ViTraM: VIsualization of TRAnscriptional Modules

Version 1.0

February 15th, 2009

Hong Sun, Karen Lemmens, Tim Van den Bulcke, Kristof Engelen,
Bart De Moor and Kathleen Marchal
KULeuven, Belgium

Contents

1 INTRODUCTION AND INSTALLATION 4
1.1 Introduction 4
1.2 Software structure 5
1.3 Requirements 5
1.4 Installing the software 5
2 INPUT FILES 7
2.1 Module file 7
2.2 Expression data file 9
2.3 Input: Loading the input files 10
3 SOFTWARE ViTraM 11
3.1 Settings Panel 13
3.2 Filter Panel 15
3.2.1 Gene Filter 15
3.2.2 Module Filter 15
3.2.3 Experiments Filter: 17
3.3 Info panel 18
3.4 Navigation Panel 19
3.5 Selection of modules 20
3.5.1 User-defined selection 21
3.5.2 Overlapping modules 21
3.5.3 Motif or Regulator in Common 22
3.6 Optimizing the Layout in the ModuleImageDisplay 23
3.6.1 Overlap Index 24
3.6.2 Score Function 24
3.6.3 Module Manually, Gene Manually and Experiment Manually 24
3.6.4 Constrained reordering of genes/ experiments 26
3.7 Optimize Layout in the GenePropsImageDisplay 28
3.7.1 Set the Gene Properties (GeneProp) score threshold 28
3.7.2 Sort Motifs/ Regulators based on their scores 29
3.7.3 Sort Motifs/ Regulators based on their presence in the modules 30
3.7.4 GeneProp manually 30
3.7.5 Reset GeneProp 30
3.8 Additional Plot 32
3.8.1 Plot Profiles 32
3.8.2 Plot and Delete Heatmap 33
3.8.3 Plot Attributes 33
3.9 Complete Display 35

4 OUTPUT FILES 36
4.1 XML file 36
4.2 SVG or PNG file 36
5 EXAMPLE 37
6 REFERENCES 48

1 INTRODUCTION AND INSTALLATION

1.1 Introduction

Revealing the complete regulatory network underlying the cell's behavior is one of the major challenges in current research. Recently, there is a growing interest in the modular description of regulatory networks. Biclustering algorithms form one type of algorithm for revealing the modularity of the network [1]. A bicluster or a 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 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. 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. 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 this study we developed ViTraM 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, ViTraM 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 regulatory program is by using the information from curated databases. 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 ViTraM. By visualizing not only the modules, but also the regulatory program, ViTraM can provide more insight into the modules and makes the biological interpretation of the identified modules less complicated for biologists.

1.2 Software structure

We aimed at creating a flexible software program. When users write their own plug-ins, they can alter and add functionality without the need to rewrite the whole program. The user can open and close any plug-in implemented in ViTraM by simply flagging the plug-in name under the *"View"* option (Figure 1).

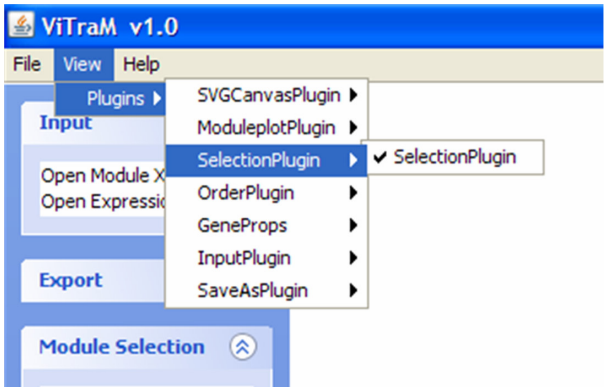

Figure 1: Plug-in structure of ViTraM. The different plug-ins can be opened and closed by using the *"View"* option.

1.3 Requirements

ViTraM runs on both Windows and Linux. Developed in JAVA, it is expected to work under other operating systems that support the JRE (1.5 or higher) and with sufficient memory depending on the size of the input data (minimum 500M).

ViTraM is free to use for academic purposes only. If you use ViTraM in your research, please cite the following publication:

Sun H. *et al.* (2009) ViTraM: Software Tool for the Visualization of Transcriptional Modules.

Also check our accompanying website for recent updates:

ftp:/ / ftp.esat.kuleuven.be/ sista/ klemmens/ ViTraM/ Index.html

For any non-academic purpose, please contact the corresponding author.

Please direct comments and questions related to the software to Kathleen.marchal@biw.kuleuven.be, providing details of your problem while running ViTraM, along with your platform characteristics.

1.4 Installing the software

The software can be downloaded from the download section on:

ftp:/ / ftp.esat.kuleuven.be/ sista/ klemmens/ ViTraM/ Index.html

After downloading the package, please follow these steps:

1. Unzip the downloaded file
2. Open the unzipped folder
3. Depending on the OS:
    - Windows:
        - If you have a small data set, you can directly double click on the file vitram.jar to run the software without assigning extra memory.
        - Or open a command line window, go to the directory where you put ViTraM, and execute the command "java -jar –Xms256m –Xmx512M vitram.jar" in the folder in which the files are stored.
    - Linux:
        - Install X-Win32 or other visualization software for Linux
        - Run ViTraM in a terminal with command "java -jar –Xms256m –Xmx2g vitram.jar".

If everything is OK, ViTraM should start right now and the following window appears:

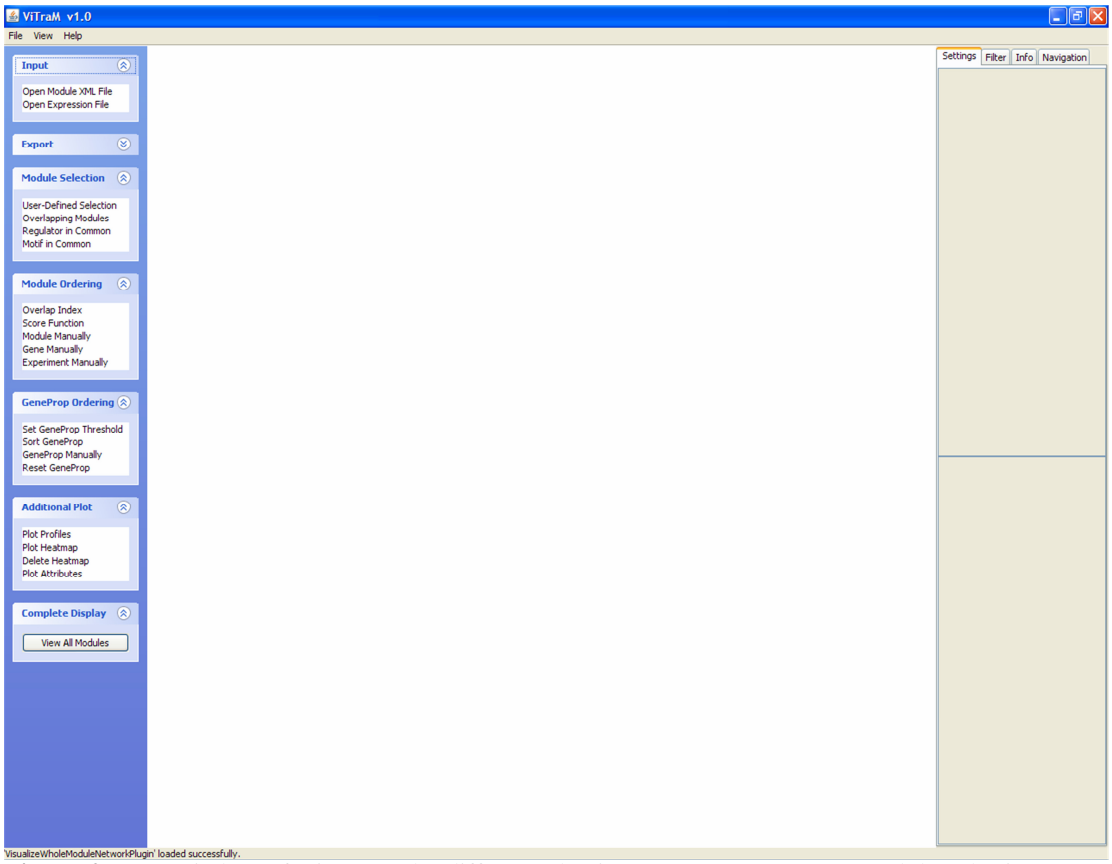

Figure 2: Start window of ViTraM. The different plug-ins "Input", "Export", "Module Selection", "Module ordering", "GeneProp Ordering", "Additional Plot" and "Complete Display" can be seen on the left panel. On the right panel, the "Settings", "Filter", "Info" and "Navigation" options are available.

2 INPUT FILES

ViTraM requires two input files:

- Module file (section 2.1)
- Expression data file (section 2.2)

2.1 Module file

The module file should be loaded first. This module file is an XML file that contains the minimal information to visualize the modules. Optionally the module properties and the gene and experiment properties can be included (see below). The expression values of the genes per experiment are given in the expression data file (see section 2.2). The structure of the 'module file' is shown in Figure 3. This file contains the following information:

- General information: the used xml version and/ or the algorithm that was used for retrieving the modules or the date can be mentioned here.

- Gene property information: This part requires minimally for each gene a unique gene name and id. The id corresponds to the row number in the expression file (see below), such that each gene can unambiguously be linked to its expression values in the expression matrix (one to one relation). The gene ids should start from 1. Optionally, additional gene properties can be added (but this is not strictly required). These optional properties include 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 the gene's upstream region. Additional gene properties are indicated as follows: a binary value is used to indicate whether a gene belongs to a particular gene ontology functional class (0 = absent, 1 = present); the interaction of a transcription factor with a gene is indicated by a binary value or a score (depending on the source of information, ChIP-chip data, motif screening,…).

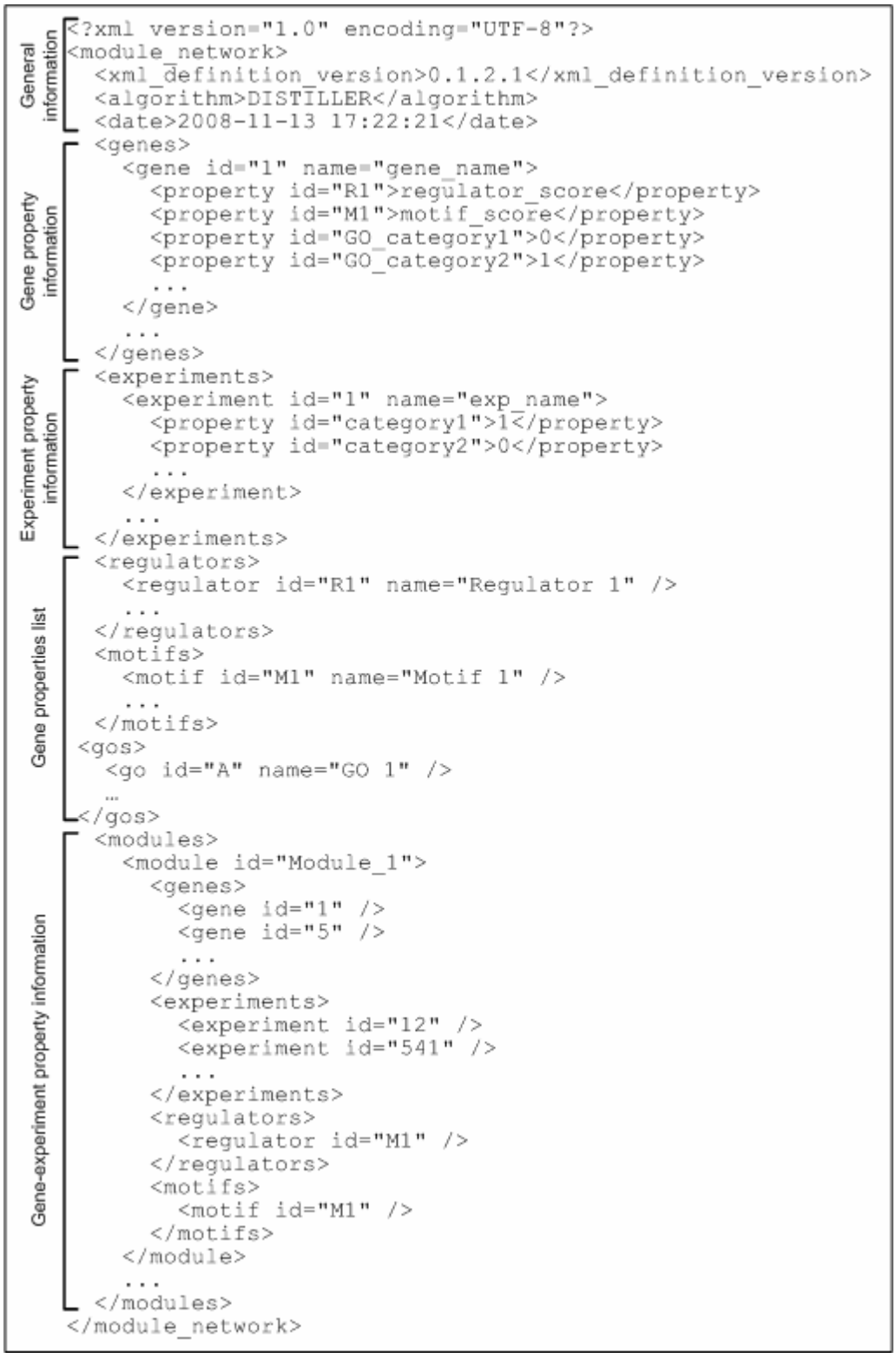

Figure 3: The structure of the module file. First some general information can be listed in the module file. Subsequently, the genes and possibly their properties are listed, followed by the experiments and their properties. Then the different gene properties such as the regulators or motifs are listed. Finally the actual module information is given. Modules minimally consist of genes and experiments, but also additional module properties like a regulator or motif are possible.

- The experiment property information: Again, for each experiment, a name and id are strictly required. The experiment id refers to the corresponding column in the expression data file (see below) and start from 1. Optionally, the user can add experiment properties, such as a classification of the experiments according to the cue that was measured in that experiment. A value "one" for a particular conditional functional class indicates that an experiment belongs to this class.

- Gene properties list: If additional gene properties, such as membership of a gene to an ontology class, a regulator or a regulatory motif, were assigned to the genes and mentioned in the "gene property information", a list of these properties should be provided in this section.

- Gene-experiment information: contains as a minimal level of information the actual composition of the regulatory modules. A regulatory module is defined here as a set of co-expressed genes and the experiments under which they are co-expressed. This part of the file should thus minimally contain the ids of the genes and experiments of which each module is composed. Depending on the inference algorithm that was used, a regulatory program can have been assigned to the module, such as the set of regulators or regulatory motifs that regulate the module. This information is called the additional "module properties". Such additional module properties can be included in the XML file, but are not required.

2.2 Expression data file

The microarray expression data should be loaded in case the user wants to plot the modules' expression profiles or heatmaps. Gene names (ids) or experiment names (ids) are not explicitly mentioned in the file, but the rows of the tab delimited text represent the genes (gene ids, see above) and the columns the different microarrays or experiments (experiment ids, see above). The tab-delimited expression file thus only contains expression levels.

2.3 Input: Loading the input files

The module and expression data input files can be loaded by clicking on respectively the *"Open Module XML file"* or *"Open Expression file"* in the *"Input"* plug-in. A window (Figure 4) will appear and the correct module or expression file can be selected and opened. If the module file is loaded, the modules will be shown in the main window, whereas a message in the left bottom corner ("Expression Data loaded successfully") will indicate when the expression file is loaded.

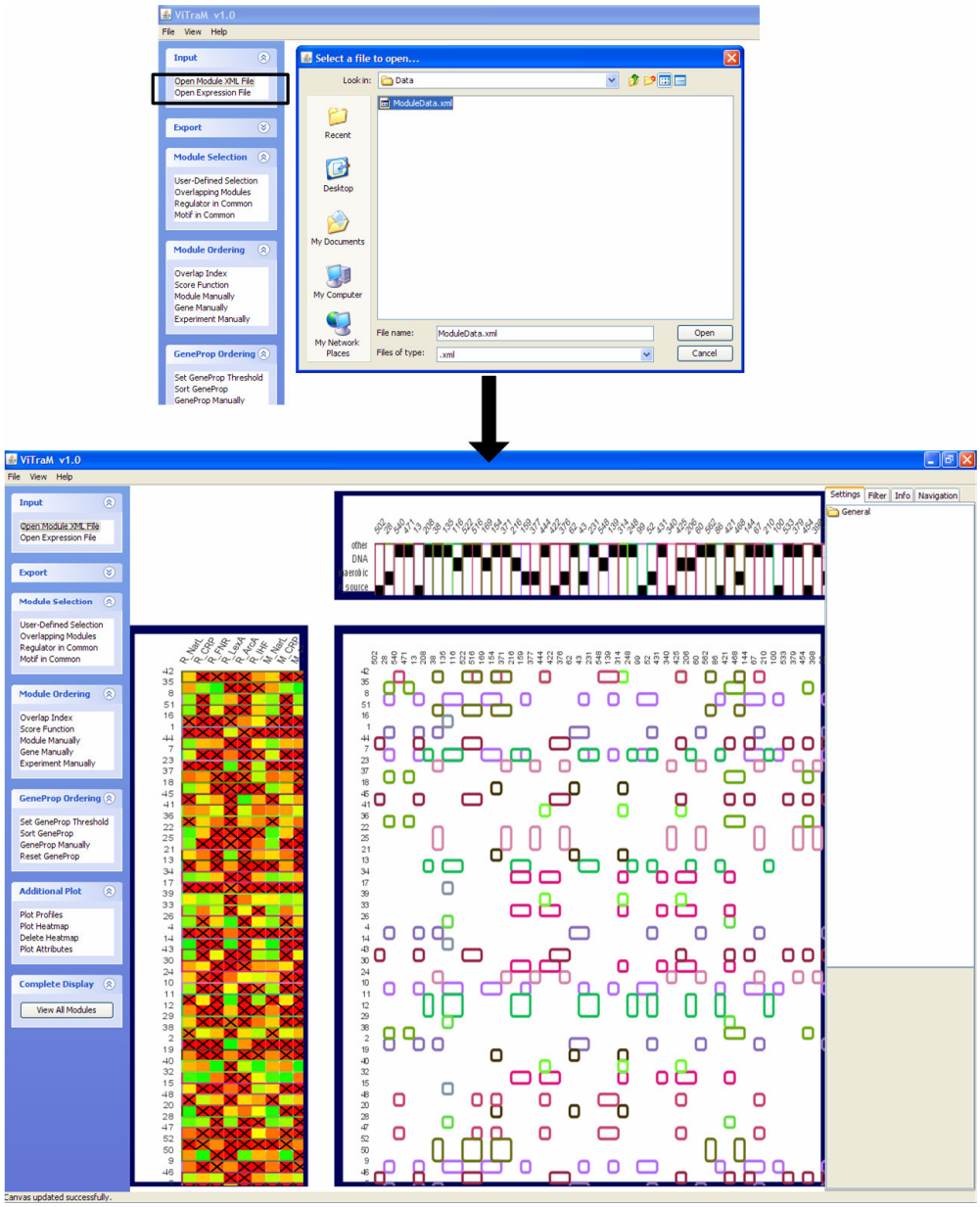

Figure 4: By clicking on "Open module XML file" in the "Input" panel, a pop-up window will appear. The module file can be opened and will be loaded in ViTraM, resulting in the visualization of the modules present in the module file.

Displaying the gene-experiment information

If the module input file is opened, the gene-experiment information will be used to display the modules in the ModuleImageDisplay. In the ModuleImageDisplay, modules are represented as colored rectangles of which the rows represent the genes (indicated by their gene id) and the columns represent the experiments (indicated by their experiment id). Modules will initially be ordered as in the module input file. This will most probably result in modules being split up in several parts in the ModuleImageDisplay. A more optimal way of representing the modules can be obtained by ordering the modules (see section 3.6).

Displaying the additional information

In addition to the ModuleImageDisplay, two other displays are shown. The GenePropsImageDisplay and ExpPropsImageDisplay provide information on properties of the genes and the experiments, respectively (as extracted from the module input file) (see Figure 5).

The rows of the GenePropsImageDisplay represent the genes whereas the columns represent different properties (indicated by their property ids). Currently are implemented as gene properties: the binding of a regulator to a gene ('regulator'), the presence of a motif in the upstream region of a gene ('motif') or the membership of a gene to an ontology class ('gene ontology'). For the properties 'regulator' and 'motif', a color gradient indicates the values of the score for a particular property and gene combination. A cross in the GenePropsImageDisplay indicates whether a certain gene property has a score for a particular gene that is in between a minimal and maximal score, i.e. a score that meets a certain threshold that can be user-specified (section 3.7.1). The default minimum score is set to 0.8, whereas the default maximum is set to 1. For the gene ontology property, a black rectangle will indicate that a gene belongs to that gene ontology class, while a white rectangle is shown otherwise.

In the ExpPropsImageDisplay, the columns represent the different experiments and the rows represent properties of the experiments (indicated by experiment property ids), like the major cue that was measured in the experiment. Again, a black rectangle will indicate that an experiment belongs to a particular experiment property.

The genes and experiments that belong to a particular module are in both displays indicated by rectangles with the same color as the color of the corresponding module in the ModuleImageDisplay. If a module is selected in the ModuleImageDisplay, it will simultaneously be highlighted in the GenePropsImageDisplay and ExpPropsImageDisplay.

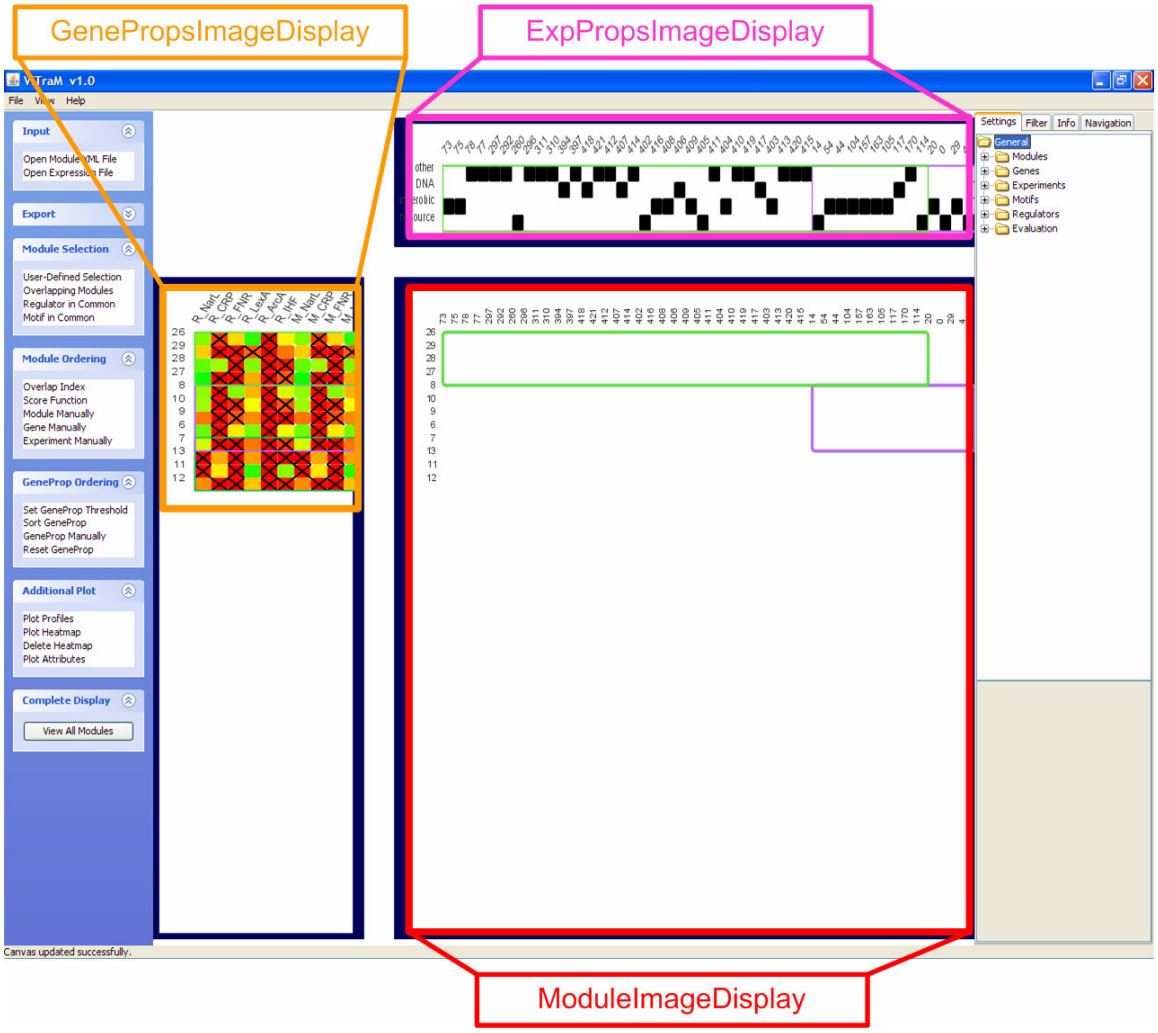

Figure 5: The three image displays are shown: the GenePropsImageDisplay, the ExpPropsImageDisplay and the ModuleImageDisplay. The GenePropsImageDisplay represents the genes in the rows, and the gene properties in the columns. Three gene properties are implemented at the moment: regulator, motif and gene ontology. The scores of each regulator/ motif for a particular gene are indicated by a color gradient. If the regulator/ motif score is above a pre-defined threshold, there will be a cross on the colored square. Membership of a particular gene ontology class is indicated with a black square. The ExpPropsImageDisplay represents the experiment properties in the rows and the experiments in the columns. The experiment properties are functional classes that indicate the cue that was measured in the experiment. If a particular experiment belongs to a particular functional class, this is indicated with a black square. The ModuleImageDisplay represents the genes in the rows and the experiments in the columns. In the ModuleImageDisplay the modules, consisting of genes and experiments, are represented by transparent rectangles of different colors. The gene and experiment order in the GenePropsImageDisplay and ExpPropsImageDisplay are according to the order of the genes and experiments in the ModuleImageDisplay.

In addition to loading and visualizing all modules obtained from the module file, ViTraM also allows to obtain more information on the displayed modules, to select modules that should be visualized and to find an optimal layout of these modules.

The panel on the right side contains four sub-panels:

- Settings Panel (section 3.1)
- Filter Panel (section 3.2)
- Info Panel (section 3.3)
- Navigation Panel (section 3.4)

The panel on the left side consists of seven plugins:

- Input (section 2.3)
- Export (section 4)
- Module Selection (section 3.5)
- Module Ordering (section 3.6)
- GeneProp Ordering (section 3.7)
- Additional Plot (section 3.8)
- Complete Display (section 3.9)

We will discuss these functionalities into more detail below.

3.1 Settings Panel

More specific information about each module that is selected and currently displayed in the ModuleImageDisplay can be obtained from the *"Settings panel"*. By clicking on *"general"*, the following information is available (see Figure 6):

- List of Modules. By clicking on a specific module, information on the (number of) genes, experiments, regulators and motifs of this selected module is provided (below in the "property" panel), as well as the names of the modules with which this selected module overlaps (below in the "overlap" panel). Modules can show overlap in their experiments, in their genes or in both their genes and experiments.
- List of all genes: contains all genes that are part of the modules currently being displayed.

- List of all experiments: contains all experiments that are part of the modules currently being displayed.
- List of all motifs derived from modules currently being displayed.
- List of all regulators derived from modules currently being displayed.

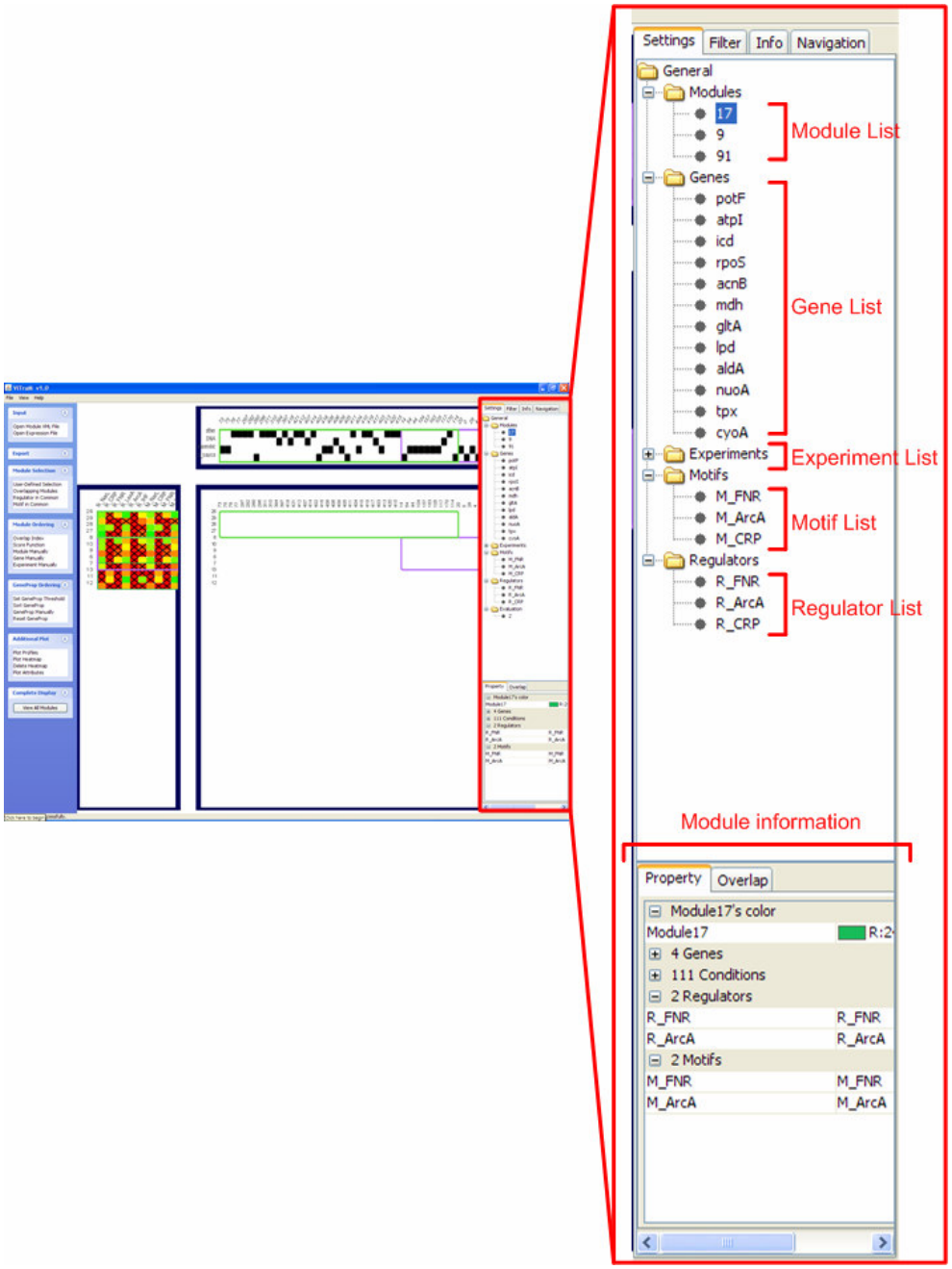

Figure 6: Information available from the *"Settings"* panel. The Settings panel contains information on the modules that are currently displayed in the ModuleImageDisplay. For each module separately, information is provided on its genes, experiments, motifs or regulators. In addition, lists of all genes, experiments, regulators, and motifs present in the currently displayed modules are shown.

3.2 Filter Panel

A filtering strategy is included in ViTraM that allows filtering genes, experiments and modules. The filtering strategy is always applied on the modules that are currently in the ModuleImageDisplay. It is thus possible to filter sequentially based on several criteria (in contrast to section 3.5).

3.2.1 Gene Filter

Genes can be filtered based on their properties such as (see Figure 7):

- regulator score
- motif score
- membership of a gene ontology class

By choosing a minimum and maximum score for a particular motif or regulator, only those genes that meet this requirement will be visualized in the three image displays (GenePropsImageDisplay, ExpPropsImageDisplay and ModuleImageDisplay). Genes can also be filtered based on their membership to a particular gene ontology class. The user can thus choose to only visualize those genes that all belong to the same gene ontology category.

3.2.2 Module Filter

Modules can be filtered based on the following criteria:

- Regulator score or Motif score:

Modules for which all genes have a minimum/ maximum regulator score or motif score can be selected. Note that in contrast to the above mentioned gene filtering, the module will only be displayed if all genes of the module satisfy the chosen requirement, while for the gene filtering, each gene is considered individually.

- Number of genes or Number of experiments

Modules can also be filtered based on a minimum/ maximum number of genes or experiments they should have. As such only the modules that contain a minimum or maximum number of genes or experiments will be shown in the ModuleImageDisplay.

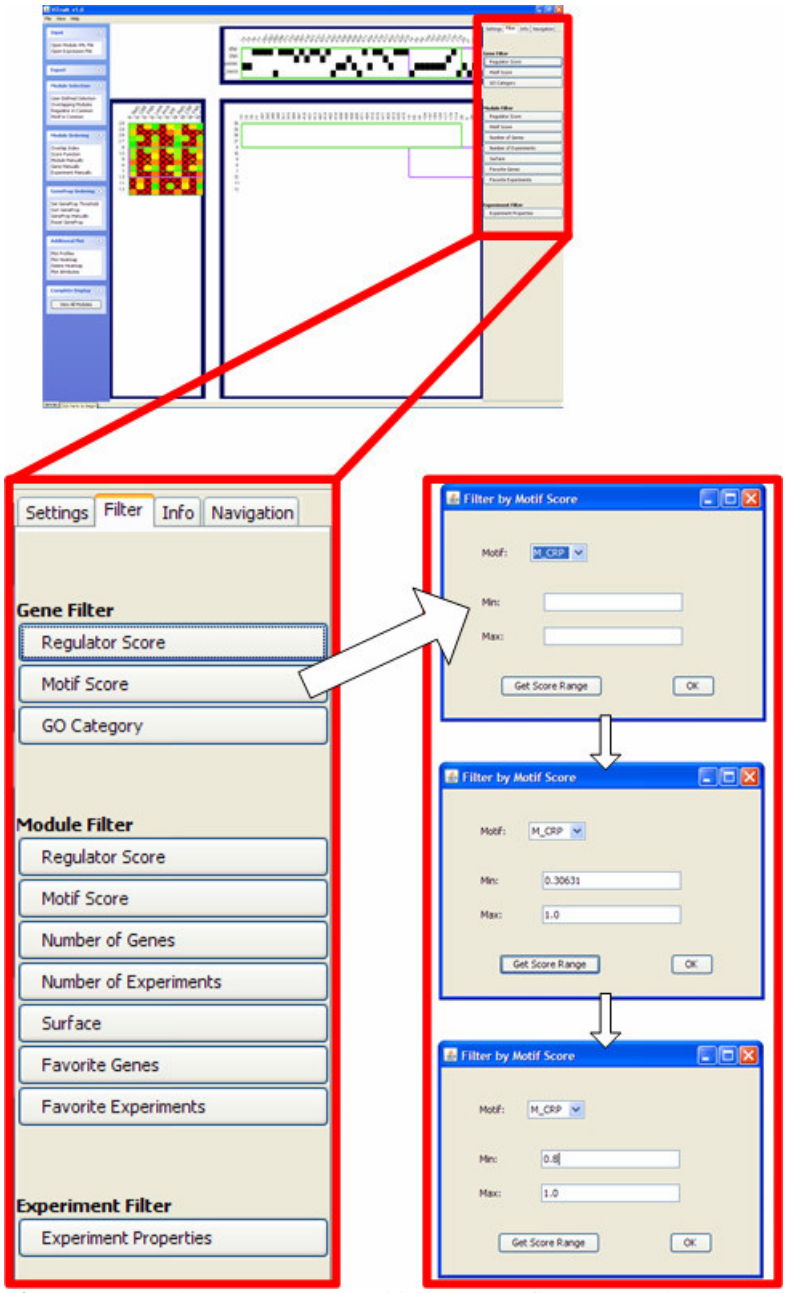

Figure 7: Modules, genes and conditions can be filtered by using the "Filter" panel of ViTraM. The filtering functions always operate on the currently displayed modules. Genes can be filtered based on their regulator score, motif score or whether they belong to a functional class. Modules can also be filtered based on their regulator or motif score. Only if all genes of the module satisfy the motif score or regulator score thresholds, the module will be displayed. A filtering based on the number of genes, number of experiments, or size (genes x experiments) of the modules is also possible. Finally, modules can be selected based on whether a particular gene or experiment is present in the module. Experiments can be filtered based on their experiment properties. Only if an experiment has a particular property, i.e. belongs to a particular conditional class, it will be shown. An example is given in which genes will be selected based on their motif score. First, a motif needs to be chosen, for instance M_CRP. Subsequently, by clicking on "Get Score Range", the user gets to know the minimum and maximum motif score of all genes that are currently displayed. Next, the user can change this minimum or maximum value. By clicking on "ok", genes will only be shown if their motif score for M_CRP is in between the minimum and maximum value.

- Module size

Modules can be filtered based on a minimum surface, i.e. the number of genes multiplied by the number of experiments, they should have.

- Favorite Genes or Favorite Experiments

Modules can also be selected based on a favorite gene or experiment (see Figure 8). This means that only modules containing (a) particular gene(s) or experiment (s) will be displayed. If, for instance, the user has two favorite genes, only those modules containing both genes will be shown.

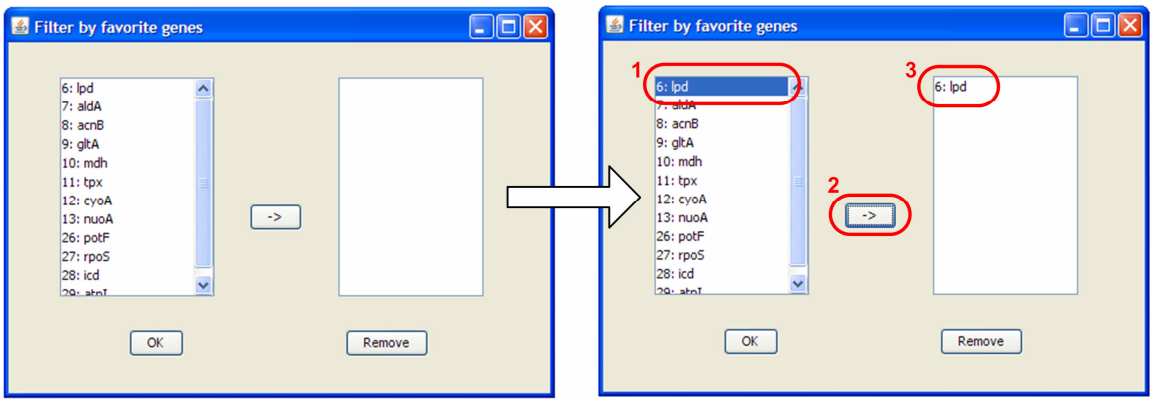

Figure 8: Example of how to filter modules containing a particular gene. The list on the left contains all genes (geneId: geneName). By selecting a gene (1) and clicking on the *"->"* button (2), a gene can be added in the list on the right (3). When clicking on *"OK"*, only modules containing the gene(s) from the list on the right will be displayed. By selecting a gene in the list on the right and clicking on *"Remove"*, this gene will be deleted from the list.

3.2.3 Experiments Filter:

Experiments can be filtered based on the experiment properties. By selecting experiment properties, only those experiments that are currently shown in the ModuleImageDisplay and that have the selected experiment property will be displayed.

3.3 Info panel

The info panel provides general information on all modules present in the module file: it shows how many modules, genes, experiments, motifs and regulators are present in the modules (Figure 9). In addition, the shortcut keys that can be used for navigation of the image displays are shown in the info panel.



Figure 9: The info panel shows general information of the modules present in the module file (red). In addition, the info panel also displays the shortcut keys to manipulate the GenePropImageDisplay, ModuleImageDisplay and ExpPropImageDisplay (blue).

3.4 Navigation Panel

The navigation panel allows scrolling through the image display of the modules, the gene properties and the condition properties, i.e. the ModuleImageDisplay, GenePropsImageDisplay, and ExpPropsImageDisplay (see Figure 10). The red rectangles in the navigation panel show which part of the modules, the gene properties and the experiment properties is currently shown in the three displays. These rectangles can be moved by left clicking and dragging the rectangle. As such, all modules can be viewed.

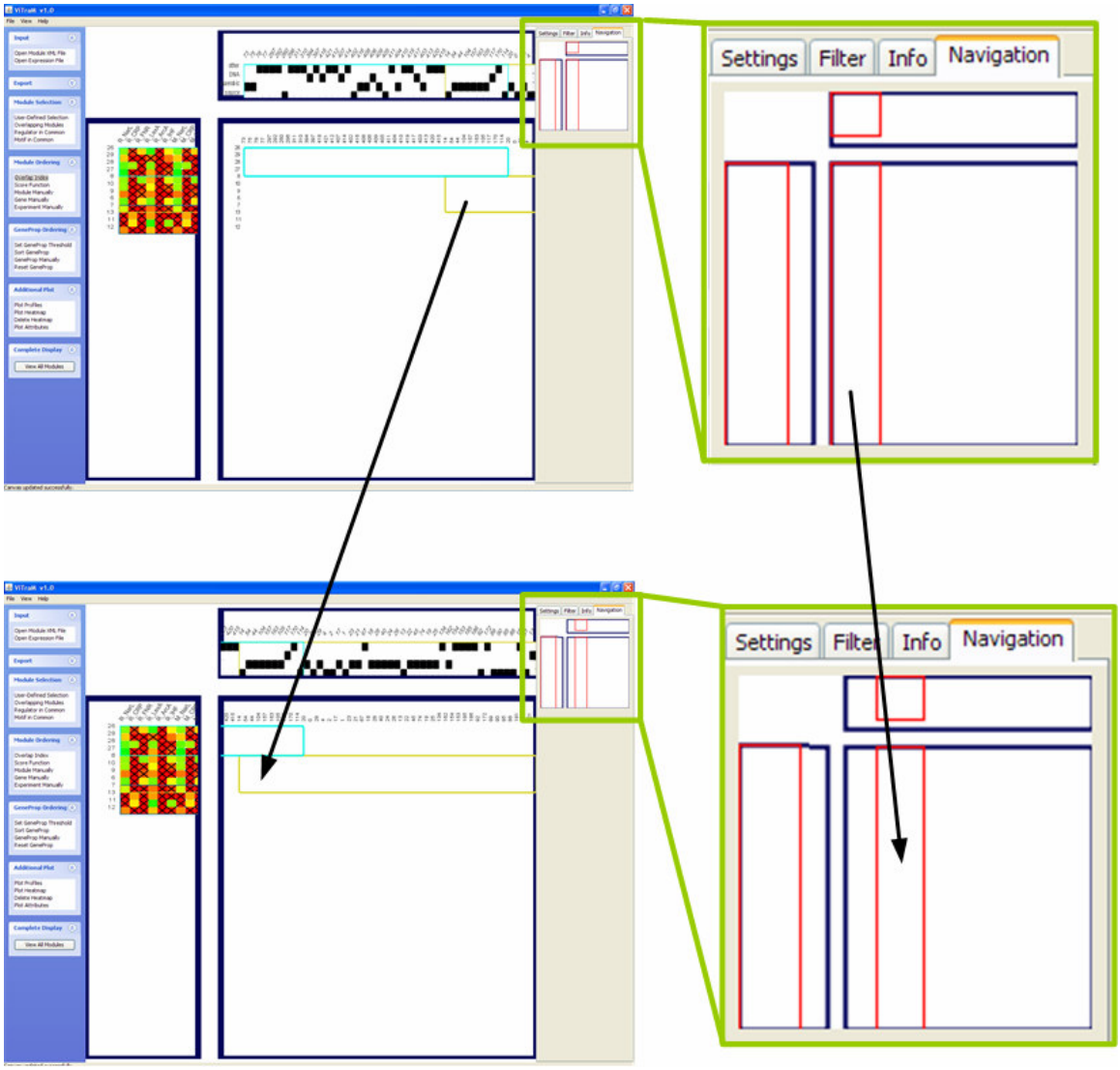

Figure 10: The ModuleImageDisplay can be navigated by using the *"Navigation Panel"*. By moving the red rectangles in the Navigation Panel, the ModuleImageDisplay, GenePropsImageDisplay and ExpPropsImageDisplay can be changed. For instance, when moving the red rectangle corresponding to the ModuleImageDisplay, a different part of the ModuleImageDisplay will be shown.

3.5 Selection of modules

The output of module detection methods usually consists of many modules, showing overlap. Although visualizing all modules is possible with ViTraM, the user might also want to select a specific subset of the modules that should be visualized. Therefore ViTraM includes several module selection techniques in the section "Module selection". These functions always apply on all modules that are present in the module file (see Figure 11):

- User-defined selection (section 3.5.1)
- Overlapping modules (section 3.5.2)
- Regulator or motif in common (section 3.5.3)

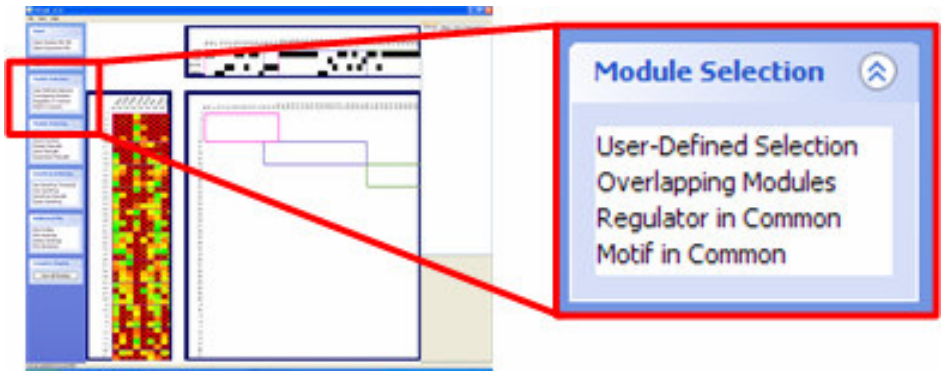

Figure 11: The different possibilities for selecting a subset of modules.

3.5.1 User-defined selection

To visualize particular modules of interest, a user-defined selection of modules is included in ViTraM. By clicking the *"User-defined selection"* button, a pop-up box appears. The user can subsequently enter the names of the modules of interest. By using the *"search"* option, information on the module such as the gene and experiment content is shown. Clicking the *"add"* button will add this module to the selection. Once the modules are selected, click *"OK"*. The ModuleImageDisplay will then be updated and only the selected modules are shown (see Figure 12).

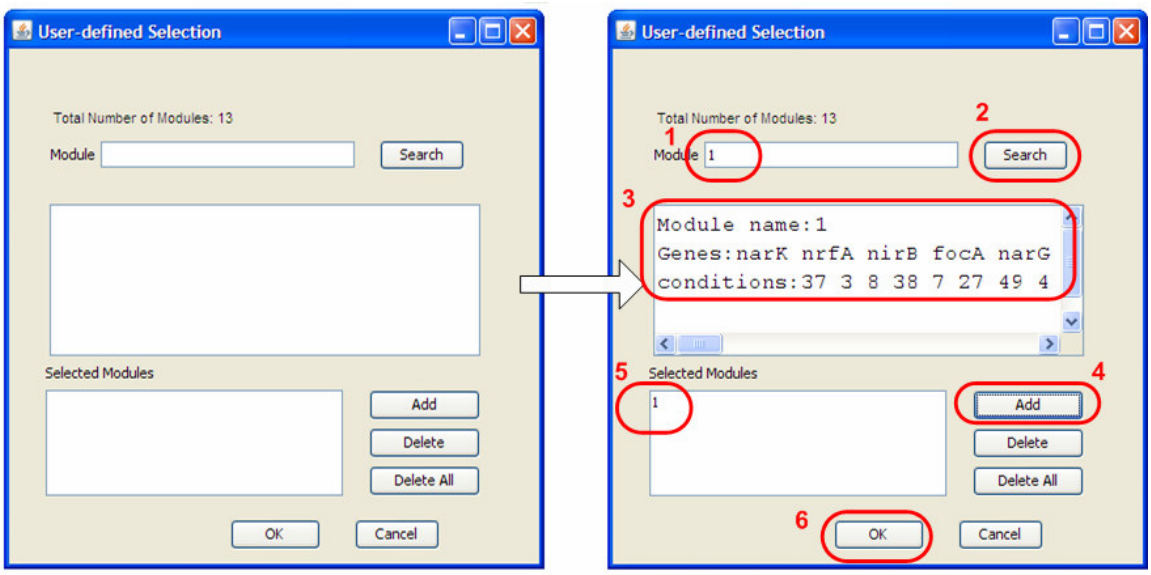

Figure 12: Pop-up window that appears when a user-defined selection is chosen. First type the name of the module (1). By clicking the *"Search"* button (2), the module's information will be shown in the box field (3). The module can be added by pushing the *"Add"* button (4 and 5). If you want to delete modules from the selected list, you can first click on the module's name in the *"Selected Modules"* field, and then click on *"Delete"*. If you want to delete all the selected modules, push *"Delete All"*. Once the selection of the modules is finished, the ModuleImageDisplay will be updated according to the module selection after clicking the *"OK"* button (6).

3.5.2 Overlapping modules

This function allows the user to start from one module of interest and to select all modules that overlap with this module (see Figure 13). The overlap can be defined as modules that overlap in genes, in experiments or in both genes and experiments. Modules are ranked according to the amount of overlap they show with the module of interest. The user can subsequently choose how many of the overlapping modules should be visualized. The GenePropsImageDisplay, ExpPropsImageDisplay and ModuleImageDisplay will be updated with this new selection.

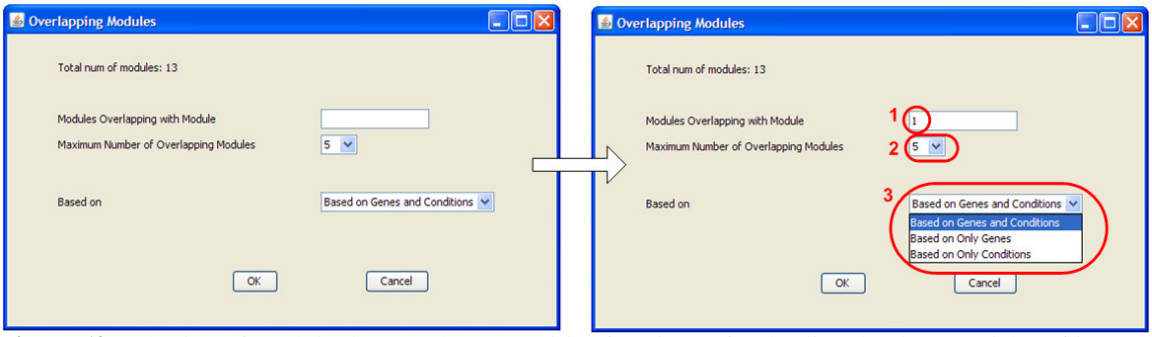

Figure 13: Selection of modules by *"Overlapping Modules"* function. After loading the data, modules of interest can be selected based on their overlap with a particular module of interest (1). The overlap can be based on genes, experiments or both genes and experiments (3). The modules that overlap with the module of interest will be ranked according to their overlap and the maximum number of modules that should be visualized can be selected (2).

3.5.3 Motif or Regulator in Common

This function allows the user to start from one regulator or motif of interest and to select all modules that are regulated by this regulator or that have this motif in their upstream region (see Figure 14). Whether a module has this regulator or motif is determined by the module detection algorithm, meaning that this module property has to be included in the input module file. The GenePropsImageDisplay, ExpPropsImageDisplay and ModuleImageDisplay will be updated after selecting modules that have the same regulator or motif. This function is always applied on all modules that were present in the input module file.

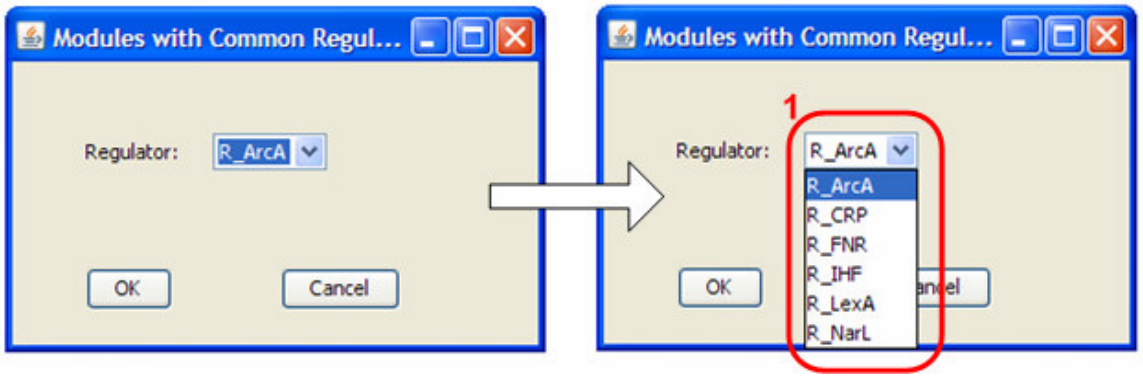

Figure 14: Selecting modules by using the "Regulator in common" function. A pop-up window will be shown after clicking the *"Regulator in Common"* button in the *"Module Selection"* panel. In this pop-up window the motif of interest can be chosen (1). After clicking the *"OK"* button, only the modules that have this motif will be visualized.

3.6 Optimizing the Layout in the ModuleImageDisplay

The goal is to get an optimal layout of the modules in which modules are not split up very frequently in different parts. Indeed, by gradually adding more modules to the display image, we lose the flexibility of some genes and experiments which causes a higher probability for the latter added modules to get split up. A more optimal layout of the different modules can be obtained by changing the ordering in which ViTraM places the modules (Figure 15). Several reordering options are included:

- Selecting an ordering criterion
  - Overlap index (section 3.6.1))
  - Score function (section 3.6.2)

- User-defined ordering (section 3.6.3)
  - Modules
  - Genes
  - Experiments

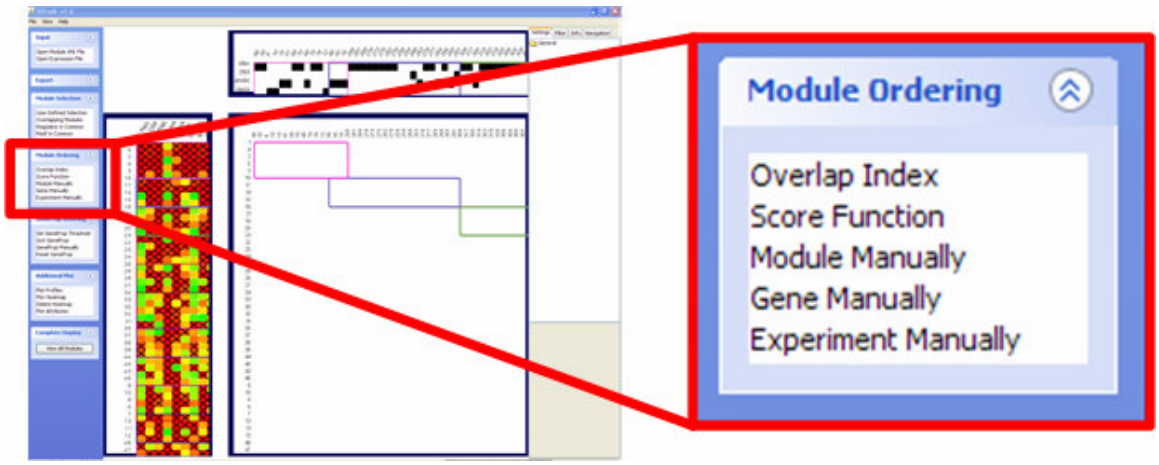

Figure 15: The different possibilities for sorting the modules.

3.6.1 Overlap Index

The overlap index method is an iterative procedure that places the modules that overlap with most other modules first in the ModuleImageDisplay. The procedure starts by calculating for each module the overlap index, i.e. the number of other modules the selected module overlaps with. The module with the largest "overlap index" will be placed first in the ModuleImageDisplay. All modules overlapping with this module will be added subsequently. Next, the module with the largest "overlap index" among the remaining modules will be selected and placed in the ModuleImageDisplay, and again all modules that overlap with this module are given a place in the layout. This procedure will be repeated until all modules are displayed in the layout. After each iteration, genes and experiments within each module will be grouped in the most optimal way as described below (see section 3.6.4).

3.6.2 Score Function

In this procedure, it is the not the number of overlapping modules that determines the ordering of placing the modules in the canvas, but the degree of overlap a particular module shows with other modules. Modules can be assigned a score based on their size (genes x experiments) and the overlap area with other modules. The overlap area is the area (number of genes x number of experiments) that a module has in common with other modules. The order score then becomes: $S = \log(size) \times \log(overlap)$.

The order of positioning the modules in the ModuleImageDisplay is determined according to this order score. The module with the highest score, i.e. a large module that shows a high degree of overlap with other modules will be placed first. Subsequently the remaining modules are placed in the ModuleImageDisplay in decreasing order of their score. Again each time after placing a module, genes and experiments within each module will be grouped in the most optimal way as described below (see section 3.6.4).

3.6.3 Module Manually, Gene Manually and Experiment Manually

Since sometimes the user prefers several genes/ experiments to be displayed together, the user can also manually decide on how to order the modules. If the user is, for instance, particularly interested in one module for which he wants to avoid fragmentation, that module should be placed first (shifted up in the list). If the user wants to manually change the order of the modules, click on the button *"Order Modules manually"* (see Figure 16).

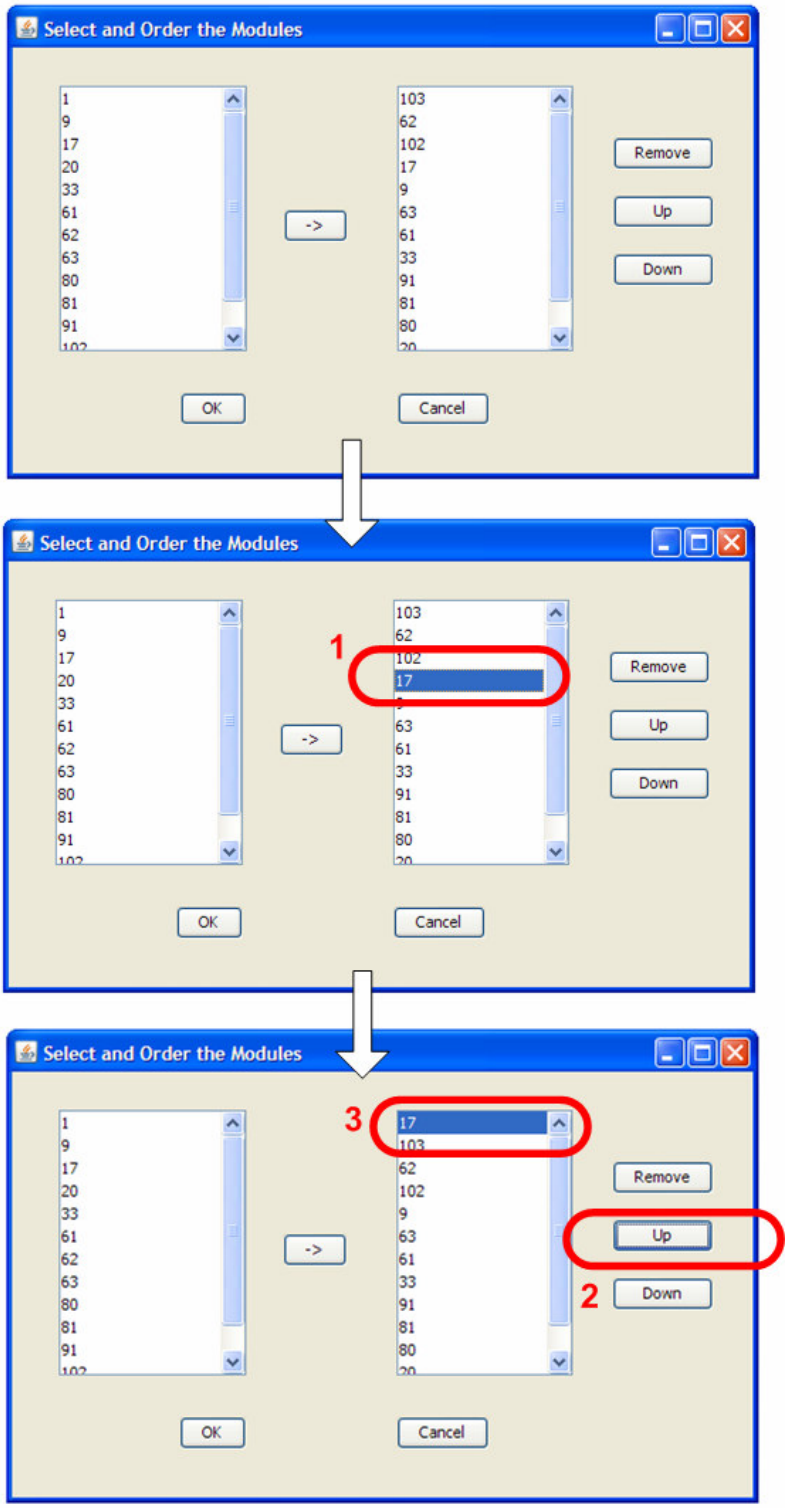

Figure 16: Pop-up window that appears when manually changing the order of the modules. Modules that should be ordered can be chosen from the left side list by clicking *"->"*. The module will then be added to the list on the right, which initially consists of all modules that are currently shown in the ModuleImageDisplay. To change the order in which the user wants to display these modules, move a selected module (1) up or down in the list by clicking *"Up"* or *"Down"* (2). Modules can also be deleted from the list by clicking *"Remove"*. Once the ordering has been determined, the user can click on *"OK"*.

A pop-up window will appear that contains two lists of modules. The first, most left side, list consists of all modules for which data are present in the input files, while the second "display list" consists of the modules that will be shown and the order in which they will be placed on the ModuleImageDisplay. Modules can be deleted from the list by clicking on them and subsequently clicking on the *"Remove"* button. Modules can be added to the display list by selecting them in the left sided list containing all modules and by clicking on *"->"*. The order of the modules can be changed by moving modules up or down in the list using the *"Up"* or *"Down"* button. Once the order has been determined, the user can click on "OK". Genes and experiments will automatically be grouped by "Constraint reordering of genes/ conditions" method (see section 3.6.4). Changing the order of the genes and experiments can be done analogously.

3.6.4 Constrained reordering of genes/ experiments

Once an initial set of modules is placed in the most optimal way, the experiments (genes) can still be rearranged in order to better group the next module. In what follows we will describe the experiment grouping (the gene grouping is analogous) by means of an illustrative example. Figure 17 shows a situation in which Module 1 and module 2 are already positioned in the canvas. Although the grouping is OK for placing both modules 1 and 2, it is still suboptimal for placing a subsequent module (i.e. module 3, green frame). In this case Module 3 shares experiments with both module 1 and 2, but is still highly fragmented. By regrouping the conditions of Module1 and Module2 slightly, we can reduce the splitting. However, we do not have full freedom in regrouping all experiments as then Module 1 and 2 will be split up again. So when rearranging experiments in order to optimally place module 3 we have to take into account that the placing of module 1 and 2 already posed constraints. In this situation the experiments (genes) can be subdivided in four groups: the set of experiments present in only module 1, the set of experiments present in only module 2, the set of experiments shared by module 1 and module 2, the set of experiments that are not present in any module. The constraints posed by placing Module 1 and 2 in advance allow only for rearranging experiments within one subdivision, but not between subdivisions. Experiments within one subdivision that overlap with module 3 can be grouped together reducing the number of splits required for placing Module 3 (Panel 3). If more modules are present, more subdivisions will exist according to which the experiments will be grouped.

When grouping the experiments (genes) into the most optimal way, we designed an iterative procedure that brings these constraints into account one by one. Note that this procedure for the constrained reordering of experiments is an inherent part of finding the optimal ordering of modules based on the "Overlap Index" and "Score Function": each time a novel module is added the constrained gene/ experiment reordering procedure is applied.

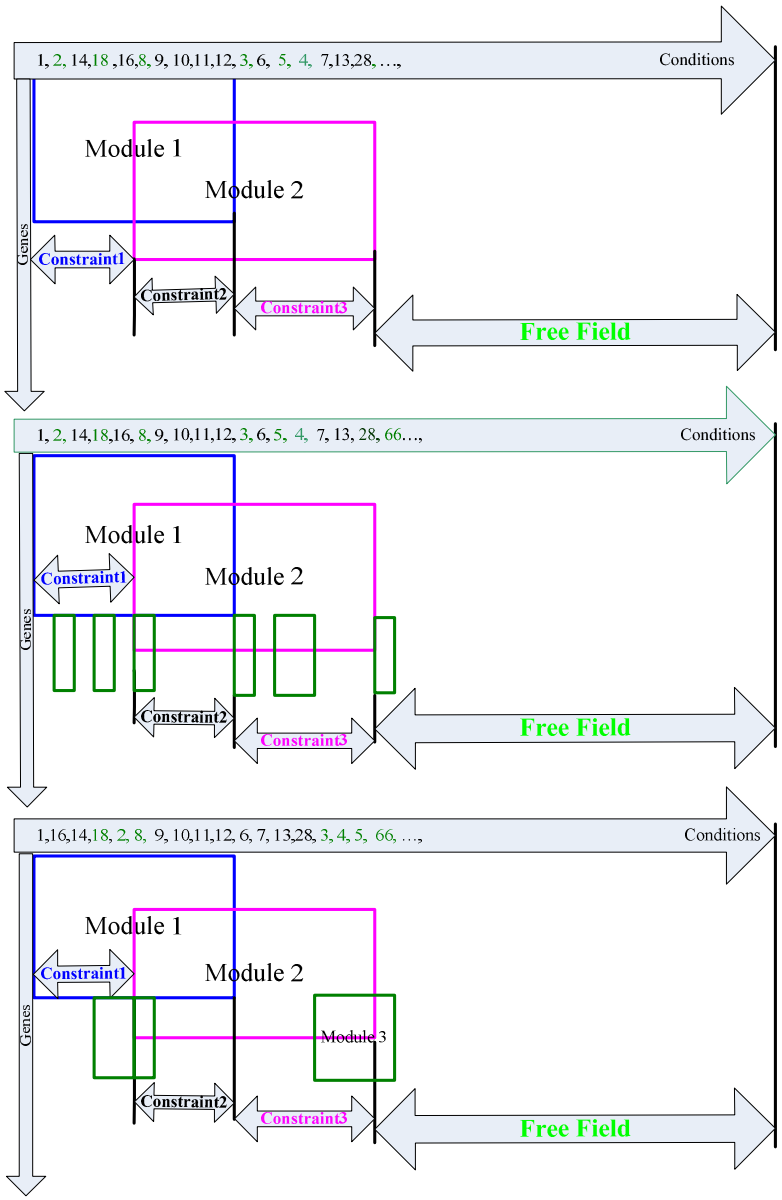

Figure 17: The "Constraint reordering of genes/ experiments". Panel a; Module1 and Module2 are placed in an optimal way. Panel b: Adding module 3 using the current order of conditions will lead to a high fragmentation of module 3. Panel c: the fragmentation of module 3 can be reduced by reordering the conditions, while taking into account the constraints posed by the optimal placing of Module 1 and 2. Experiments can only be freely reordered within one subdivision. Therefore experiments within each subdivision that overlap with module 3 (in this case first 2, 8 and 18, then 3, 4 and 5 will be grouped together. Experiment 66 will be added the latest since it is not overlapping with any of the other two modules. As such, the number of splits needed to place Module 3 is reduced.

3.7 Optimize Layout in the GenePropsImageDisplay

The user can not only manipulate the ModuleImageDisplay, but also the GenePropsImageDisplay. Ordering of the gene properties might be important during the analysis of the modules that are visualized. Motifs can, for instance, be ordered according to their score for the genes in the currently displayed modules. As such the regulatory program of the modules can be investigated. We have included several options for ordering the gene properties (see Figure 18):

- Choosing the regulator/ motif score threshold (section 3.7.1)
- Sorting the regulators/ motifs based on:
  - Their scores (section 3.7.2)
  - Their assignment to modules by the used module detection algorithm (section 3.7.3)
- User-defined sorting of the regulators/ motifs (section 3.7.4)
- Resetting the ordering of the gene properties (section 3.7.5)

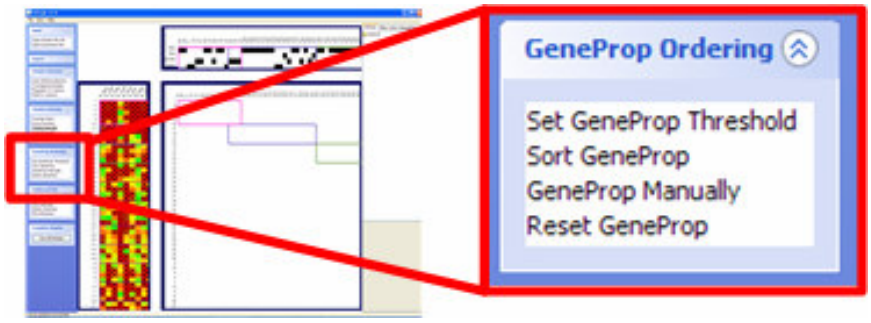

Figure 18: Overview of the different possibilities for ordering the gene properties.

3.7.1 Set the Gene Properties (GeneProp) score threshold

By clicking the button *"Set GeneProp Threshold"*, the user can set the minimum and maximum threshold for the gene properties (GeneProp) motifs and regulators (see Figure 19). The default threshold is set to 0.8-1.0 for motifs and regulators. The genes for which the regulator/ motif score falls in between the minimum and maximum score will be indicated in the GenePropsImageDisplay by a cross.

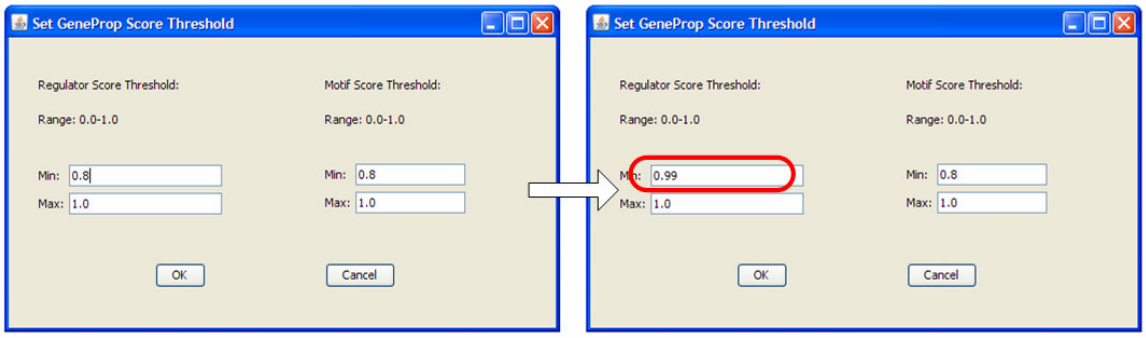

Figure 19: Pop-up window for setting the score thresholds of the gene properties. The ranges of the scores for regulators and motifs are shown. The default score will be shown in the text field when opening this window. A new minimum and/ or maximum score can be chosen for both regulators and motifs.

3.7.2 Sort Motifs/ Regulators based on their scores

As scores for many regulators/ motifs could be available, it is interesting to be able to sort the regulators/ motifs based on their scores. Regulators or motifs in the GenePropsImageDisplay can be ordered based on their scores for the genes in the modules currently displayed in the ModuleImageDisplay. As such the user can immediately see which regulators/ motifs are important for the currently displayed modules.

By clicking on the "Sort GeneProp" in the "GeneProp Ordering" panel, the user can choose to order either the regulators or the motifs (see Figure 20):

- Sort Motifs based on Motif Score
- Sort Regulators based on Regulator Score

The top 5 regulators/ motifs will be shown in the GenePropsImageDisplay.

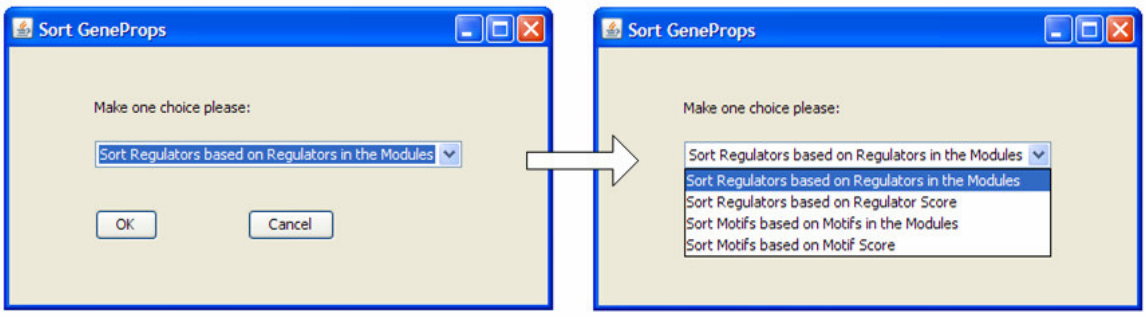

Figure 20: The pop-up window that appears when choosing the "Sort GeneProp" in the "GeneProp Ordering" panel. The user can choose to order regulators/ motifs based on their scores or based on their presence in the currently displayed modules. After clicking "OK", the regulators/ motifs will be sorted according to the selection option.

3.7.3 Sort Motifs/ Regulators based on their presence in the modules

Regulators or motifs can also be assigned to modules by the module detection algorithm (thus independent of their scores). This information can be included in the input module file, and it is also possible to order regulators/ motifs based on their presence in the modules.

By clicking on the "Sort GeneProp" in the "GeneProp Ordering" panel, the user can choose to order either the regulators or the motifs (see Figure 20):

- Sort Motifs based on Motifs in the modules
- Sort Regulators based on Regulators in the modules

3.7.4 GeneProp manually

Finally, it is also possible to change the ordering of the gene properties manually by clicking on the button *"GeneProp Manually"* in the "GeneProp Ordering" panel (see Figure 21). A pop-up window will appear that contains two lists of gene properties. The first (most left side) list consists of all gene properties that were originally in the module input file, while the second "display list" consists of the gene properties that will be shown and the order in which they will be placed on the GenePropsImageDisplay. Gene properties can be deleted from the display list by selecting them and subsequently clicking on the *"Remove"* button. Gene properties can be added to the "display list" by selecting the gene properties in the left side list containing all gene properties and subsequently clicking on *"->"*. The order of the gene properties in the display list, and hence in the GenePropsImageDisplay, can be changed by moving the gene properties up or down in the list using the *"up"* or *"Down"* button.

3.7.5 Reset GeneProp

The "Reset GeneProp" button allows showing the initial list of gene properties, consisting of all regulators, motifs and gene ontology classes, in their initial order.

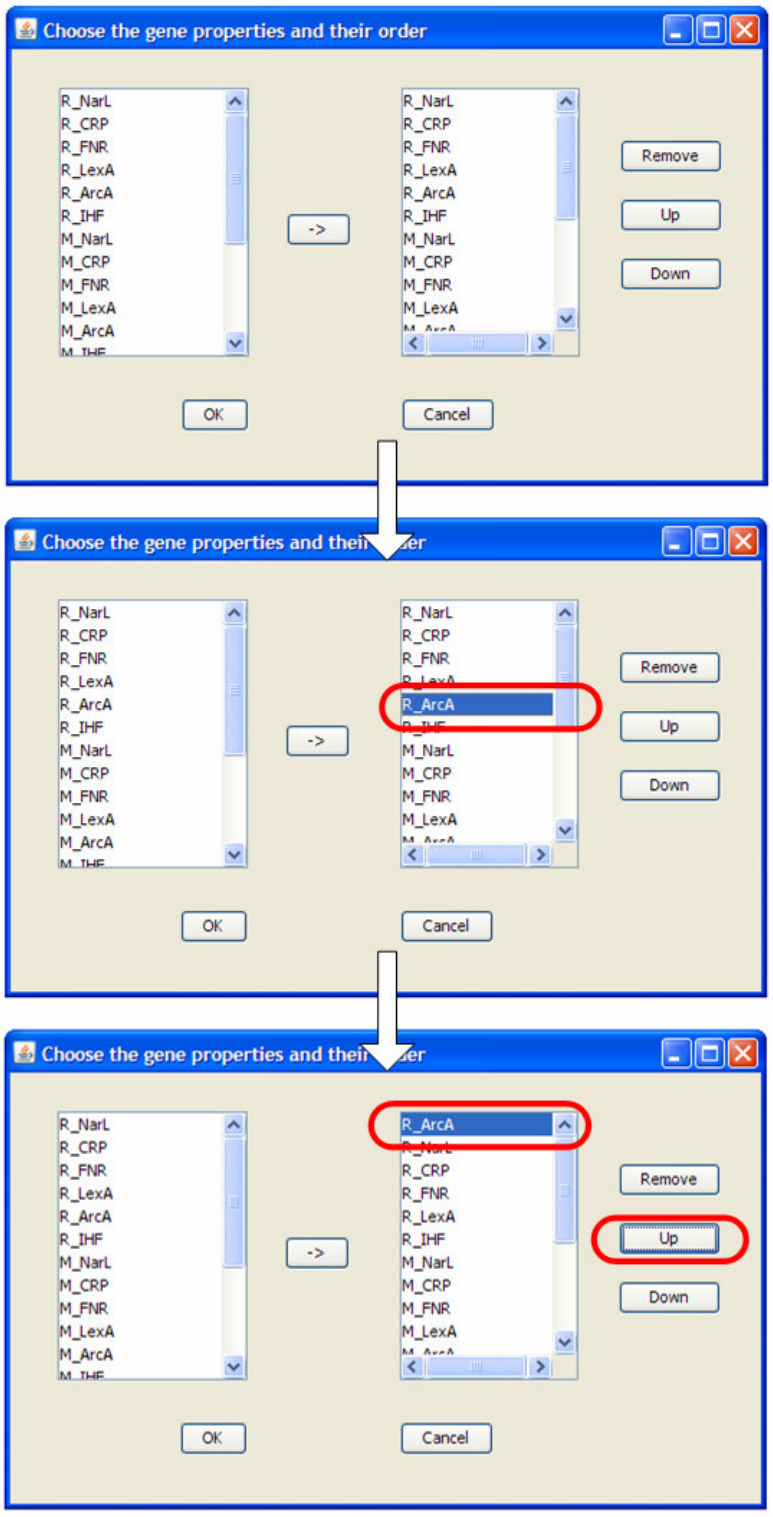

Figure 21: Manually change the order of the gene properties pop-up box. The list on the left contains all gene properties. The list on the right consists of the gene properties that will be shown and the order in which they are displayed. A gene property from the left-sided list can be added to the list on the right by clicking "->". To change the order in which the user wants to display these gene properties, a selected gene property can be moved up or down in the list by clicking *"Up"* or *"Down"*. A gene property will be deleted from the list on the right by clicking on *"Remove"*.

3.8 Additional Plot

In addition to the three image displays, ModuleImageDisplay, GenePropsImageDisplay and ExpPropsImageDisplay, ViTraM allows for some additional plots (see Figure 22):

- Plot of the expression profiles of the modules (section 3.8.1)
- Add/ delete a heatmap of the expression values to the ModuleImageDisplay (section 3.8.2)
- Plot of the regulators/ motifs that were assigned to the modules by the module detection algorithm (section 3.8.3)

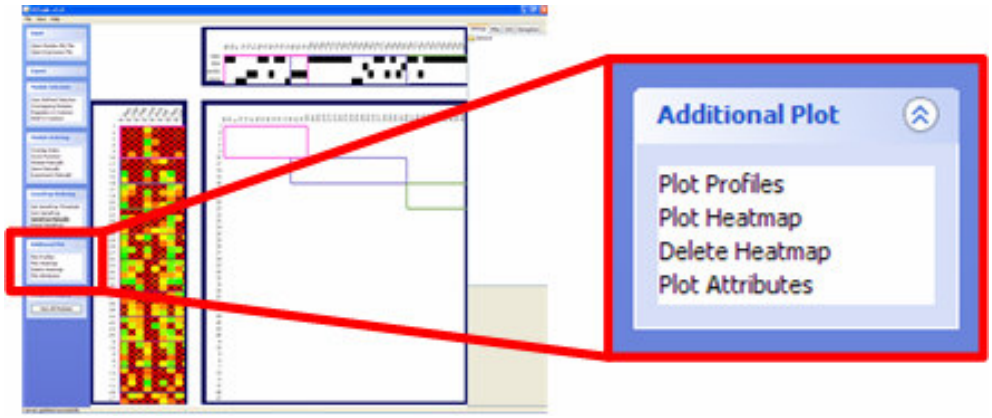

Figure 22: The different additional plots that ViTraM can provide.

3.8.1 Plot Profiles

The average expression profile of the genes in a module is shown after clicking the *"Plot Modules"* button in the "Additional Plot" panel (see Figure 23). The average expression profile is visualized for all modules that are currently displayed. Each module is represented by a separate color. The order of the experiments in this module plot corresponds to the order of the experiments in the ModuleImageDisplay.

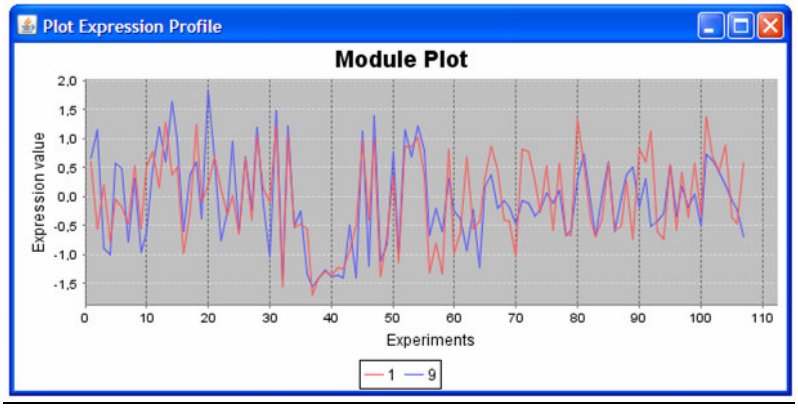

Figure 23: Plot of the average expression profile of the modules currently displayed in the ModuleImageDisplay. The X-axis represents the conditions. The order of the experiments corresponds to the order of the experiments in the ModuleImageDisplay. The Y-axis represents the expression values. For each module, the average expression values of all genes in the module is calculated for each experiment. Each module is represented by one color. The legend at the bottom of the figure indicates which color corresponds to which module. The module 9 for instance, is shown in blue in the graph.

3.8.2 Plot and Delete Heatmap

ViTraM allows visualizing also the expression profile of each gene separately in the modules. By clicking the *"Add Heatmap"* button in the "Additonal Plot" panel, a heatmap will be added to the current ModuleImageDisplay (Figure 24). A heatmap represents the expression values of each gene in each experiment. Low expression values are colored green, while high expression values are colored red. Expression values around zero will be assigned a black color. This heatmap can also be removed from the ModuleImageDisplay by clicking *"Delete Heatmap"*.

Although it is possible to continue analyzing the modules with ViTraM when this heatmap is included, we suggest to always remove the heatmap from the ModuleImageDisplay when the other functions are applied on the data because of memory requirements.

3.8.3 Plot Attributes

To have an overview of how good regulators/ motifs scores are for all genes in a module, ViTraM allows plotting this information by means of the *"Plot Attributes"* in the *"Additional plot"* panel (Figure 25). The rows in this plot represent the different modules and the columns represent the different gene properties, e.g. regulators or motifs. In this plot the average score for a particular regulator or gene for all genes in a module is indicated by a color gradient. The highest values are indicated in red, while the lowest values are indicated in green. For each module, the regulator and motif with the highest average score are indicated by means of a black cross.

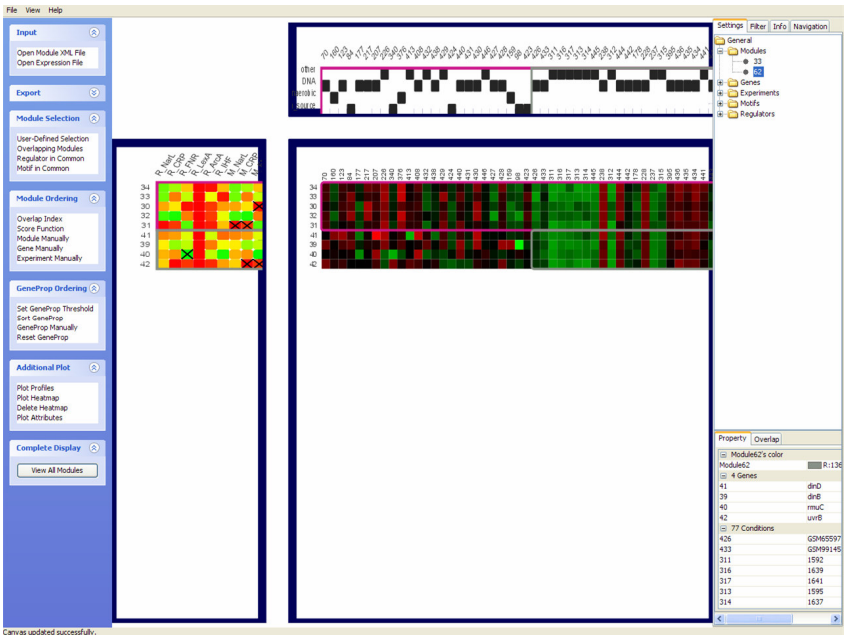

Figure 24: Add a heatmap to the ModuleImageDisplay. The high expression values of the expression data will be colored red, and the low expression values will have a green color. Expression values around zero are colored black. The remaining expression values are represented by a gradient from green to black, and from black to red. The modules are indicated as rectangles, for instance, the blue rectangle on the figure.

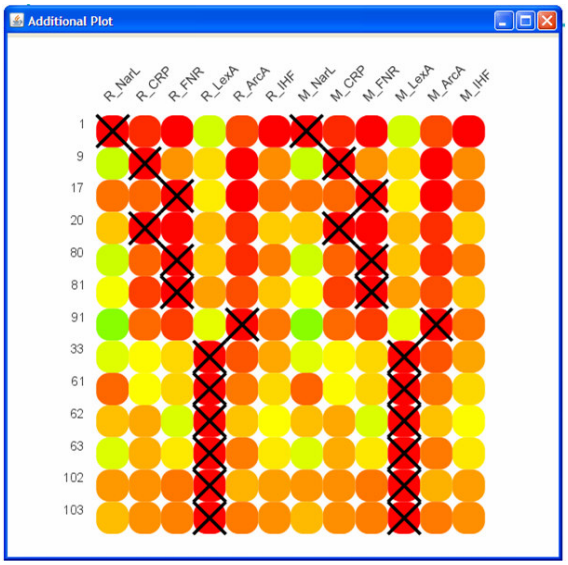

Figure 25: The "Module Attributes" plot. The rows represent the modules, and the columns represent gene properties, i.e. regulators and motifs. For each module-property combination, the average score of the regulator/ motif for all genes in the module was calculated. A color gradient indicates this average score for each module and regulator/ motif combination: red are the highest values, green are the lowest values. A cross indicates for each module the regulator and motif with the highest average value.

3.9 Complete Display

Sometimes, too many modules are included in the ModuleImageDisplay such that it becomes impossible to view all modules as a whole. Therefore, ViTraM allows visualizing all modules that are currently displayed such that an overview on the modules can be obtained. From this overview it might be easier for the user to see which modules show, for instance, overlap in experiments or genes.

By clicking on the *"View All Modules"* button in the *"Complete Display"* panel, the OverviewDisplay is shown (Figure 26). This display shows an overview on all modules in the current ModuleImageDisplay. Whenever the content or layout of the ModuleImageDisplay changes, the overview of all modules will change accordingly. It is possible to zoom in "CTRL + I" or out "CTRL + O" on this overview display.

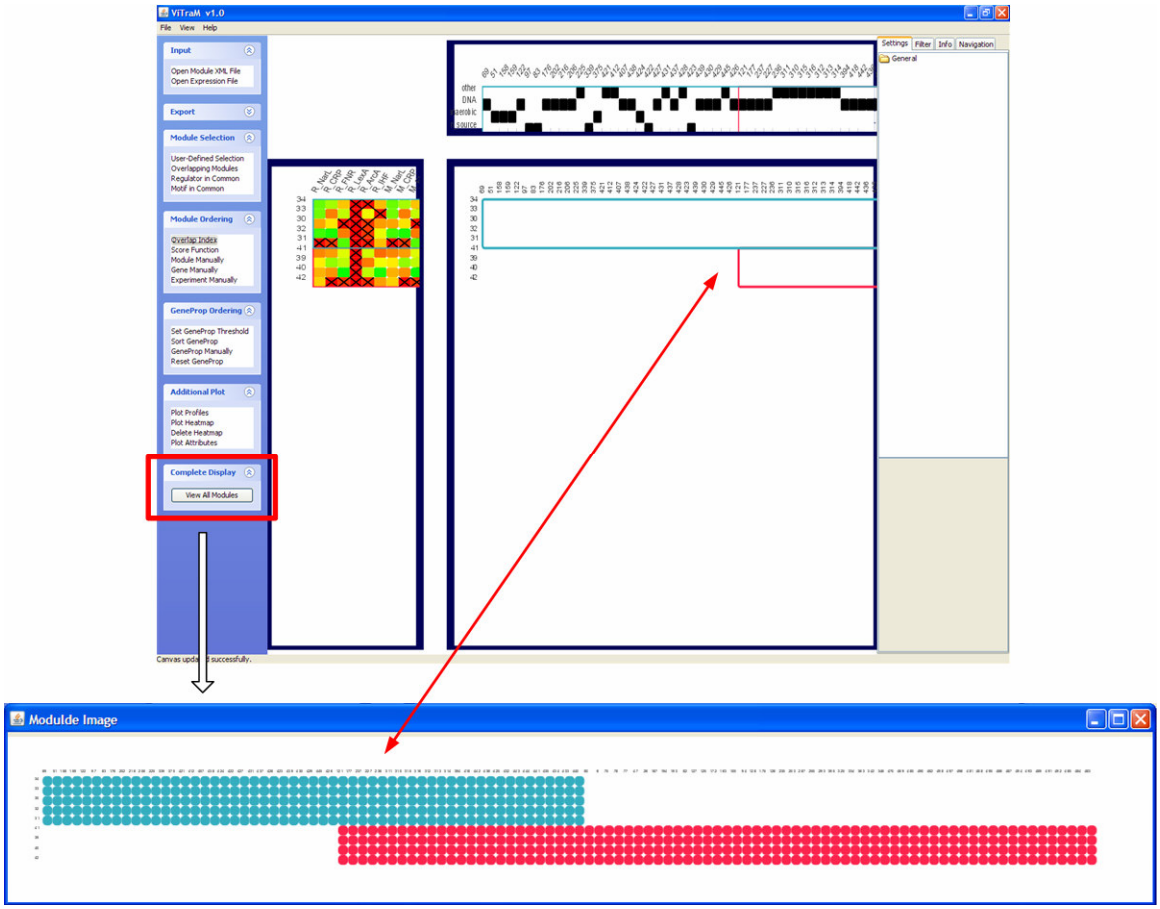

Figure 26: An overview of all modules that are currently displayed in the ModuleImageDisplay can be obtained by clicking on the "Visualize Whole" button in the "Visualize Whole Modules" panel. It is possible to zoom in or out using "CTRL + I" or "CTRL + O". This overview display will automatically be updated according to the changes of the ModuleImageDisplay.

4 OUTPUT FILES

The resulting layout of the modules can be exported to (see Figure 27):

- An XML file
- A figure SVG or PNG file

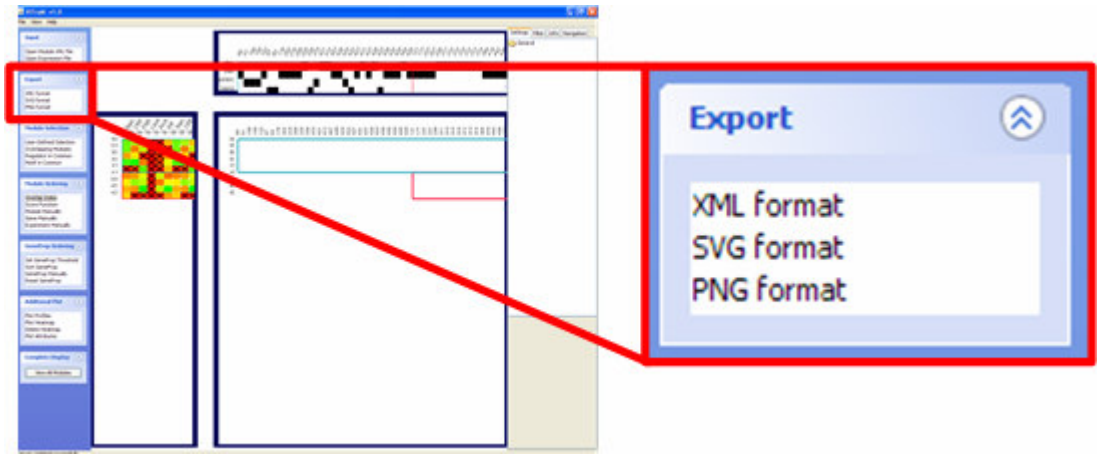

Figure 27: Several possibilities for exporting data from ViTraM are available.

4.1 XML file

In case a group of interesting modules was selected from the total list of modules, it is possible to export these modules as an XML file. This file can then be reloaded in a next session. To save the currently selected modules in an XML file, click on *Export Modules >> XML format*.

4.2 SVG or PNG file

Once an optimal layout of the modules is obtained, it is also possible to save this as a figure in SVG or PNG format. By clicking on *Export Modules >> SVG/PNG format*.

5 EXAMPLE

In this illustrative example, ViTraM was applied on the results obtained by the module detection tool DISTILLER [2]. DISTILLER was applied on *E. coli* data and this resulted in the identification of overlapping regulatory modules and the corresponding regulatory program (in this case the motifs(s) responsible for the observed co-expression = module properties). In addition, the scores of each motif for each gene were obtained from a motif screening (so not automatically assigned by the algorithm). The information on the regulatory modules' composition, the module properties and the gene properties (the motif scores) were included in the module file for ViTraM.

To be able to fully exploit all possibilities of ViTraM, additional information for these modules was included in module file. As DISTILLER assigns only motifs to modules, all functionalities regarding regulatory motifs (module properties) can be used. To allow the user to apply all functionalities regarding regulators on our example data set, the corresponding regulators of these motifs were added to the module file as module properties. As such all functionalities of ViTraM for regulators can also be used.

In addition to these module properties, we also included scores that indicate the binding of a regulator to each promoter region. As such, also all functionalities of ViTraM for the gene property 'regulator' can be used. To obtain these scores, the corresponding motif scores, obtained from a motif screening, were multiplied by 100. As such the flexibility of ViTraM in handling all kinds of data can be demonstrated.

Finally, all genes of a module were assigned to a functional class and all experiments were assigned to some experiment conditional classes. Expression data for the obtained modules were available in the expression data file.

These data can be found on our website:

ftp:/ / ftp.esat.kuleuven.be/ sista/ klemmens/ ViTraM/ Index.html

Step 1 After loading the module and expression data file into ViTraM, all modules are initially displayed according to their order in the input file, resulting in a non-optimal representation of the modules.

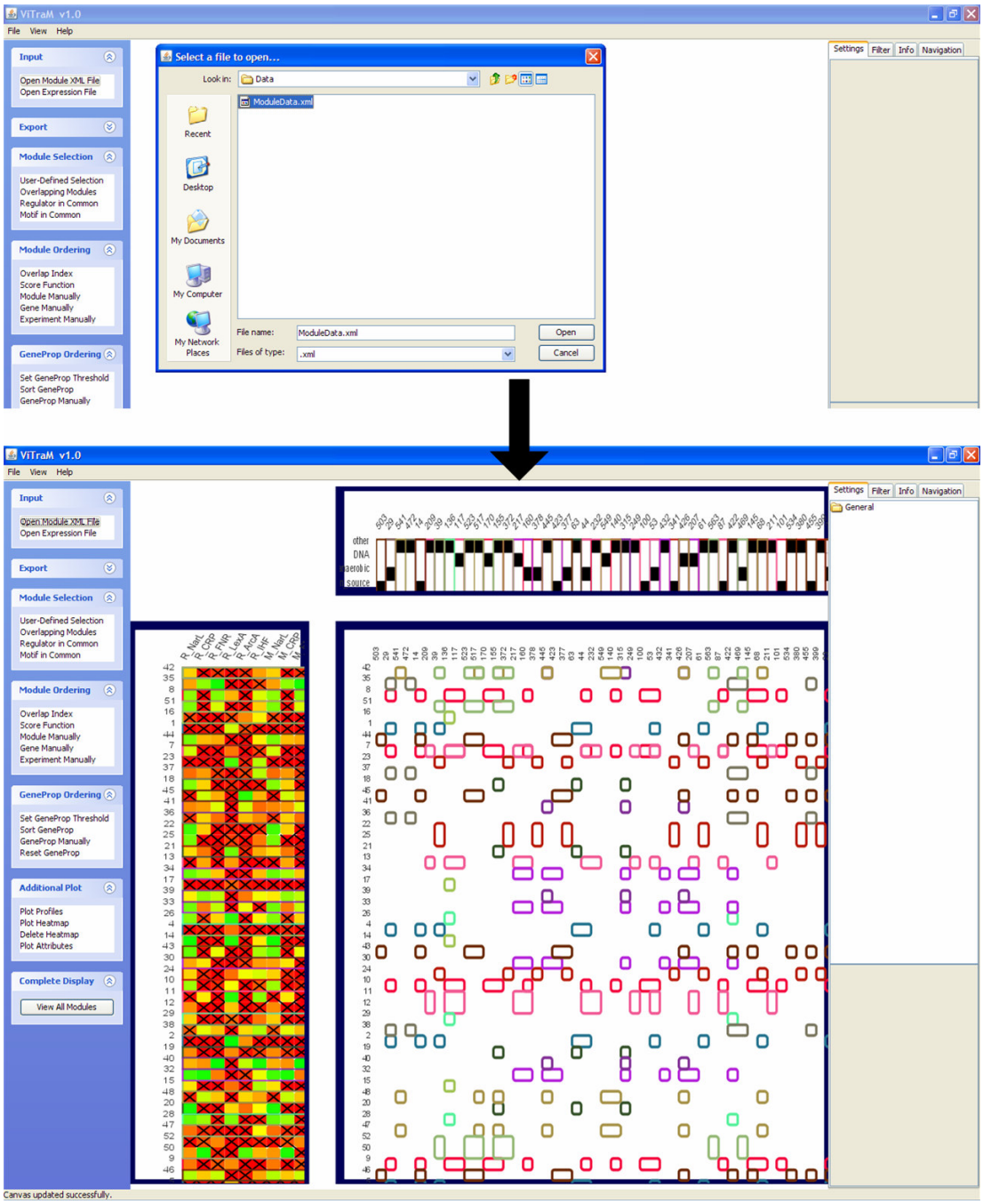

Figure 28: Loading the module file. Initially all modules are displayed according to their order in the module file.

Step 2 By using one of the ordering algorithms of ViTraM, for instance the overlap index, the optimal layout of the modules is determined, resulting in an improved layout. From this ordered ModuleImageDisplay, it is possible to see which modules show overlap in genes and/ or experiments with other modules.

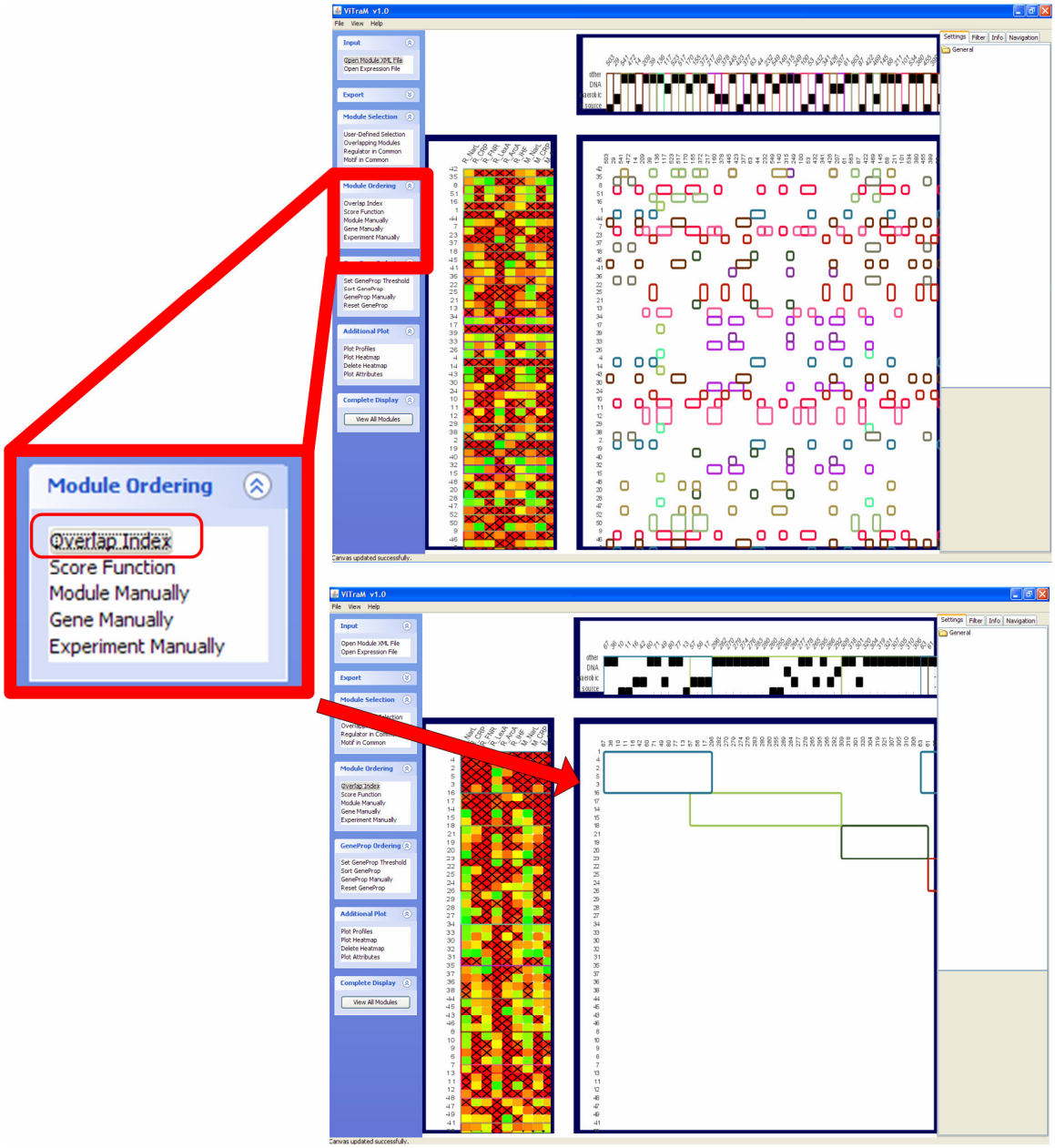

Figure 29: ViTraM includes several ways for obtaining an optimal layout of the modules. After using, for instance, the "Overlap Index" function the modules are ordered such that genes and experiments of a module are grouped together as much as possible, bringing into account overlap with other modules. Ordering the modules therefore allows for a more clear overview on the modules and makes the biological analysis of the modules more easy.

Step 3 The optimal layout of the modules can also be viewed by means of the OverviewDisplay. The layout before and after ordering is shown in Figure 30.

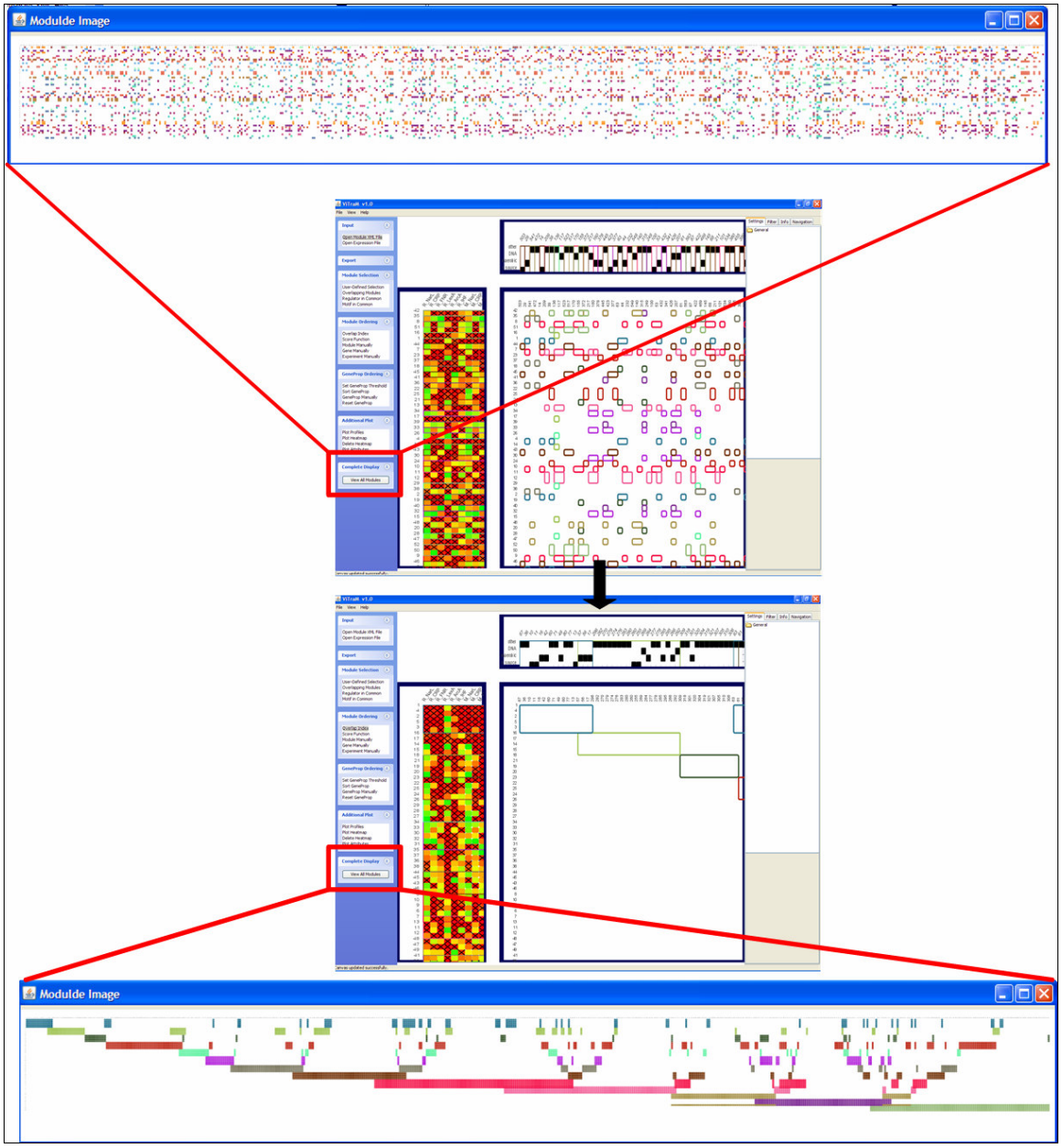

Figure 30: The layout of the modules can be viewed by using the OverviewDisplay. The layout before and after ordering of the modules can as such be displayed.

Step 4 In addition to viewing all modules, ViTraM also has several options for selecting modules the user wants to display. These functionalities will always operate on all modules in the module file. ViTraM allows selecting modules based on a common regulator or motif, such as for instance CRP. It is possible to select all overlapping modules of a particular module of interest. A user-defined selection is also possible: manually selecting modules 9 and 20 for instance will give the same result as selecting modules that have the CRP motif in common.



Figure 31: A subset of modules can be selected in several ways. Selecting all modules with a common motif M_CRP, for instance, results in the selection of two modules out of 13. Only these two modules will be displayed.

Step 5 After selecting modules, an optimal layout can again be obtained by ordering the modules by using the overlap index.

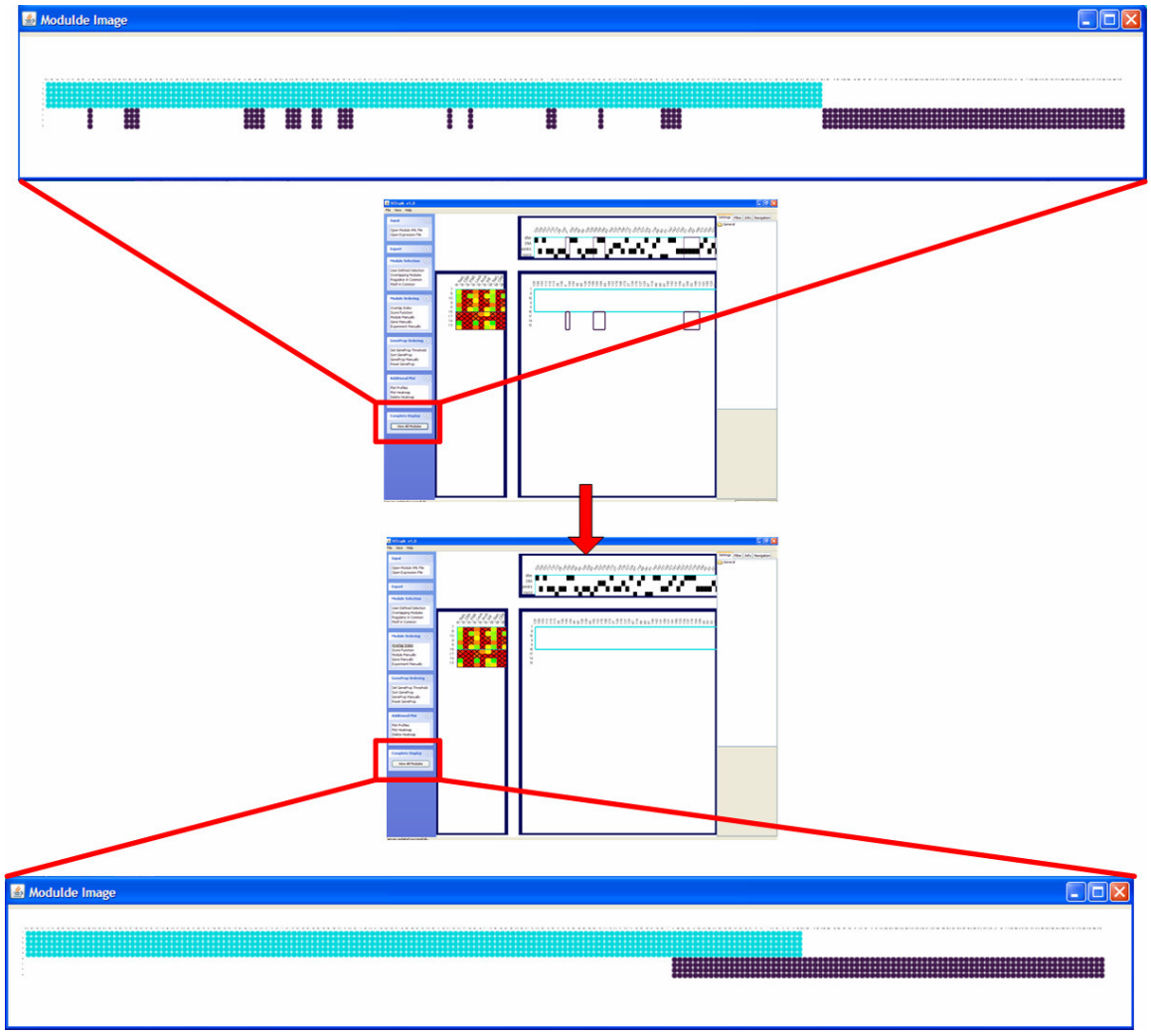

Figure 32: The optimal layout for the selected modules can be determined by using one of ViTraM's ordering techniques, for instance *"Overlap Index"*. As can be seen from the Overview Display, the genes and conditions are indeed grouped together such that no module is split up anymore.

Step 6 In the GenePropsImageDisplay, the scores of the regulators and motifs for the genes in the CRP modules are shown. The scores that satisfy the default thresholds are indicated by a cross. Changing the default score settings is possible by clicking "Set GeneProp Threshold" in the "GeneProp Ordering" panel. We changed the default minimum score for the motifs into 0.999 instead of 0.8.

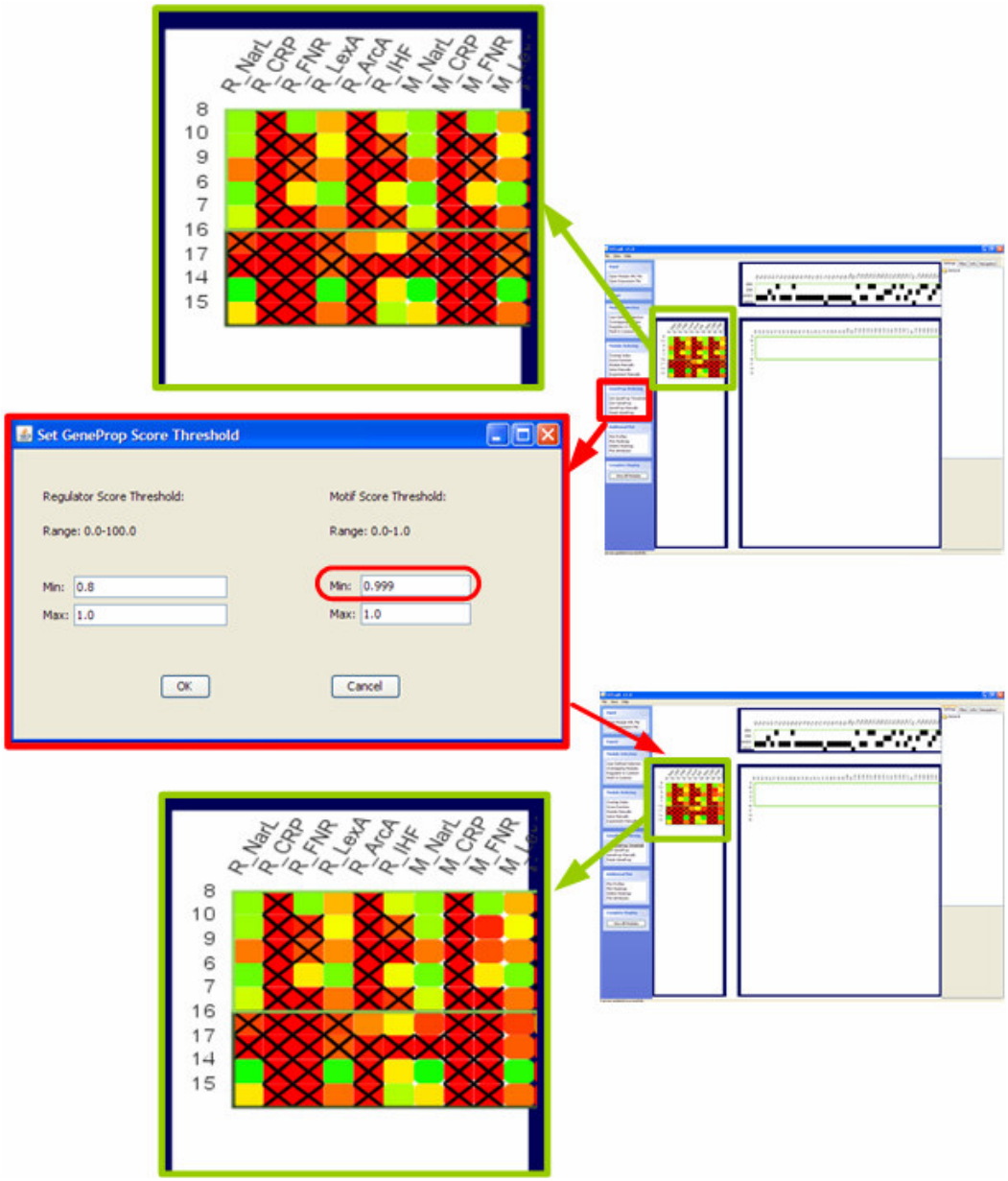

Figure 33: The gene properties are shown in the GenePropsImageDisplay. Scores for regulators and motifs are represented by a color. In red are the highest values, in green are the lowest values. A cross indicates that a particular score is in between the score thresholds. The minimum score threshold is default set to 0.8, while the default maximum score is 1. These thresholds can be changed by using the "Set GeneProp Threshold" button. In this example, we changed the minimum motif score to 0.999.

Step 7 When sorting the motifs in this display according to the scores, it is clear that in addition to the CRP motif, also the ArcA motif is important for one module, as this motif has high scores for all genes in this module. On the other hand, the FNR motif has high scores for the genes in the second module.

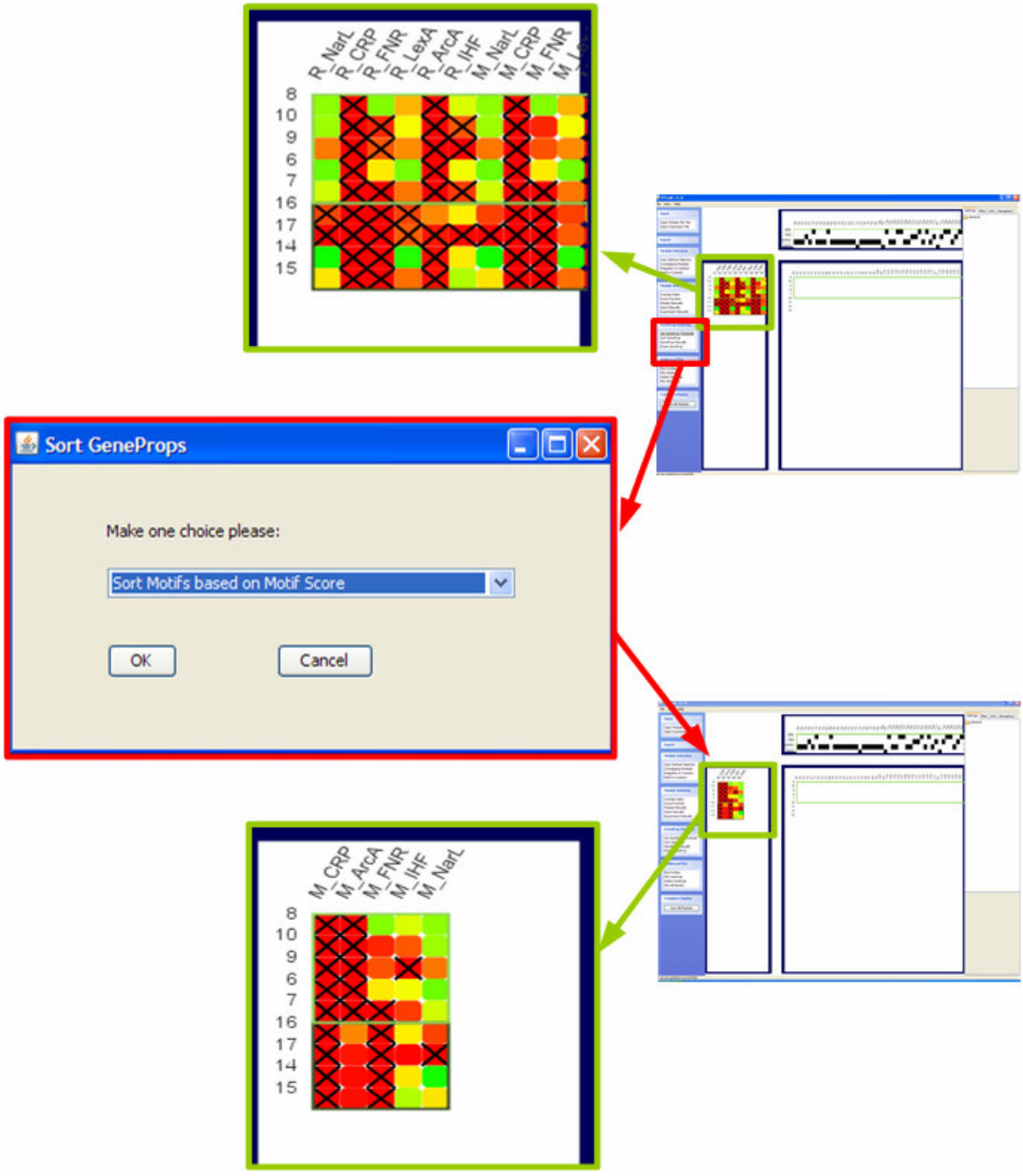

Figure 34: ViTraM includes several options for ordering the gene properties. In this example, the motifs were ranked according to their score for the genes in the currently displayed modules. Motif M_CRP is the motif with the best scores, followed by motifs M_ArcA and M_FNR.

Step 8 ViTraM includes a filtering of modules, genes and experiments. The filtering is always performed on the currently displayed modules. It is thus possible to apply several filtering steps sequentially. Modules can for instance be filtered, based on a minimum/ maximum of genes. This means that only modules that have a number of genes in the user-defined range will be shown.

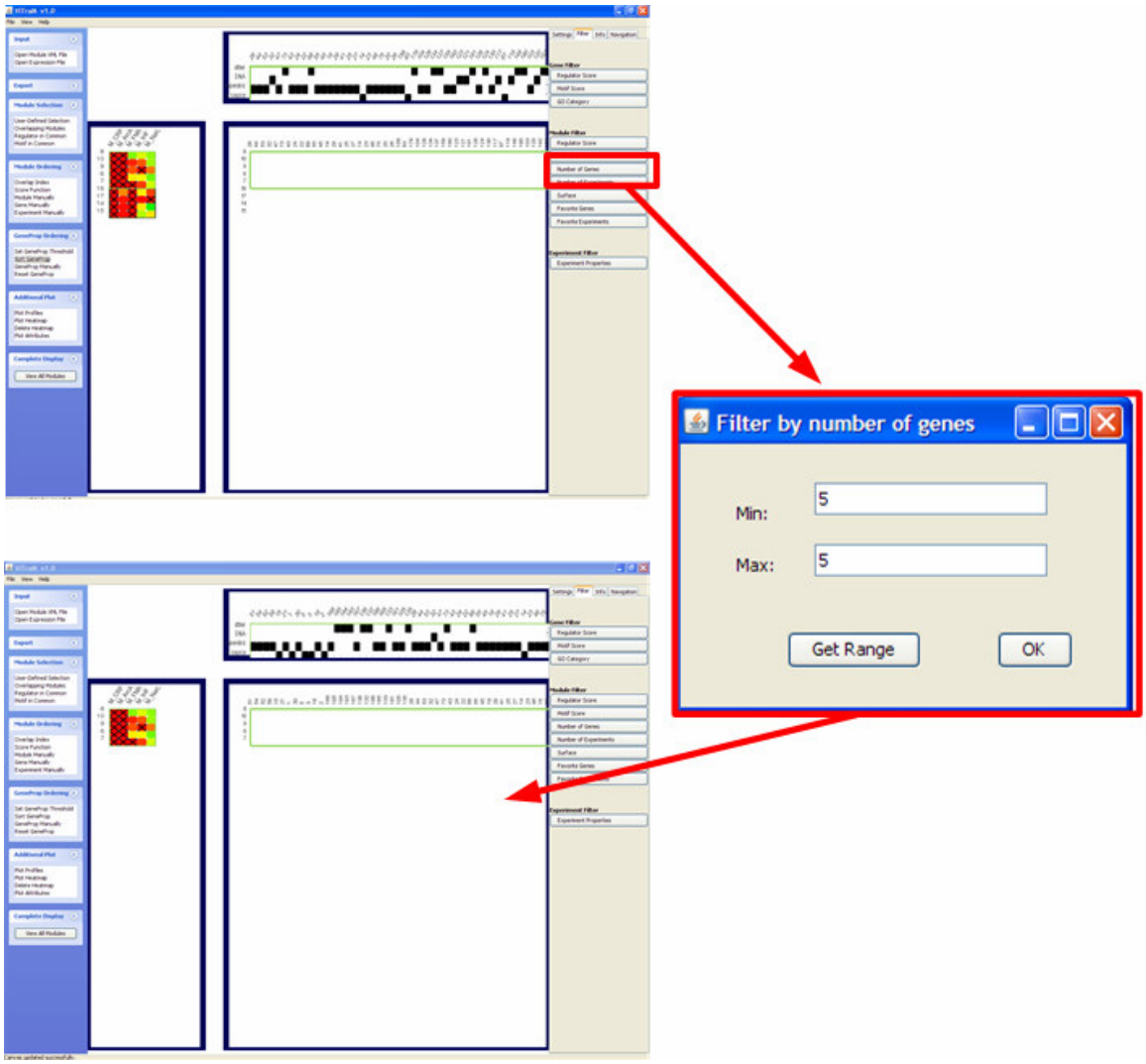

Figure 35: Filtering the modules by the number of genes they contain is possible. When choosing, for instance, modules with exactly 5 genes, only one module of the two satisfies this criterion and only that module is displayed.

Step 9 The ExpPropsImageDisplay shows that many experiments in which the genes of this module were co-expressed belong to either the carbon_source category or the anaerobic_aerobic category. Filtering the experiments by carbon_source reduces the number of experiments.

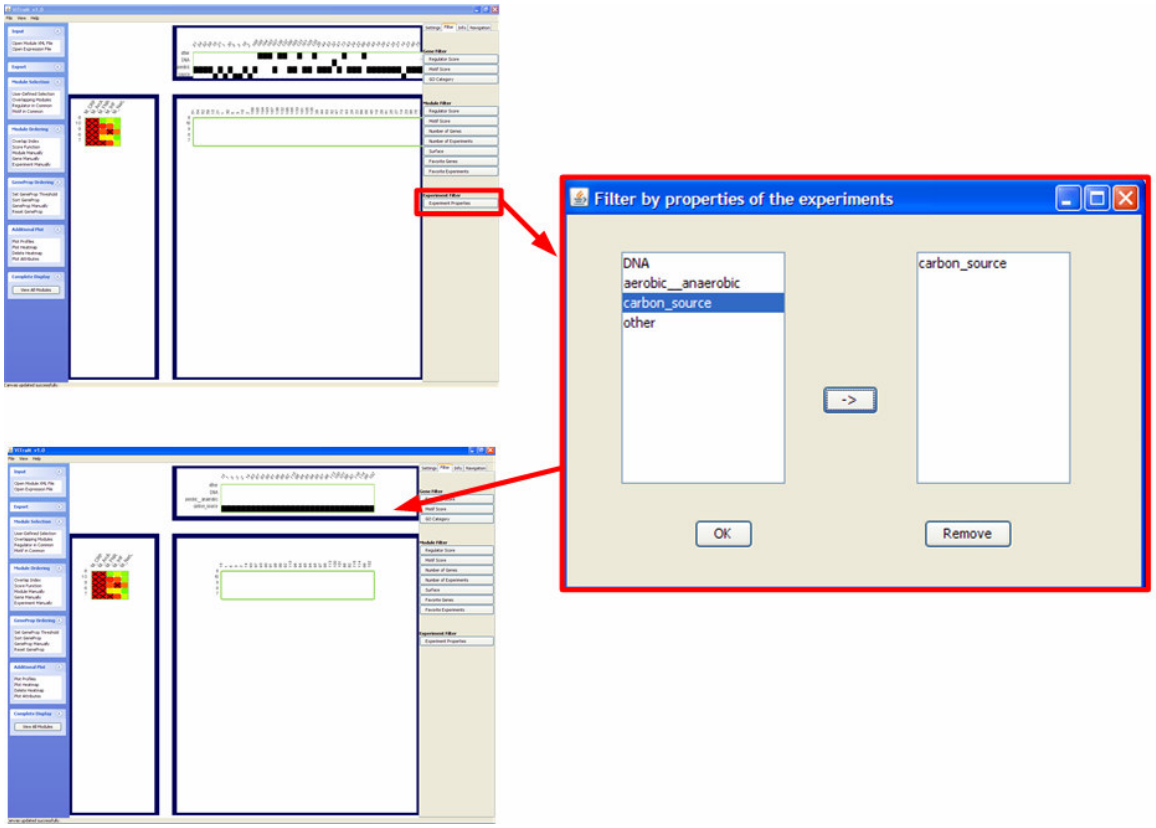

Figure 36: Several filtering procedures are included in ViTraM. Filtering the experiments by their properties is one of the options. If., for example, a filtering is done based on the property carbon_source, only those experiments that belong to this conditional class will be shown. This results in a reduced number of experiments that are displayed.

Step 10 The selected modules can be exported to a module XML file. This file can later on be imported again in ViTraM. The layout of the modules in the ModuleImageDisplay can also be exported as an SVG or PNG figure.

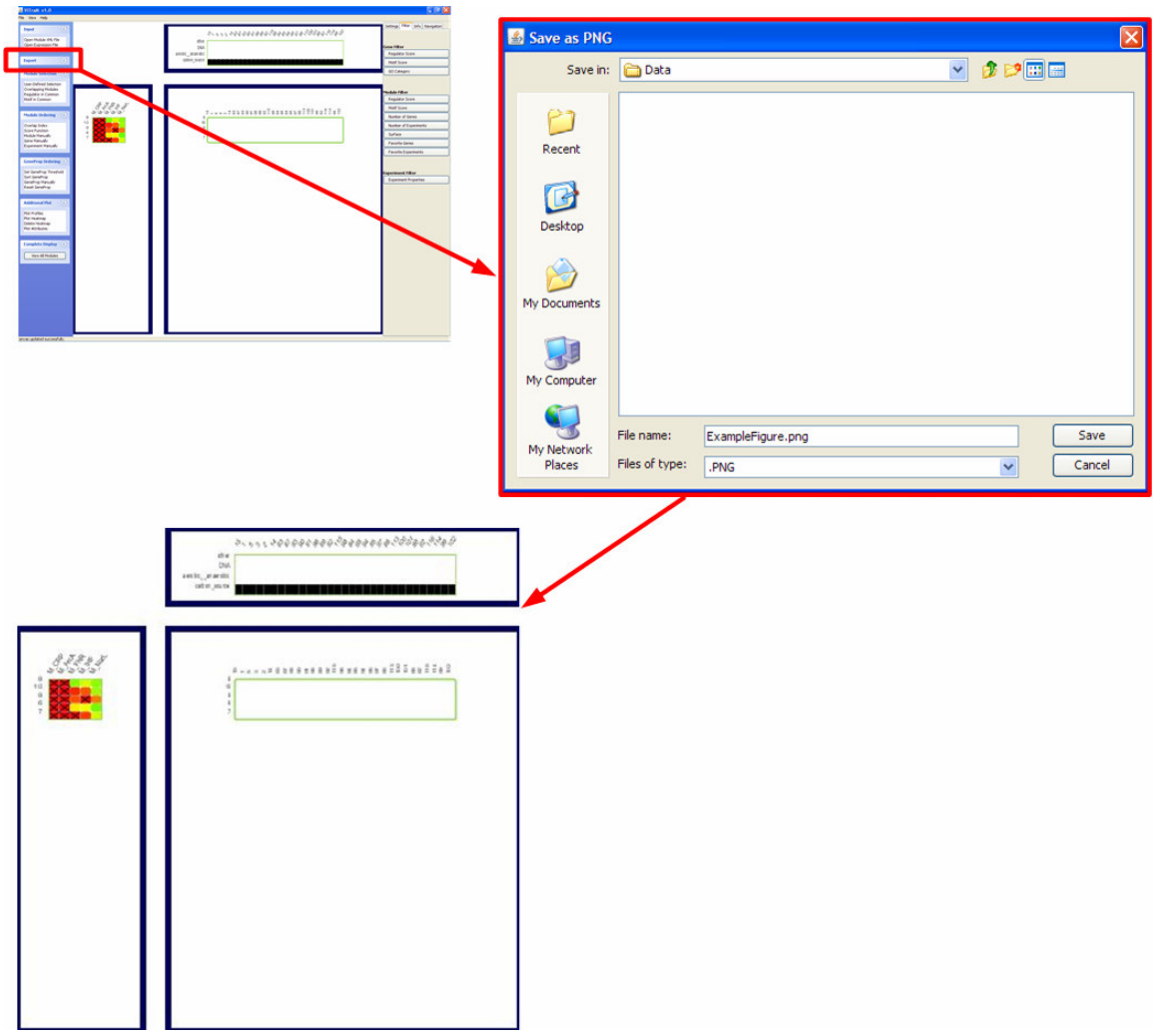

Figure 37: Saving the current display as a figure. The current display can be saved as either PNG or SVG format.

6 REFERENCES

1. Madeira SC, Oliveira AL. **Biclustering Algorithms for Biological Data Analysis: A Survey**. *IEEE Transactions on Computational Biology and Bioinformatics*. 2004. **1**: 24–45
2. Lemmens K, De Bie T, Dhollander T, De Keersmaecker S, Thijs I, Schoofs G, De Weerdt A, De Moor B, Vanderleyden J,Collado-Vides J, Engelen K, Marchal K**. Condition-dependent combinatorial regulation in Escherichia coli**. submitted 2009.