2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing

Layout and Post-Processing of Transcriptional Modules

Hong Sun, Tim Van den Bulcke, Bart De Moor

Department of Electrical Engineering
Katholieke Universiteit Leuven
Leuven, Belgium

hong.sun@esat.kuleuven.be
tim.vandenbulcke@esat.kuleuven.be
bart.demoor@esat.kuleuven.be

Karen Lemmens, Kristof Engelen, Kathleen Marchal

Department of Microbial and Molecular Systems
Katholieke Universiteit Leuven
Leuven, Belgium

karen.lemmens@esat.kuleuven.be
kristof.engelen@biw.kuleuven.be
kathleen.marchal@biw.kuleuven.be

***Abstract*—Visualization of transcriptional modules together with their transcriptional program is a non-trivial task. We have therefore developed a module visualization tool that allows visualizing overlapping transcriptional modules in a very intuitive way. By visualizing not only the genes and the experiments in which the genes are co-expressed, but also additional properties of the modules such as the regulators and regulatory motifs that are responsible for the observed co-expression, our tool can assist in the biological analysis and interpretation of the output of module detection tools.**

Keywords - software; transcriptional modules; visualization

I. INTRODUCTION

Organisms are able to adapt their cellular machinery to changing environmental conditions. This complex cellular behavior is mediated by the underlying regulatory network. When the regulation takes place at the level of mRNA regulation, we talk about the transcriptional network. Previous studies have unveiled the modular and hierarchic organization of the transcriptional network [1-3]. Indeed, biological processes consist of pathways that mainly act on their own although communication exists between these pathways. Therefore one might expect that the distinct biological processes are organized in discrete and separable modules.

Biclustering tools form one type of transcriptional module detection tools. These algorithms make use of microarray compendia to reveal the modularity of the transcriptional network (for an overview, see [4]). A bicluster (or module) is defined as a group of genes that show a similar expression profile in a subset of experiments. Genes within a bicluster usually belong to the same pathway or have a related biological function. Other transcriptional module detection tools [3;5-9] go one step beyond. Not only do they search for the modules, but they also identify the regulatory program responsible for the observed co-expression behavior of the genes in the module.

Usually, many overlapping modules are identified by module detection tools. Indeed, genes can be involved in multiple pathways. In addition, multiple pathways can be triggered in one particular environmental cue. Having a visual overview of how these modules overlap, gives insight in the structure of the biological system. Biclustering software usually includes the possibility to visualize the retrieved modules one at the time, but rarely simultaneously. For instance, BiVisu [10], BicAt [11] and Expander [12] allow visualizing the genes and experiments of one module by means of an expression profile or a heat map.

The problem with visualizing overlapping modules simultaneously is that the overlap in multiple dimensions complicates the choice of an appropriate layout. Therefore few tools exist that are capable of visualizing modules simultaneously. In [13], for instance, a tool for the visualization of multiple, overlapping biclusters in a two-dimensional gene-experiment matrix was developed. As each bicluster is represented in this layout-matrix as a contiguous submatrix, genes and experiments that belong to multiple overlapping biclusters will be duplicated to obtain an optimal layout of the biclusters. This duplication of genes and experiments, however, complicates the biological interpretation of the biclusters. The recently developed tool BicOverlapper [14] displays overlapping biclusters by means of a graph-based representation. The nodes in the graph represent respectively experiments and genes of the data set. An edge between two nodes indicates that the connected nodes are part of the same bicluster. A bicluster is thus represented as an undirected, fully connected subgraph. The nodes are positioned in the display based on their bicluster assignment: nodes of the same bicluster will be placed close to each other while nodes belonging to different biclusters will be positioned at a larger distance. Nodes that are in common between multiple biclusters will be placed in between those biclusters.

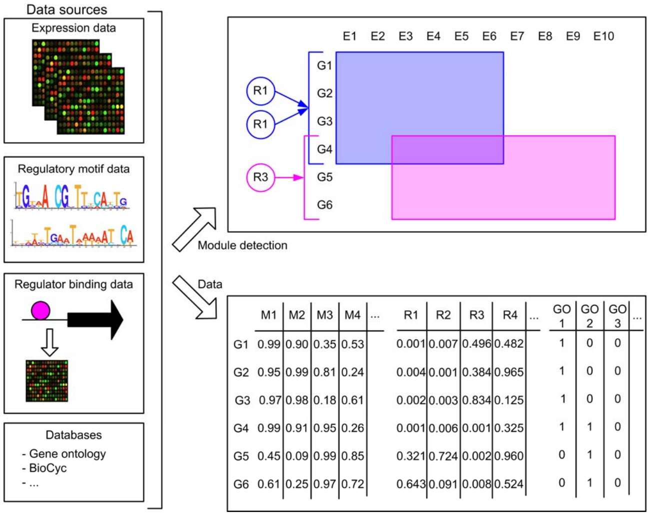
In this study we developed a tool that allows for a dynamic visualization of overlapping transcriptional modules in a 2D gene-experiment matrix. Multiple methods are included for obtaining the optimal layout of the overlapping modules. In addition to the previously developed tools for visualizing multiple modules, our tool also allows to display additional information on the regulatory program of the modules. The regulatory program consists of the transcription factors and their corresponding motifs. A first way of obtaining information on the

978-0-7695-3739-9/09 $25.00 © 2009 IEEE
DOI 10.1109/IJCBS.2009.95

regulatory program is by using the information from curated databases or other data sources. This information can be used to further analyze modules inferred by biclustering algorithms. Secondly, information on the regulatory program can also be the outcome of a module inference tool itself. Both types of information on the regulatory program can be included by our visualization tool (see Figure 1). By visualizing not only the modules, but also the regulatory program, our tool can provide more insight into the modules and makes the biological interpretation of the identified modules more accessible to biologists.

II. Methods

We have developed a java-based tool for visualizing multiple overlapping modules together with additional information on their regulatory program. The minimal information that is required to visualize modules are the genes and experiments composing the modules (see Figure 1). Additional properties of the genes, experiments and modules are optional. A gene property includes membership to a particular gene ontology class or the presence of a transcription factor binding site (motif) in or the binding of a transcription factor to its upstream region. An experiment property includes the membership of an experiment to a particular conditional class, which gives information on the major cue that was measured during the experiment. A module property is only available when the module inference tool was capable of identifying the regulatory program of the module. This module property consists of the list of transcription factors or transcription factor binding sites that were assigned by the inference algorithm to the module. Note that the latter information on the regulatory program is different from the gene properties which are not inferred, but derived from curated databases.



	M1	M2	M3	M4	...	R1	R2	R3	R4	...	GO 1	GO 2	GO 3	...
G1	0.99	0.90	0.35	0.53		0.001	0.007	0.496	0.482		1	0	0	
G2	0.95	0.99	0.81	0.24		0.004	0.001	0.384	0.965		1	0	0	
G3	0.97	0.98	0.18	0.61		0.002	0.003	0.834	0.125		1	0	0	
G4	0.99	0.91	0.95	0.26		0.001	0.006	0.001	0.325		1	1	0	
G5	0.45	0.09	0.99	0.85		0.321	0.724	0.002	0.960		0	1	0	
G6	0.61	0.25	0.97	0.72		0.643	0.091	0.008	0.524		0	1	0	

Figure 1. Example of input for our tool. Module detection tools can make use of several types of data, such as expression data, regulatory motif data, regulator binding data, databases, etc. for the identification of regulatory modules and their program. This example shows two modules (blue and purple) that consist of a set of genes that are co-expressed in a set of experiments. The blue module, for instance, consists of genes G1, G2, G3 and G4 and experiments E1, E2, E3, E4, E5 and E6. This is the minimum information that is required to visualize the modules. Additional information of the regulatory program can also be given as input to our tool and visualized. This information can be inferred by the module detection tool. In that case it represents a module property. The regulators 1 and 2 (R1 and R2), for instance, were assigned to the blue module by a module inference tool and are thus examples of properties of this module. In addition to module properties, additional information can be derived from curated databases or other datasources: in that case it represents a gene property. In the example, the presence of a motif (M1-M4) is indicated by a score ranging from 0 to 1. The higher the score the more likely the presence of the motif. Similarly the physical binding of a regulator (R1-R4) to a gene as derived from ChIP-chip data is indicated by its p-value. Membership to a gene ontology functional class (GO1-GO3) is indicated by a binary value.

A. *General visualization*

The modules are visualized in a 2D display, called the ModuleImageDisplay, in which the rows represent the genes and the columns the experiments. Each regulatory module is represented in this display, as a transparent colored rectangle. General information on the currently displayed modules such as their gene, experiment, motif or regulator content is shown by our tool.

In addition to this display, two other displays the, GenePropsImageDisplay and ExpPropsImageDisplay, show respectively the gene properties and the experiment properties. Both displays are dynamically linked to the ModuleImageDisplay, meaning that if the order of genes or experiments changes in the ModuleImageDisplay, their order will also change in the other two displays.

The properties that are displayed in the GenePropsImageDisplay are the gene properties (i.e., membership to a gene ontology class, presence of transcription factor binding site or binding of a transcription factor). The rows represent the genes whereas the columns of this 2D display represent the gene properties. For the properties ‘regulator’ and ‘motif’, a color gradient indicates the values of the score for a particular property and gene combination. This score can be derived from, for instance, a motif screening in the case of the motifs or from the results of a ChIP-chip experiment in case of regulator binding. For the gene ontology membership, binary values are available: the gene either belongs or doesn’t belong to the functional class. These gene properties can be included as additional information for the analysis of the regulatory modules, but are not required.

The ExpPropImageDisplay displays the experiment properties. In this image the rows represent the different experiments of the experiments and the columns show the conditional categories to which the different experiments can be assigned.

B. *Finding an optimal layout for the modules*

By using microarray data, module inference algorithms usually retrieve multiple modules that overlap in both genes and experiments. When loaded into the tool, these modules initially will be displayed according to their order in the input module file. This initial non-optimal ordering will result in modules being split up in the ModuleImageDisplay. Indeed by changing the order in which the genes and experiments are displayed, the first modules can be shown completely without being split up. Gradually adding more modules overlapping with the first displayed one reduces the flexibility of reordering, causing the last added module to be split up again. When visualizing multiple modules, it is therefore essential that the modules are placed in such a way that an optimal overview of the results can be obtained.

To improve the visualization of overlapping modules, two different ordering algorithms were included: one is based on the "overlap index" and a second one based on the "Order score". If the user is specifically interested in a module that should not be split up, a user-defined ordering of the modules can also be imposed.

The overlap index is defined as the number of modules a particular module overlaps with. In order to get the optimal layout of the modules, in which as few modules as possible are split up, the module that shows overlap with the largest number of modules, i.e. the module with the largest overlap index, is displayed first. All modules overlapping with this module will be added subsequently. Next, the module with the largest "overlap index" amongst the remaining modules will be selected and placed in the ModuleImageDisplay, and again all modules that overlap with this module are positioned in the layout. This procedure will be repeated until all modules are displayed in the layout.

The second layout algorithm is based on the following "Order score" S:

$$S = \log(\text{module size}) \times \log(\text{overlap area}) \qquad (1)$$

The module size is determined by the product of the number of genes and number of experiments in a module. The overlap area is the area (number of genes x number of experiments) that a module has in common with other modules. Each module will be assigned an order score S. The module with the highest score, a large module showing much overlap with other modules will be positioned first. Subsequently the remaining modules are placed in the ModuleImageDisplay in decreasing order of their score.

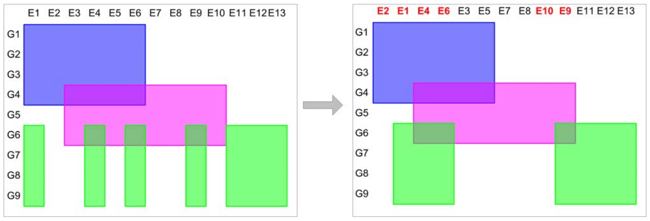

E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13
G1
G2
G3
G4
G5
G6
G7
G8
G9
E2 E1 E4 E6 E3 E5 E7 E8 E10 E9 E11 E12 E13
G1
G2
G3
G4
G5
G6
G7
G8
G9

Figure 2. Sorting experiments in order to obtain an optimal layout. Three modules are shown. The rows represent the genes and the columns the experiments. Once the order of positioning the modules in the 2D visualization is determined, according to the overlap index or the order score function (in this case first the purple module with its overlapping blue module, then the green module), experiments can be resorted to improve the layout. In our example the left panel shows that although the order of the experiments for the blue and purple module is perfect, the green module is still split up in 5 parts. After reordering the experiments (experiment E1, E2, E4, E6, E9 and E10), the green module consists of only 2 parts, while the layout of the blue and purple modules remains unchanged.

Based on either the overlap index or the order score, the optimal order in which the modules are placed in the ModuleImageDisplay is determined. After the positioning of each module an additional resorting step of the experiments is applied to finalize the layout optimization as is illustrated in Figure 2.

C. *Additional functionalities*

1) Selection of modules, genes and experiments in the ModuleImageDisplay

Although all modules resulting from a biclustering or module detection method can be visualized simultaneously, the user might also want to zoom in on a specific subset of modules. A subselection of modules can be made from the complete set of modules or from the currently displayed modules. For the selected modules, the layout can also be optimized using the ordering algorithms mentioned above (B. Finding an optimal layout for the modules). Several module selection criteria are provided:

- Selecting all modules that overlap with one module of interest. Overlapping modules can be defined based on overlap in experiments, overlap in genes or overlap in both genes and experiments.
- Selecting all modules to which the same regulator or the same motif has been assigned by a module detection algorithm.
- A user-defined selection of modules.

In addition to these module selection criteria, our tool also includes the possibility to further filter the output. In contrast to the selection procedures, the filtering options will always function on the currently displayed modules which allows for sequential filtering according to several criteria. The filtering techniques include:

- Filtering of genes: genes can be filtered based on their gene properties (see also A. General visualization). If a gene does not satisfy the user-defined criteria, it will not be visualized.
- Filtering of experiments: experiments can be filtered based on their experiment properties (see also A. General visualization). This allows the user to only visualize those experiments that measure the same cue.
- Filtering of modules. Several criteria are provided for filtering the modules, such as the number of genes or experiments contained within a module, the module size (genes x experiments) or the presence of a particular gene/experiment in the module. In addition, based on the motif/regulator score our tool allows selecting those modules for which a motif/ regulator is present in all genes of the module.

2) Selection of gene properties in the GenePropsDisplay

Sorting the gene properties helps with the biological interpretation of the modules visualized in the ModuleImageDisplay. Motifs can, for instance, be ordered according to their score for the genes in the currently displayed modules in order to investigate the modules' regulatory program. We have included two options for ordering the gene properties in the GenePropsDisplay:

- Based on the score of the regulators/motifs. The higher the scores of a particular regulator/motif are for the genes in the currently displayed modules, the higher these regulators/motifs will get ranked in the list of gene properties.
- Based on the assignment of regulators/motifs to the modules. It is possible to only show those regulators/motifs that were assigned by a module detection algorithm to the currently displayed modules.

3) Additional visualizations

The general display consisting of the ModuleImageDisplay, GenePropsImageDisplay and ExpPropsImageDisplay, is used to display in detail a selection of modules, their genes and experiments together with their properties. Depending on the modules' size and number, the ModuleImageDisplay can usually only display a partial view of the selected modules in one window. Interactively navigating through the ModuleImageDisplay allows to see the rest of the selected modules into detail. The OverviewDisplay, given in a separate window, provides a less detailed but total overview of all currently displayed modules and allows the user to keep track of which part of the module selection is currently displayed in the ModuleImageDisplay.

Our tool also provides two ways of viewing the expression values of the genes in the modules. First, as a heatmap of the expression values in the ModuleImageDisplay (low expression values are colored green, while high expression values are colored red). Secondly, by means of the average expression profile of the genes in a module in a separate window.

4) Export

When a group of interesting modules is selected from the total list of modules, this list of modules can be exported as a module XML file, which can then be reloaded in a next session. Once an optimal layout of the modules is obtained, the resulting image can be saved as a figure in SVG or PNG format.

III. Result

To demonstrate that our tool can assist a user in analyzing the output of a module detection tool, it was applied on the results obtained by DISTILLER [6]. DISTILLER is a data integration tool that uses expression data and regulatory motif data to identify modules together with their regulatory program. When applied to *E. coli* data, overlapping regulatory modules were obtained together with (a) motif(s) assigned by DISTILLER to each separate module. In addition to the motifs assigned by DISTILLER, we screened all genes in the dataset for the presence of known regulatory binding sites according to RegulonDB [15]. The regulatory modules together with their assigned motifs (module properties), and the additional motif information obtained by motif screening (gene properties) were used as input.

To be able to fully exploit all possibilities of our tool, additional information for these modules was included. The following gene and experiment properties were added to the input file: the functional classes to which module genes belonged and the conditional classes of which module experiments are part of. Expression data for the obtained modules were available in a separate expression data file.

When loading the modules and the additional information, all modules are initially displayed according to their order in the input file, resulting in a scattered representation of the modules. By using one of the ordering algorithms of our tool, a more optimal layout of the modules is obtained from which the overlap structure of the different modules becomes clearer. Subsequently, we selected modules to which the module detection tool has assigned the motif CRP (see Figure 3). In the GenePropsImageDisplay, the scores of motifs for the genes in the CRP modules are shown. When sorting the motifs in this display according to their scores, it is clear that in addition to the algorithmically assigned CRP motif, also the ArcA motif is important for at least one module and the FNR motif for a second module, as both motifs had high scores for all genes in their respective modules. The ExpPropsImageDisplay shows that many experiments in which the genes of these two modules were co-expressed belong to either the conditional category "carbon-source" or the "anaerobiosis_aerobiosis". These findings are consistent with the known functions of the assigned regulator CRP. The catabolite repressor is known to be active during glucose starvation and known to interact with the regulators ArcA and FNR in response to oxygen [15].

IV. Conclusions

We have developed a user-friendly tool for the visualization and analysis of transcriptional modules and their regulatory program. The previous example shows how our tool allows zooming in on and studying in detail a subset of modules and their properties while maintaining an overview on how the different modules are related to each other. This interactive exploration of results can help biologists in the interpretation of the many modules that are present in the output of module inference tools

V. Appendix

The most recent version of our tool, user manual and an example data set can be found on the following website: ftp://ftp.esat.kuleuven.be/sista/marchal/ViTraM/Index.html

Acknowledgment

We thank Jan Bollen and Tim Bottelbergs for their kind assistance. This work is supported by: 1) KUL: GOA AMBioRICS, GOA/08/011, CoE EF/05/007, SymBioSys; CREA/08/023 2) IWT: SBO-BioFrame; 3) IUAP P6/25 (BioMaGNet); 4) FWO IOK-B9725-G.0329.09; 5) ZKB8933/CREA/08/023/ BOF.

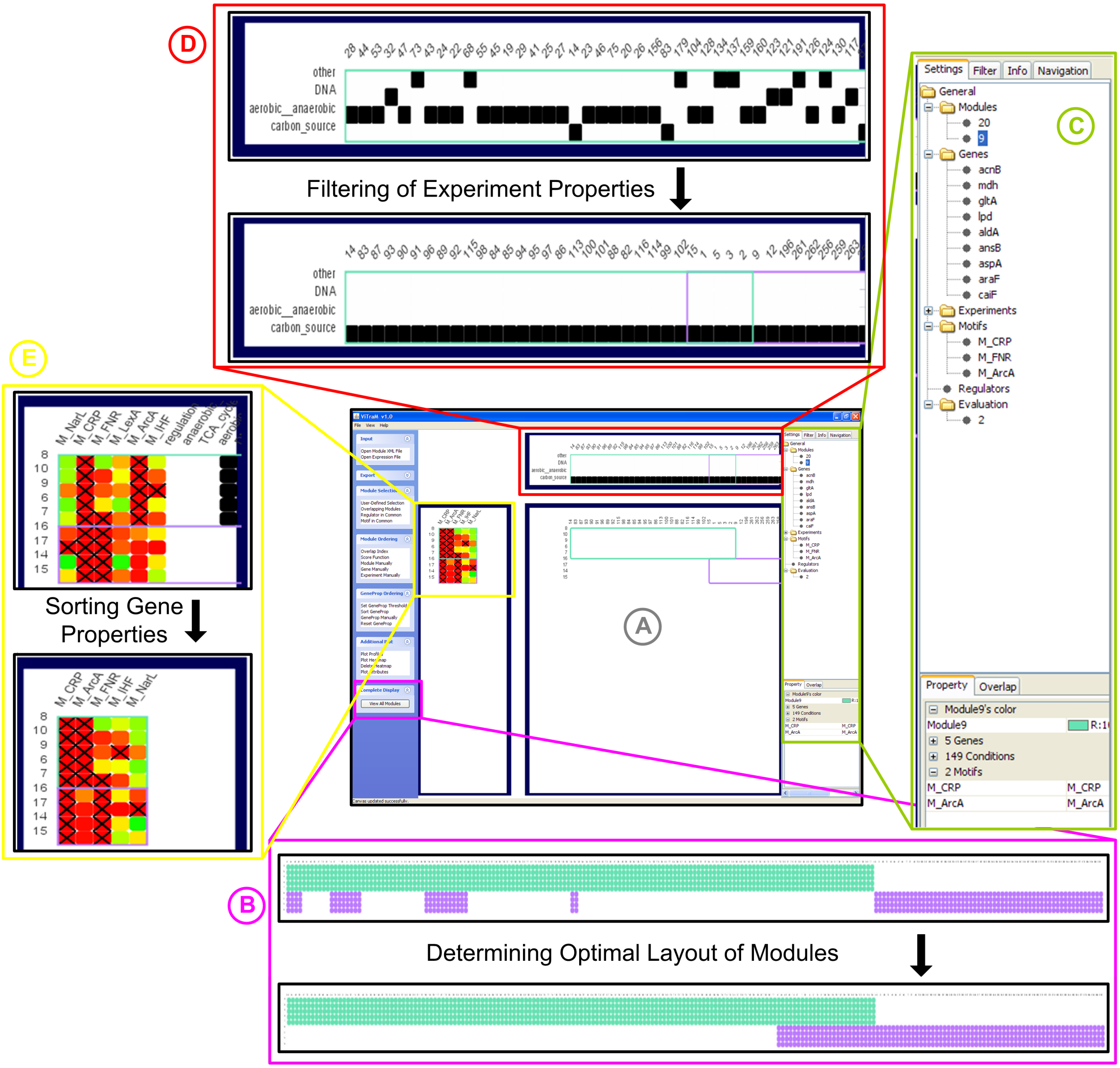


Figure 3. Overview on some of the functionalities of our module visualization tool. All modules obtained from the DISTILLER module detection tool were loaded . Subsequently modules to which the motif CRP was assigned by DISTILLER were selected by using one of the module selection techniques. This resulted in the selection of two modules. (A) The ModuleImageDisplay allows to investigate the two selected modules into more detail: which genes and experiments are present in one or both modules. Because the modules contain many experiments, the second module is not completely visible in the ModuleImageDisplay. Interactively navigating through the ModuleImageDisplay is, however, possible and allows to see all modules into detail. (B) An overview of the modules can be seen in the OverviewDisplay. Initially, one of the modules is split up in five parts. After applying one of our ordering methods, all modules can now be displayed in a coherent way. (C) More information on the currently displayed CRP modules, such as the genes, experiments, motifs and regulators assigned to the modules, can be obtained from the information panel. (D) The ExpPropsImageDisplay shows the experiments present in the modules and their properties. A selection of experiments that measure for instance the influence of the carbon source (category carbon_source) was made. (E) The GenePropsImageDisplay shows the gene properties. Scores for the binding of a regulator or the presence of a motif are indicated by a color gradient in which green is the lowest value and red the highest value. The cross indicates those scores that satisfy a pre-defined threshold. In this example we choose the threshold for the motif scores to be in between 0.999 and 1. The gene properties can be sorted based on the motif scores. After sorting it is clear that the CRP motif has indeed a high score for both modules. For the first (upper displayed module) the ArcA motif scores are high, whereas for the second module (lower displayed module) the FNR scores are high.

REFERENCES

[1] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," Nature, vol. 402, Dec. 1999, pp. C47-C52.

[2] N. Guelzim, S. Bottani, P. Bourgine, and F. Kepes, "Topological and causal structure of the yeast transcriptional regulatory network," Nat.Genet., vol. 31, May 2002, pp. 60-63.

[3] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir, "Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data," Proc.Natl.Acad.Sci.U.S.A., vol. 101, Mar. 2004, pp. 2981-2986.

[4] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," IEEE/ACM.Trans.Comput.Biol Bioinform., vol. 1, Jan. 2004, pp. 24-45.

[5] K. Lemmens and others, "Inferring transcriptional modules from ChIP-chip, motif and microarray data," Genome Biol., vol. 7, 2006, pp. R37.

[6] K. Lemmens and others, "DISTILLER: a data integration framework to reveal condition dependency of complex regulons in *Escherichia coli*," unpublished, 2008.

[7] Z. Bar-Joseph and others, "Computational discovery of gene modules and regulatory networks," Nat.Biotechnol., vol. 21, Nov. 2003, pp. 1337-1342.

[8] E. Segal and others, "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," Nat.Genet., vol. 34, June 2003, pp. 166-176.

[9] X. Xu, L. Wang, and D. Ding, "Learning module networks from genome-wide location and expression data," FEBS Lett., vol. 578, Dec. 2004, pp. 297-304.

[10] K. O. Cheng, N. F. Law, W. C. Siu, and T. H. Lau, "BiVisu: software tool for bicluster detection and visualization," Bioinformatics, vol. 23, Sept. 2007, pp. 2342-2344.

[11] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler, "BicAT: a biclustering analysis toolbox," Bioinformatics, vol. 22, May 2006, pp. 1282-1283.

[12] R. Shamir and others, "EXPANDER--an integrative program suite for microarray data analysis," BMC Bioinformatics, vol. 6, 2005, pp. 232.

[13] G. A. Grothaus, A. Mufti, and T. M. Murali, "Automatic layout and visualization of biclusters," Algorithms.Mol Biol, vol. 1, 2006, pp. 15.

[14] R. Santamaria, R. Theron, and L. Quintales, "BicOverlapper: a tool for bicluster visualization," Bioinformatics, vol. 24, May 2008, pp. 1212-1213.

[15] S. Gama-Castro and others, "RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation," Nucleic Acids Res., vol. 36, Jan. 2008, pp. D120-D124.