Using the NeAT toolbox to compare networks to networks, clusters to clusters and network to clusters

Sylvain Brohée[1,2]

[1] ESAT-SCD-SISTA (Bioinformatics group). Katholieke Universiteit Leuven. Kasteelpark Arenberg 10, B-3001 Leuven. Belgium. email: sbrohee@esat.kuleuven.be

[2] Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe). Université Libre de Bruxelles (ULB), CP 263, Campus La Plaine. Boulevard du Triomphe, 1050 Bruxelles, Belgium.

Email:

*Corresponding author

Abstract

In this chapter, we present some possible applications that can be performed with NeAT, a set of web tools aimed at studying biological networks (or graphs) and classifications.

These approaches are of particular interest for biologists and scientists who

need to assess the reliability of new data sets (either experimental or predicted) by comparing them to established references.

Firstly, we describe the steps that will allow a non specialist user to compare two networks in order to compute their union and the statistical significance of their intersection. In the second part, we will show how to map functionnal classes (e.g. GO categories, sets of regulons or complexes) onto a biological network. The third protocol explains how to compare two sets of functionnal classes, e.g., to assess statistatically the biological relevance of some computationaly returned groups of genes (clustering).

The metrics as well as the results obtained by following the different protocols are extensively described and explained.

NeAT is available at the following URL : `http://rsat.ulb.ac.be/neat/`.

keywords

network biology, network comparison, clustering assessment, network visualization.

1 Introduction

Throughout this book, we have been confronted with plenty of networks types implying bacterial proteins, genes or processes that illustrate the large variety of existing types of networks. In this chapter, we will describe methods to compare networks to other networks and to classifications.

In the following, the term network (or graph) will be considered in its simpler

meaning, i.e., a set of nodes (elements) which are connected by edges. Graphs can be directed (an edge going from node a to node b is not equivalent to an edge from node b to node a). In a directed graph, nodes presenting outgoing edges are called source nodes while nodes with incoming edges are called target nodes. Note that in a non-directed graph, the source and target node denomination often remains even if they may be swapped.

A classification is a set of elements organized in a set of classes (or clusters) that may overlap (some elements may belong to more than one class).

The methods described in the following will be applied to three different study cases focused on bacterial study cases. However, these methods are applicable to any kind of network and of classification (biological or not).

1.1 Study case 1: Network - network comparison. Comparing two infered regulatory networks

Network comparison presents many application domains. For example, the statistical merits of predictive methods can be calculated by comparing an infered network to a well established data set while taking the intersection (i.e. the common edges) of two or more networks increase their reliability as remaining common edges were discovered several times by different methods.

In this first study case, we will mainly focus on two networks obtained from the STRING database of interactions. This database contains many types of weighted links between genes. These interactions can be functional (e.g. co-transcription, phylogenetic co-occurence) as well as physical (experimentally

described protein-protein interaction). A score is attributed to each interaction, reflecting its reliability [1].

In this first section, we will compare two regulatory networks in *Escherichia coli* and visualize their union using the popular *Cytoscape* network tool. The first graph is a co-expression network built using the experimental data of a large number of microarray experiments. The second one is a network infered from neighborhood data (sets of genes that found repeatedly in close neighborhood within different genomes) and that may belong to the same operon (co-transcriptional network). This will allow us to consider the level of overlap between these two regulatory network inference approaches as well as its statistical significance. In a second step, we will randomize one of the two networks and apply the same graph comparison procedure to a randomized network in order to calculate the random expectation.

1.2 Study case 2: Network - clusters comparison. Mapping and extracting clusters from a co-expression network

Many classifications or pre-defined clusters are available to biologists : functional classes (Gene Ontology [2]), list of complexes or of regulons (sets of genes regulated by the same transcription factor). Moreover, various computational methods produce clusters (microarray or network clustering, supervised classification methods, etc). It is of interest to visualize whether groups of nodes belonging to the same class are grouped together in the network.

To this end, in the second example, we are going to extract and map

groups of annotated co-regulated genes (regulons) obtained from the RegulonDB database [3] onto the microarray co-transcription network described in the previous section. We will then be able to visualize the annotated regulons together with the predicted co-transcription edges. Indeed, for each edge of the network of co-expressed genes, we will determine whether it is intra-cluster or extra-cluster, i.e., whether it connects two nodes belonging to the same class (regulated by the same factor) or not. We distinguish the terms *mapping* (annotation of each edge as being intra or an extra-cluster) and *extraction* (removing all intra-cluster edges).

1.3 Study case 3: Clusters - clusters comparison. Comparing annotated to predicted regulons

As said in the previous section, besides the various annotated classifications, there exist many algorithms that produce clusters or classes according to varying criteria. A very classical example are the groups of co-expressed genes computed by comparing expression profiles. Another common example coming from network biology consists in constituting clusters from groups of highly interconnected subparts of the graph. Indeed, for example, in a protein protein interaction network, these highly connected parts of the network might correspond to protein complexes (e.g. see [4–7] and the chapters of this book dedicated to network clustering).

Having two classifications, each consisting in a set of classes, we will compare each class of the first classification to each class of the second one and compute

the statistical significance of their intersection size (the probability of reaching an intersection of at least that size to be obtained by chance only). This kind of approach is widespread in bioinformatics (e.g. see [8–10]). Indeed, in a large collection of clusters, it allows the easy retrieval of the clusters constituted of a significant number of elements sharing at least one characteristics, the elements of these clusters, not annotated as sharing the common characteristics can then be new potential investigation targets. For example, comparing set of highly connected proteins of a protein interaction network to a set annotated complexes will help us deciphering which clusters might be biologically relevant (those having a large intersection with the annotated complexes) and detecting new members of well established complexes.

In the following, we will compare an annotated set of regulons (obtained from the regulonDB database) to clusters constituted of the higly interconnected subparts of the STRING co-expression network described above (computed with the MCL network clustering algorithm (described in this book, [5]). It will be interesting to see whether the clusters returned by MCL do actually correspond to existing regulons.

2 Material

2.1 Software

1. Neat

   The NeAT suite (Network Analysis Tools) is a set of tools allowing users to perform statistical operations on networks (network compaison, node

statistics computation, network randomization, etc) and clusters (mapping of clusters onto a network, classification comparison, etc). In the present chapter, we will only describe the web server (`http://rsat.ulb.ac.be/neat/`) but the tools are also accessible using SOAP web services as well as simple command line tools [11].

The NeAT home page displays two frames. The left frame presents a menu of the available tools, the right frame displays the forms and result pages. Tools are modular and the output of one tool can easily be used as the input of another tool.

To solve the study cases described above, we will mainly use the following programs :

- *compare-graphs* which performs comparaison between two networks by computing the edges that are common to both, or found in only one of the two networks.
- *graph-get-clusters* which performs a mapping of a classification on a network by computing the intra-cluster and extra-cluster edges.
- *random-graph* which randomizes a network and that will be used to compute random controls.
- *compare-classes* which takes as input two classifications and computes the statistical significance of the intersection of each pair of classes.

2. *Cytoscape*

NeAT is not focused on the problem of network visualization. This is why we will use *Cytoscape* for a proper visualization of the network.

Cytoscape is the most known platform dedicated to the study of networks in bioinformatics. In its basic version, it is mainly devoted to visualization, as it includes a set of network layout algorithms. Among those, the spring embedding layout is very common for network visualization. Its principle is quite intuitive : linked nodes are attracted (edges are seen as springs) while two non connected nodes repell each other (like similarly charged particles). Starting from a random solution, at each iteration, the algorithm calculates the forces acting on each node and moves it according to them. After many iterations, an equilibrium state is reached where the force applied on each node is close to zero.

Additional plugins can be added in order to bring new functionnalities (clustering, topological analysis, ...). This program must be installed and runs on all major platforms. It is downloadable from **http://www.cytoscape.org** [12].

2.2 Intersection statistics

In several sections of this protocol, we try to detect significant intersections between two classifications or between two graphs. In both cases, we can consider that we have a finite set of N elements. For *compare-classes*, N is the total number of elements that can be member of any reference or query class (e.g. all the genes of an organism). For *compare-graphs*, the N elements are all the

possible edges between any pair of nodes (of both graphs).

Let us then define N, the total number of elements, R the reference set containing N_r elements, Q a query set containing N_q elements, C the intersection between a query and the reference set and N_c the number of elements in this intersection.

1. Calculating the maximum number of edges in a graph

   Knowing if it is directed and may contain self-loops (edges where the source node is also the target node), we can easily calculate the maximum number of edges in a network of n nodes with no duplicated edges.

Directed	Self-loops	N
Yes	Yes	x^2
Yes	No	$(x^2 - x) = x(x-1)$
No	Yes	$(x + x^2)/2 = x(x+1)/2$
No	No	$(x^2 - x)/2 = x(x-1)/2$

2. Jaccard coefficient

   The Jaccard coefficient (index) simply consists in the quotient of the size of the intersection between the classes by the size of the union. Although this index seems very intuitive, its main disadvantage is that is does not reflect the sizes of the compared sets. Indeed, an intersection of 1 between two sets of 2 and 1 element can have the same Jaccard index as an intersection of 100 between two sets of 100 and 300 elements, even if the latter is intuitively more significant. For this reason, we will prefer the

hypergeometric significance described below.

$J = \frac{R \cap Q}{R \cup Q}$

3. Hypergeometric significance

The hypergeometric distribution is generally used to estimate the statistical significance of the intersection between two random selections in a set. Its classical example of application is the random selection without replacement in an urn containing a set of white and black balls. The reference set (classes or graph) corresponds to the black balls of the urn example. The query set corresponds to the selection without replacement (indeed, a member cannot appear several times in the same class, and an edge cannot appear several times in the same graph). The hypergeometric P-value (p-val) indicates the probability to observe by chance at least x elements at the intersection between the query set and the reference set.

$p - val = P(X \geq x) = \sum_{i=N_c}^{N_q} \frac{C_{N_r}^{i} C_{N-N_r}^{N_q - i}}{C_N^{N_q}}$

When the number of comparison is large, a correction for multi-testing must be introduced. Indeed, when comparing many classes together (like in the third study case) the number of comparison is the number of reference classes multiplied by the number of query classes, a low P-value is expected to be found by chance. A classical way to apply this correction is to compute the E-value (e-val) as following.

$e - val = P - val \cdot T$

The E-value corresponds to the expected number of false positives at a given P-value for a battery of T tests. To get a more intuitive view of the E-value, the hypergeometric significance is then commonly calculated as following.

$sig_H = -log_{10}(e - val)$

The sig score gives an intuitive perception of the exceptionality of the intersection between sets: a negative significance indicates that an intersection of at least that size is likely to occur by chance, a positive value means that it is significant.

2.3 File formats

1. Networks can be saved in a variety of formats. NeAT and Cytoscape support some of them, which are described hereafter. The NeAT program *convert-graph* facilitates the handling of these formats, by supporting interconversions between some input and output formats.

   The *tab-delimited* format is a convenient and intuitive way to encode a graph where each row represents an edge and each column contains its attributes. Generally, the two first columns contain the nodes linked by an edge. Logically, in undirected graph, the columns containing the source and the target nodes may be swapped. Orphan nodes (nodes without neighbour) can be included by specifying a source node without target. Some additional edge attributes (weight, label, color) can be placed in additional columns.

The *GML* format (`http://www.infosun.fim.uni-passau.de/Graphlet/GML/`) allows one to specify layout characteristics of the network (node size, color, label). GML is made up of nested key-value pairs. The most popular graph editors (like Cytoscape and yEd) support GML as input format.

2. Classification Classifications used by NeAT are files composed of two columns delimited by tabulation. The first column indicates the name of an element while the second column specify the cluster this element belongs to.

2.4 Sample files

The files that will be used for our study cases can be downloaded from the sample file repository on `http://rsat.ulb.ac.be/rsat/data/published_data/MiMB2009/`.

`string_ecoli_coexpression.tab` and `string_ecoli_neighborhood.tab` contain the STRING network infered from microarray data and from gene neighborhood respectively. These networks are encoded in the tab delimited format. The two first columns contain the node names and the third column the weight of the edge linking both nodes.

`regulonDB.tab` contains a list of regulons extracted from the RegulonDB database. This is a NeAT classification file as the first column contain the name of the regulated gene and the second column their regulator.

`string_ecoli_coex_mcl.tab` is the tab-delimited file obtained when applying the MCL graph clustering algorithm on the `string_ecoli_coexpression.tab` described above. The first column contains the node names and the second column the clusters they belong to.

2.5 Supplementary material

Supplementary figures and data are stored on the following webpage `http://rsat.ulb.ac.be/rsat/data/published_data/MiMB2009/neat_supmater.html`

3 Methods

3.1 Network comparisons and visualization

In this subsection we will show how to calculate the union of two networks. This procedure takes as input two networks (called the query and the reference networks), and returns a larger colored network corresponding to their union. Each edge is colored according to the fact it belongs only to the query graph (red), to reference graph (violet) or to both (green).

1. Download the sample networks from the data repository described in section 2.4. To this end, with your web browser go to the repository website (`http://rsat.ulb.ac.be/rsat/data/published_data/MiMB2009/`) and right click on the file `string_ecoli_coexpression.tab`, choose *Save link as* and save it on your computer under the same name. Do the same for the classification file `string_ecoli_neighborhood.tab`.

2. Open a connection to the NeAT web server: `http://rsat.ulb.ac.be/neat/`.

3. Click on 'Network comparison' in the left frame.

4. Select the reference and the query networks that you want to compare by clicking on the *Browse* button. Select the previously downloaded file `string_ecoli_neighborhood.tab` as query and `string_ecoli_coexpression.tab` as reference.

5. The source and target node columns are correctly attributed by default (1 and 2). However, as these graphs are weighted, specify for both networks that 3 is the column containing the weight.

6. Select *GML* as output format (so that we can open the resulting file with Cytoscape).

7. Check that union is selected as *Output type.* In this menu, you could have chosen *Intersection*, *Difference* (which would remove all the common edges from the query graph), *Query graph only* or *Reference graph only*,

8. There are different types of weights for the resulting union graph. By default, on common edges, the weight of both edges is returned. You can also select the sum, the mean, the minimal and maximal values of both edges or only the weights of the query or the reference.

9. Finally, you can indicate whether or not the graph may admit self-loops (edges having the same node as source and target) or whether it is di-

rected. In our study case, the graph is undirected and has no self-loop. Indeed, networks that may contain loops or which are directed contain a larger number of edges. The maximum number of edges is used for the computation of the intersection size significance.

10. Click on the *GO* button. The operation might take up to one minute depending on how many people are connected and working on the NeAT server.

11. The result page displays some comparison statistics (that will be discussed in the Result section) as well as the resulting network (blue link to a GML file). Below the link to the file, there are some buttons displaying the names of some other graph tools and that allow us to use the union graph that we just computed as input for these tools.

12. To visualize the network with Cytoscape, right click on the link to the union graph and store it on your computer.

13. Open Cytoscape.

14. In the File menu, select Import and Network (Multiple file types) and select the GML file you just copied on your computer. The union graph appears on the screen, the nodes being placed on a oblique line.

15. In the *Layout* menu, select *yFiles* > *organic* for a more intuitive visualisation of the network. The network is now displayed in the classical spring embedding layout discussed above.

16. Back to the NeAt webpage, to build the negative control, in the NeAT left menu, click on the *Randomize network* link. The form of the *random-graph* tool appears.

17. Upload one of the two graphs we used in the graph comparison by clicking on the *Browse* button (e.g. `string_ecoli_neighborhood.tab`).

18. Choose the type of randomization. We recommand you to use the node degree conservation that consists in shuffling the edges between the nodes, each node keeping the same number of neighbors as in the original graph. This procedure is specially designed to avoid duplicating edges, unless you check the option "Allow duplicated edges" (which should not be done).

19. Click on the *GO* button. After some seconds of computation, a link to the results is displayed as well as buttons proposing to use the result file as input to other operations.

20. Click on the button *Compare this graph to another one*, as we want to compare the randomized neighborhood network to the non randomized coexpression network. The *compare-graphs* form described above is displayed with a notable difference. The query graph part of the form is pre-filled with the randomized network. To run the comparison, follow again steps 4 to 11 of this section.

3.2 Network - clusters comparison and visualization

In this section, we will describe how to compare a classification (set of classes) to a network.

1. Download the sample network and classification files from the data repository described in section 2.4. To this end, with your web browser, go to the repository website (`http://rsat.ulb.ac.be/rsat/data/published_data/MiMB2009/`) and right click on the file `string_ecoli_coexpression.tab`, choose *Save link as* and save it on your computer under the same name. Do the same for file `regulonDB.tab`.

2. Click on the *Map clusters onto network* link in the NeAT left menu *Map clusters onto network*. The form of *graph-get-clusters* is displayed on the screen

3. Select the co-expression graph that you want to compare by clicking on the *Browse* button next to the *Upload graph from file* field.

4. The source and target node columns are correctly attributed by default (1 and 2). However, as these graphs are weighted, specify that the third column contains the weight.

5. Select the clusters file, by clicking on the *Browse* buttons next to the *Upload clusters from file* field.

6. The main choice for the tool *graph-get-clusters* is the output type. The program supports different types of operations between the network and

the clusters. We will test the two major ones. (i) The option *annotated graph* labels each edge according to its intra or inter-cluster nature while (ii) the *intracluster edges* option selects a subnetwork restricted to the intracluster edges (intercluster edges are deleted from the network). Let us start by selecting the latter option 'intracluster edges'.

7. Select the GML output format to visualize the output network with Cytoscape.

8. Click on the *GO* button. The operation might take up to one minute depending on how many people are connected and working on the NeAT server.

9. As usual, the result page contains a blue hypertext link to the result network as well as some buttons proposing to use the result network as input for other operations.

10. To visualize the network, perform operations 12 to 15 of section 3.1.

11. To map the clusters, perform again steps 2 to 10 of this section. However, at step 6, select the option *annotated graph (all edges)*.

3.3 Clusters - clusters comparison

Here, we will show how to compare two classifications and find clusters presenting a statistically larger intersection.

1. Download the classification files from the data repository described in section 2.4. To this, with your web browser go to repository website

(`http://rsat.ulb.ac.be/rsat/data/published_data/MiMB2009/`) and right click on the file `regulonDB.tab`, then choose *Save link as* and save it on your computer under the same name. Do the same for file `coexpression_single_names_mcl_i_1_8.tab`.

2. Click on the *Compare classes/clusters* link in the NeAT left menu. The input form of the *compare-classes* tool appears.

3. Upload the classifications files by clicking on the *Browse* buttons. Query classification corresponds to clusters returned by the MCL algorithm (`coexpression_single_names_mcl_i_1_8.tab`) while the reference file is the file containing the regulons annotated in the RegulonDB database (`regulonDB.tab`).

4. As there is no score for these classification files, the *score column* field may remain empty.

5. The options *Thresholds on return* fields allows one to combine various constraints to select the most significant intersections between query and reference clusters/classes. The default threshold on significance ($sig\_Z > 0$) usually gives satisfactory results. For our study case, the other thresholds do not need to be changed.

6. Click on the *GO* button. The operation might take up to one minute depending on how many people are connected and working on the NeAT server.

7. Here, the result page contains two hypertext links to the results. The first link will redirect you to a tab delimited text file and the second one to

an HTML conversion of the first file. These files consist mainly of a table where each line reports the result of one comparison. The HTML format is useful for inspecting and handling the result on the web browser. The table can be sorted according to the data of any column simply by clicking on the header of these columns. The first lines of the result file indicates the parameters used for the analysis, and documents the content of the columns. This is followed by a results table, where each line reports the comparison between one MCL cluster and one complex (Figure 3).

3.4 Study case 1: Network - network comparison. Comparing two infered regulatory networks

The first study case consisted of computing the union of two infered regulatory networks downloaded from the STRING database. The results of the procedure can be seen on Figures 1 and S 4

Figure 1 shows the result page of the NeAT *compare-graphs* tool. On this page, some network characteristics are summarized (number of edges, nodes in the query or reference network, etc). The coexpression network used as reference contains 1,609 genes and 16,267 edges while the query graph (neighborhood) contained 3,699 nodes and 34,138 edges. The intersection between those graphs is apparently weak: 1,344 edges only. The Jaccard coefficient indicates that this intersection represents no more than 2.7% of the union. However, the number of edges expected by chance at the intersection is even smaller: $E(Q \cap R)$=68. The hypergeometric P-value indicates the probability to observe at least 1,344

edges at the intersection when 68 are expected by chance. In this study case, we observe that even with no more than 0.68% of edges at the intersection, the P-value is extremely low ($< 10^{-300}$). In other terms, the number of edges at the intersection is much too high to be explained by chance, and is more likely to result from the biological relevance of both datasets. In contrast, the number of edges in common between the randomized query and the reference only delivers about 100 edges which means a Jaccard index of $\sim 0.2\%$ and a much higher P-value ($2.3 \cdot 10^{-5}$, i.e, more than 10^{295} times lower). The results obtained with the random control may vary, this is why generally, random controls are computed multiple times to obtain an estimate of their score distribution.

Figure S 4 contains the union of both reference and query graph plotted in Cytoscape with the *Organic* layout. This figure and the network itself can be downloaded from the supplementary material webpage.

3.5 Study case 2: Network - clusters comparison. Mapping and extracting clusters from a co-expression network

The network resulting from the cluster extraction methodology described in section 3.2 consists in the STRING coexpression network where each inter-cluster edge, i.e., the edges connecting a source and a target node not belonging to the same annotated regulon has been removed (Supplementary figure S5). It is thus only composed of intra-cluster edges, the edges being colored according to the cluster they belong to. Some edges may belong to more than one cluster, in this case, the program duplicates the edges so that the color of each cluster

is represented.

Figure 2(a) shows an enlarged part of the intra-cluster edges network consisting of the genes implicated in sulfur metabolism which are all regulated by the cysB factor. This representation also shows that genes cysH, cysI and cysJ are also annotated as regulated by another factor (the global repressor FIH) as these three genes are linked by an edge presenting a different color (here, dashed and wider for vizualisation purposes).

Figure 2(b) shows a part of the network resulting from the cluster mapping where all the inter-cluster edges have been colored in grey and intra-cluster edges have been assigned a different color. For visualization purposes, the intra-cluster edges are wider and dashed. With this representation, the user can compare the annotated data (here, the different regulons from regulonDB) to the inferred network and directly correct them. Indeed, if a set of genes connected by edges of the same colors (belonging to the same regulon) are also linked to one or more genes of similar function (predicted as co-expressed), this might mean that the annotations are not complete. This is what we observe, indeed, the gene fadH is predicted as coexpressed with the genes of the fadR regulon (implied in fatty acid metabolism) but is not annotated as such in regulonDB.

Figure 2 as well as the different resulting networks can be seen in colors on the supplementary material webpage.

3.6 Study case 3: Clusters - clusters comparison. Comparing annotated to predicted regulons

Figure 3 shows the comparaison results between the dense clusters (obtained by applying the MCL clustering algorithm on a coexpression network) and the regulons of regulonDB, ranked by the hypergeometric significance. The header gives a short explanation for the content of each column of the results table. Each row describes the comparison between one MCL cluster and one regulon. For example, the first row compares the MCL cluster "cl_12", which contains 17 genes, with a set of 24 genes regulated by the CysB factor (implied in cysteine metabolism regulation). The intersection contains 12 genes, which represents a high fraction of both the MCL cluster, and the annotated complex. The probability to observe such an intersection by chance is $2.4 \cdot 10^{-22}$. The E-value, obtained with the correction for multi-testing, indicates that the number of false positives expected with such a *p-value* would be $9.1 \cdot 10^{-18}$. In other terms, the correspondence between this MCL cluster and the cysteine metabolism gene is too high to be explained by chance.

Acknowledgements

KULeuven SCD-SISTA lab is funded by the Research Council KUL (GOA MaNet, GOA AMBioRICS, CoE EF/05/007 SymBioSys, PROMETA, START 1, several PhD/postdoc and fellow grants, GOA 2006/12), FWO [PhD/postdoc grants, projects G.0241.04 (Functional Genomics), G.0499.04 (Statistics), G.0232.05 (Cardiovascular), G.0318.05 (subfunctionalization), G.0553.06 (VitamineD), G.0302.07

(SVM/Kernel), research communities (ICCoS, ANMMM, MLDM)], G.0733.09 3UTR; G. 082409 (EGFR), G.0254.05 (Genetics of human heart develoment), IWT (PhD Grants, GBOU-McKnow-E (Knowledge management algorithms), GBOU-ANA (biosensors), TAD-BioScope-IT, Silicos; SBO-BioFrame, SBO-MoKa, TBM Endometriosis), the Belgian Federal Science Policy Office [IUAP P6/25 (BioMaGNet, Bioinformatics and Modeling: from Genomes to Networks, 2007-2011), IUAP P5/25 (Molecular Pathology of Genetic Diseases) and the EU-RTD (ERNSI: European Research Network on System Identification; FP6-NoE Biopattern; FP6-IP e-Tumours, FP6-MC-EST Bioptrain, FP6-STREP Strokemap). SB was supported by a FRIA doctoral grant (Fonds pour la Recherche dans l'Industrie et l'Agriculture) during the development of NeAT, in the BiGRE lab and is now supported by a post-doc grant of the CheartED project (in the SISTA lab).

4 References

References

1. Jensen LJ, Kuhn M, Stark M, Chaffron S, Creevey C, Muller J, Doerks T, Julien P, Roth A, Simonovic M, Bork P, von Mering C: **STRING 8–a global view on proteins and their functional interactions in 630 organisms.** *Nucleic Acids Res* 2009, **37**(Database issue):D412–D416, [http://dx.doi.org/10.1093/nar/gkn760].

2. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nat Genet* 2000, **25**:25–29, [http://dx.doi.org/10.1038/75556].

3. Gama-Castro S, Jiménez-Jacinto V, Peralta-Gil M, Santos-Zavaleta A, Peñaloza-Spinola MI, Contreras-Moreira B, Segura-Salazar J, Muñiz-Rascado L, Martínez-Flores I, Salgado H, Bonavides-Martínez C, Abreu-Goodger C, Rodríguez-Penagos C, Miranda-Ríos J, Morett E, Merino E, Huerta AM, Treviño-Quintanilla L, Collado-Vides J: **RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation.** *Nucleic Acids Res* 2008, **36**(Database issue):D120–D124, [http://dx.doi.org/10.1093/nar/gkm994].

4. King AD, Przulj N, Jurisica I: **Protein complex prediction via cost-based clustering.** *Bioinformatics* 2004, **20**(17):3013–3020, [http://dx.doi.org/10.1093/bioinformatics/bth351].

5. Enright AJ, Dongen SV, Ouzounis CA: **An efficient algorithm for large-scale detection of protein families.** *Nucleic Acids Res* 2002, **30**(7):1575–1584.

6. Blatt, Wiseman, Domany: **Superparamagnetic clustering of data.** *Phys Rev Lett* 1996, **76**(18):3251–3254.

7. Bader GD, Hogue CWV: **An automated method for finding molecular complexes in large protein interaction networks.** *BMC Bioinformatics* 2003, 4:2.

8. Naamane N, van Helden J, Eizirik DL: **In silico identification of NF-kappaB-regulated genes in pancreatic beta-cells.** *BMC Bioinformatics* 2007, 8:55, [http://dx.doi.org/10.1186/1471-2105-8-55].

9. Dessailly BH, Lensink MF, Wodak SJ: **Relating destabilizing regions to known functional sites in proteins.** *BMC Bioinformatics* 2007, 8:141, [http://dx.doi.org/10.1186/1471-2105-8-141].

10. Barriot R, Sherman DJ, Dutour I: **How to decide which are the most pertinent overly-represented features during gene set enrichment analysis.** *BMC Bioinformatics* 2007, 8:332, [http://dx.doi.org/10.1186/1471-2105-8-332].

11. Brohée S, Faust K, Lima-Mendez G, Sand O, Janky R, Vanderstocken G, Deville Y, van Helden J: **NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways.** *Nucleic Acids Res* 2008, **36**(Web Server issue):W444–W451, [http://dx.doi.org/10.1093/nar/gkn336].

12. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res* 2003, **13**(11):2498–2504, [http://dx.doi.org/10.1101/gr.1239303].

Figure captions

Figure 1: Study case 1. Comparison between two infered regulatory networks downloaded from the STRING database (query : neighorhood and reference : microarray data). Result page of the NeAT tool *compare-graphs* detailing the network characteristics and the intersection statistics.

Figure 2: Study case 2. Extraction and mapping of co-regulated genes clusters (regulons) on a regulatory network. (a) Graphical representation of the cluster extraction : each inter-cluster edge is removed and edges are colored differently according to the cluster they belong to. (b) Enlargement of the CysB regulon (sulfur metabolism) showing that some edges may belong to more than one cluster. (c) Enlargment of the representation of the cluster mapping consisting in the fadR regulon (fatty acid metabolism). It shows that some genes are not annotated as regulated by fadR but are found to be co-expressed with fadR genes.

Figure 3: Study case 3. Most significant associations between MCL clusters versus regulonDB regulons. The figure only shows the top of the table returned by the program *compare-classes*. Each row represents the comparison between one regulon (reference) and one MCL cluster (query).

Figure 4: (Supplementary figure) Study case 1. Graphical representation (with Cytoscape) of the union of both reference and query networks.

Figure 5: (Supplementary figure) Study case 2. Graphical representation of the mapping of the regulons of the RegulonDB database onto the STRING coexpression network.

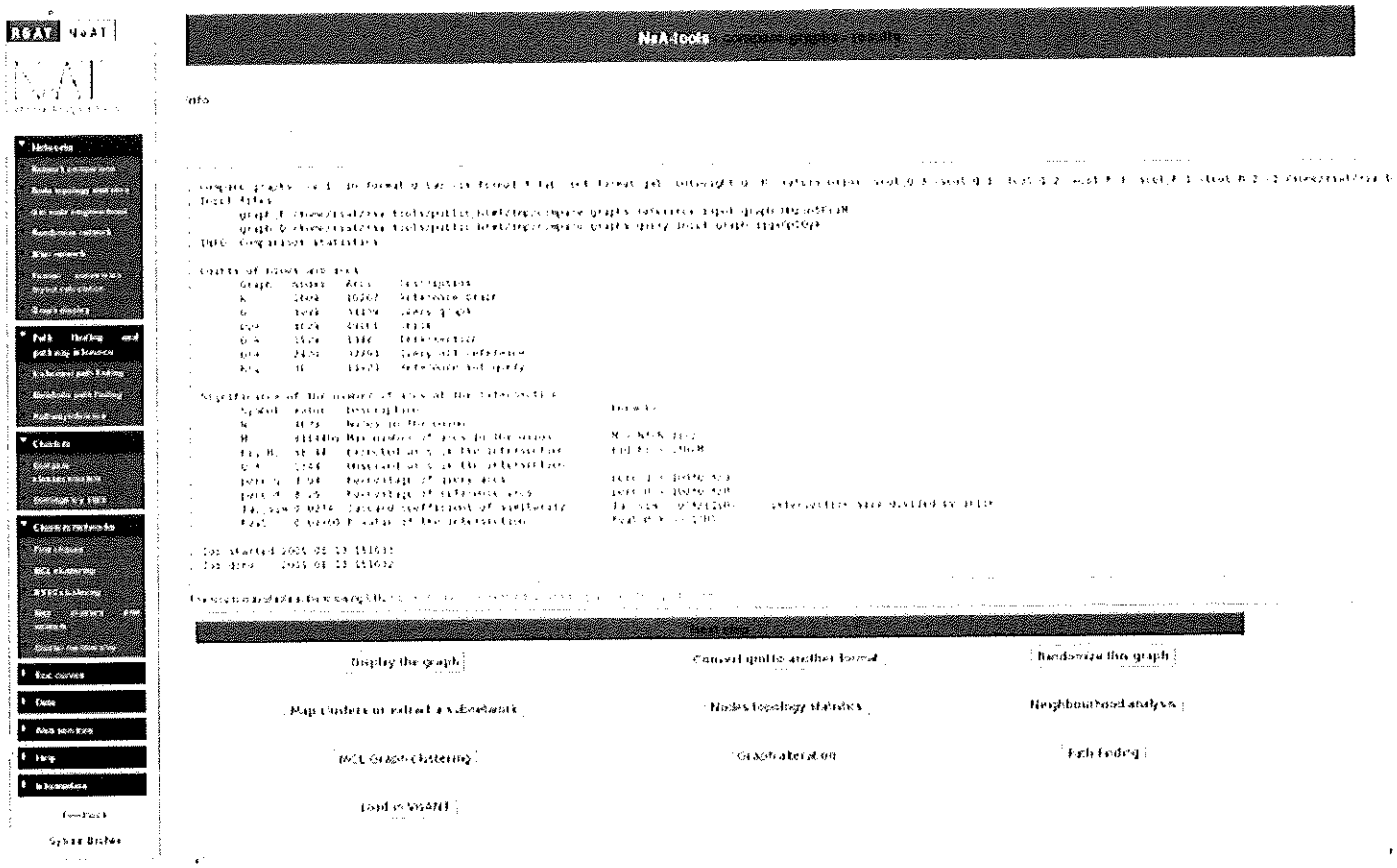

NeA-tools
Info
Display the graph
Convert graphs another format
Randomize this graph
Map clusters or extract a subnetwork
Nodes topology statistics
Neighbourhood analysis
MCL Graph clustering
Graph alteration
Path finding
Load in VISANT

Fig 1

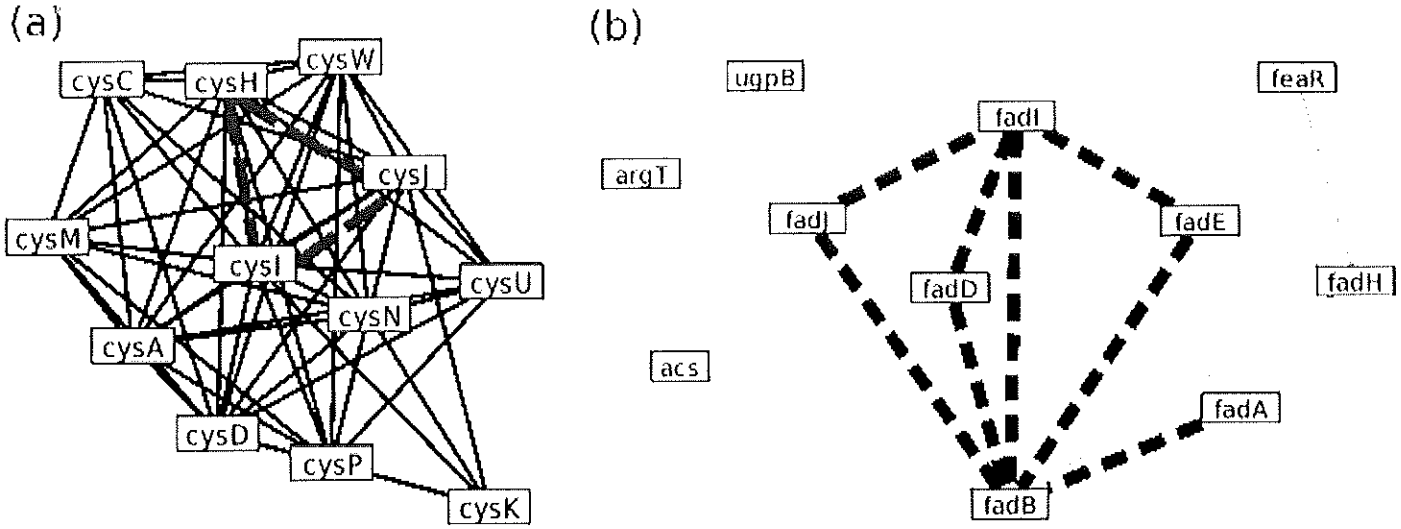
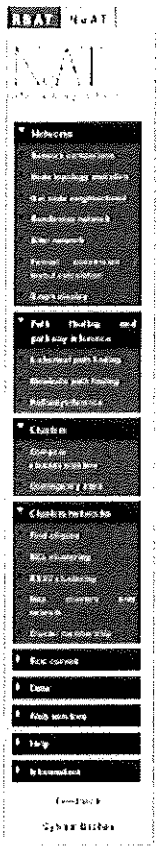

(a)
cysC
cysH
cysW
cysJ
cysM
cysI
cysU
cysN
cysA
cysD
cysP
cysK
(b)
ugpB
feaR
fadI
argT
fadJ
fadE
fadD
fadH
acs
fadA
fadB

Fig 2



rank	ref	query	B	Q	QB	QxB	RBQ	QBR	BQ2B	P_val	E_val	sig a
1	PyvB	cl	12	21	12	29	12	4	2.95E-2	2.1e-22	9.6e-18	1.04
2	HypB	cl	32	12	7	7	12	5	0	2.35E-1	9.3e-18	3.9e-13	12.14
3	AppY	cl	26	10	9	3	12	5	2	2.37E-1	9e-17	3.9e-12	11.22
4	AtpIC	cl	16	28	11	9	99	10	2	2.95e-2	9.9e-17	2.1e-12	11.67
5	PurR	cl	15	11	11	9	33	22	2	2.35E-1	1.6e-16	6.2e-12	11.21
6	HlyD	cl	9	15	21	14	92	21	7	2.29E-1	1.2e-15	6.9e-12	11.19
7	GalR	cl	25	6	0	0	0	0	3	2.37E-1	3.3e-15	1.1e-11	10.90
8	GalX	cl	13	17	13	8	22	9	5	2.36E-1	1.2e-14	4.6e-11	10.41
9	Fur	cl	27	28	9	9	28	69	0	2.36E-1	2.2e-14	3.6e-09	8.99
10	OxyR	cl	15	18	6	6	18	12	0	2.36E-1	7.4e-14	2.8e-09	8.55
11	FadR	cl	17	11	11	6	16	5	5	2.34E-1	8.3e-14	3.2e-08	7.50
12	BetI	cl	15	26	6	6	26	20	0	2.36E-1	9e-14	3.5e-08	7.46
13	HlyDC	cl	29	13	8	8	104	72	0	2.36E-1	1.9e-12	7.2e-08	7.13
14	GalR	cl	14	27	11	7	34	28	6	2.35E-1	1.2e-11	6.5e-07	6.19
15	IscR	cl	26	26	9	6	24	20	5	2.35E-1	4.4e-11	2.4e-06	5.61

Fig 3