connected clusters (written into a *.clst file or standard output), combined matrix file, and the log files obtained after preprocessing, indexing and clustering. Figures 3a and 3b show the result pages obtained by running the Multi-netclust web server with the sum rule and product rule, respectively. The

resulting cluster and combined matrix files are available for download from the page. In addition, the clustering results can be also viewed in the page itself.

Multi-netclust

Prepare command: multi-netclust -n 2 -a 0.50,0.50 -c 0,0 -o result_matrix.txt -m 0 -f 3 -w 0 -l 0.1 -C 0.1 matrix1 matrix2

Execute command ... Done.

Download files: combined matrix, clusters

```
**********************
* Multi-netclust Log *
**********************

   InputFiles           Alpha                Cutoff
1) matrix1              0.500000             0.000000
2) matrix2              0.500000             0.000000

   OutputFile           result_matrix.txt
   WeightType           Similarity
   AggregationMethod    Summation
   NetIndexCutoff       0.100000
   NetClustCutoff       0.100000

***************
* Netclust Log *
***************
 InputFile              result_matrix.txt
 InputNidxFile          result_matrix.txt.nidx
 InputEidxFile          result_matrix.txt.eidx
 OutputClstFile         result_matrix.txt.clst
 OutputLogFile          result_matrix.txt.netclust.log
 WeightType             Similarity
 WeightCutoff           0.100000
 NumNodes               13
 NumEdges               16
 NumEdgesPassed         16
 NumClusters            3
```

Cluster ID (#)	Members
1 (6)	1, 2, 5, 4, 3, 6
2 (4)	7, 10, 8, 9
3 (3)	11, 13, 12

Figure3a

Multi-netclust

Prepare command: multi-netclust -n 2 -a 0.50,0.50 -c 0,0 -o result_matrix.txt -m 1 -f 3 -w 0 -l 0.1 -C 0.1 matrix1 matrix2

Execute command ... Done.

Download files: combined matrix, clusters

```
**********************
* Multi-netclust Log *
**********************

   InputFiles           Alpha                Cutoff
1) matrix1              0.500000             0.000000
2) matrix2              0.500000             0.000000

   OutputFile           result_matrix.txt
   WeightType           Similarity
   AggregationMethod    Product
   NetIndexCutoff       0.100000
   NetClustCutoff       0.100000

***************
* Netclust Log *
***************
 InputFile              result_matrix.txt
 InputNidxFile          result_matrix.txt.nidx
 InputEidxFile          result_matrix.txt.eidx
 OutputClstFile         result_matrix.txt.clst
 OutputLogFile          result_matrix.txt.netclust.log
 WeightType             Similarity
 WeightCutoff           0.100000
 NumNodes               8
 NumEdges               8
 NumEdgesPassed         8
 NumClusters            2
```

Cluster ID (#)	Members
1 (5)	1, 2, 5, 3, 4
2 (3)	7, 10, 8

Figure 3b

The sum rule corresponds to the union operation while the product rule is equivalent to the intersection operation involving the different data networks. As a result, the sum rule yields clusters connected in either of the networks whereas the product rule yields (strongly) clusters connected in all of the networks.

Example 2

Purpose:

Delineate distantly related proteins into SCOP superfamilies based on pairwise sequence and/or structure similarities.

Input:

This experiment involves a validation set taken from the Protein Classification Benchmark database (Sonego *et al.*, 2007). The set consists of 1357 protein domains classified into 24 superfamilies (see the `SCOP40mini.class` file), commonly used to evaluate machine learning algorithms. The input matrices for the Multi-netclust are provided in the `SW.matrix` and `DALI.matrix` files: the former matrix stores all-*versus*-all protein sequence similarities calculated by the Smith-Waterman algorithm (Smith and Waterman, 1981) with the BLOSUM62 matrix, as implemented in the SSEARCH program (version 3.4t16), and the latter matrix stores all-*versus*-all protein structure similarities (raw scores) calculated by the DALI-lite program, version 2.4.2 (Holm and Park, 2000).

We applied different clustering cutoffs (C=0, 251 and 448) and aggregation rules on the similarity matrices to compare the clustering results. The following commands were issued:

a) No aggregation

```
multi-netclust -n 1 -C 251 -f 2 SW.matrix

multi-netclust -n 1 -C 448 -f 2 SW.matrix

multi-netclust -n 1 -C 251 -f 2 DALI.matrix

multi-netclust -n 1 -C 448 -f 2 DALI.matrix
```

a) Sum rule

```
multi-netclust -n 2 -m 0 -C 251 -o SW+DALI.matrix -f 2
SW.matrix DALI.matrix

multi-netclust -n 2 -m 0 -C 448 -o SW+DALI.matrix -f 2
SW.matrix DALI.matrix
```

b) Product rule

```
multi-netclust -n 2 -m 1 -C 251 -o SWxDALI.matrix -f 2
SW.matrix DALI.matrix

multi-netclust -n 2 -m 1 -C 448 -o SWxDALI.matrix -f 2
SW.matrix DALI.matrix
```

Output:

The output of the Multi-netclust program are the cluster files (*.clst suffix) in the "one cluster member per line" format, combined matrix files (`SW+DALI.matrix` and `SWxDALI.matrix`) and the log files (*.log suffix).

Results:

Table 1.

Dataset	Correct	Incorrect	Singletons
SW (251)	316	0	1041
DALI (251)	56	1266	35
SW (448)	74	0	1283
DALI (448)	336	782	239
SW + DALI (251)	325	812	220
SW × DALI (251)	910	0	447
SW + DALI (448)	843	0	514
SW × DALI (448)	545	0	812

Numbers in parentheses denote the similarity cutoffs used. '×' and '+' refer to the product and sum aggregation rules, respectively. 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.

As an example, the superfamily of NAD(P)-binding Rossmann-fold domains (SCOP c.2.1) of 149 protein members, using the SW data alone as an input to the Multi-netclust, groups 30% of the superfamily correctly. With the DALI data, the superfamily is clustered with several other superfamilies, whereas the combination of the two (SW × DALI) and (SW + DALI) clustered

correctly 94% and 97% of the superfamily, respectively. Overall, combining the SW and DALI similarity data improves the grouping of the proteins into same SCOP superfamilies.

Note: The statistics in Table 1 were generated by the `EvaluateClusters.pl` Perl script.

Example 3

Purpose:
Delineate distantly related proteins into SCOP superfamilies based on pairwise sequence and/or structure similarities.

Input:
This experiment involves a validation set taken from the Protein Classification Benchmark database (Sonego *et al.*, 2007). The set consists of 1357 protein domains classified into 24 superfamilies (see the `SCOP40mini.class` file), commonly used to evaluate machine learning algorithms. The input matrices for the Multi-netclust are given in the `BLAST.matrix` and `DALI.matrix` files: the former matrix stores all-*versus*-all protein sequence similarities as calculated by the BLAST algorithm (Altschul *et al.*, 1990), version 2.2.13, BLOSUM62 matrix was used with default parameters, and the latter matrix stores all-*versus*-all protein structure similarities as calculated by the DALI-lite program, version 2.4.2 (Holm and Park, 2000). The matrices were also normalized by dividing each row element by the diagonal value.

We applied different clustering cutoffs (0.1 and 0.4) and aggregation rules on the similarity matrices to compare the clustering results. The following commands were issued:

a) No aggregation

```
multi-netclust -n 1 -C 0.1 -f 2 BLAST.matrix

multi-netclust -n 1 -C 0.4 -f 2 BLAST.matrix

multi-netclust -n 1 -C 0.1 -f 2 DALI.matrix

multi-netclust -n 1 -C 0.4 -f 2 DALI.matrix
```

a) Sum rule

```
multi-netclust -n 2 -m 0 -c 0.1 0.1 -o BLAST+DALI.matrix -f 2
BLAST.matrix DALI.matrix

multi-netclust -n 2 -m 0 -c 0.4 0.4 -o BLAST+DALI.matrix -f 2
BLAST.matrix DALI.matrix

multi-netclust -n 2 -m 0 -c 0.1 0.4 -o BLAST+DALI.matrix -f 2
BLAST.matrix DALI.matrix
```

b) Product rule

```
multi-netclust -n 2 -m 1 -c 0.1 0.1 -o BLASTxDALI.matrix -f 2
BLAST.matrix DALI.matrix
```

```
multi-netclust -n 2 -m 1 -c 0.4 0.4 -o BLASTxDALI.matrix -f 2
BLAST.matrix DALI.matrix

multi-netclust -n 2 -m 1 -c 0.1 0.4 -o BLASTxDALI.matrix -f 2
BLAST.matrix DALI.matrix
```

Output:

The output of the Multi-netclust program are the cluster files (*.clst suffix) in the "one cluster member per line" format, combined matrix files (`BLAST+DALI.matrix` and `BLASTxDALI.matrix`) and the log files (*.log suffix).

Results:

Table 2.

Dataset	Correct	Incorrect	Singletons
BLAST (0.1)	66	1101	190
DALI (0.1)	0	1352	5
BLAST (0.4)	36	0	1321
DALI (0.4)	798	468	91
BLAST (0.1) + DALI (0.1)	0	1357	0
BLAST (0.1) × DALI (0.1)	525	420	412
BLAST (0.4) + DALI (0.4)	803	469	85
BLAST (0.4) × DALI (0.4)	20	0	1337
BLAST (0.1) + DALI (0.4)	10	1317	30
BLAST (0.1) × DALI (0.4)	888	0	469

Numbers in parentheses denote the similarity cutoffs used. '×' and '+' refer to the product and sum aggregation rules, respectively. 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.

Overall, combining the BLAST and DALI similarity data improves the grouping of the proteins into same SCOP superfamilies.

Note: The statistics in Table 2 were generated by the `EvaluateClusters.pl` Perl script.