ViTraM: VIsualization of TRAnscriptional Modules

Version 2.0

October 1st, 2009

KULeuven, Belgium

Contents

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 (See Figure 1**Error! Reference source not found.**).
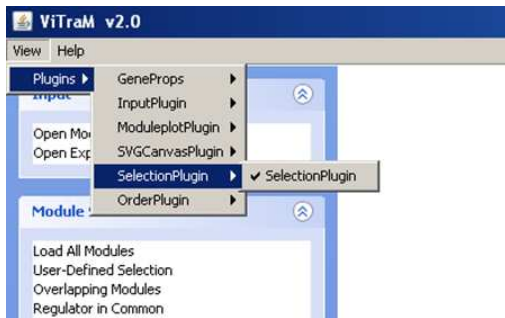

Figure 2: Plug-in structure of ViTraM. The different plug-ins can be opened and closed by using the *"View"* option.

1.3 Requirements

Developed in JAVA, ViTraM is platform-independent and 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.

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. Bioinformatics.* (2009) ViTraM: Visualization of Transcriptional Modules.

Also check our accompanying website for recent updates:

http://homes.esat.kuleuven.be/~kmarchal/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:

http://homes.esat.kuleuven.be/~kmarchal/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 operating system:

- Windows:
    o If you have a small data set, you can directly double click on the file ViTraM_Windows.bat to run the software.
    o 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 or Mac:
    o Run ViTraM in a terminal "./ViTraM_Linux&Mac.sh" to run the software on a smaller data set
    o Or run ViTraM in a terminal with command "java -jar – Xms256M – Xmx2G ViTraM.jar" if you want to assign more memory. .

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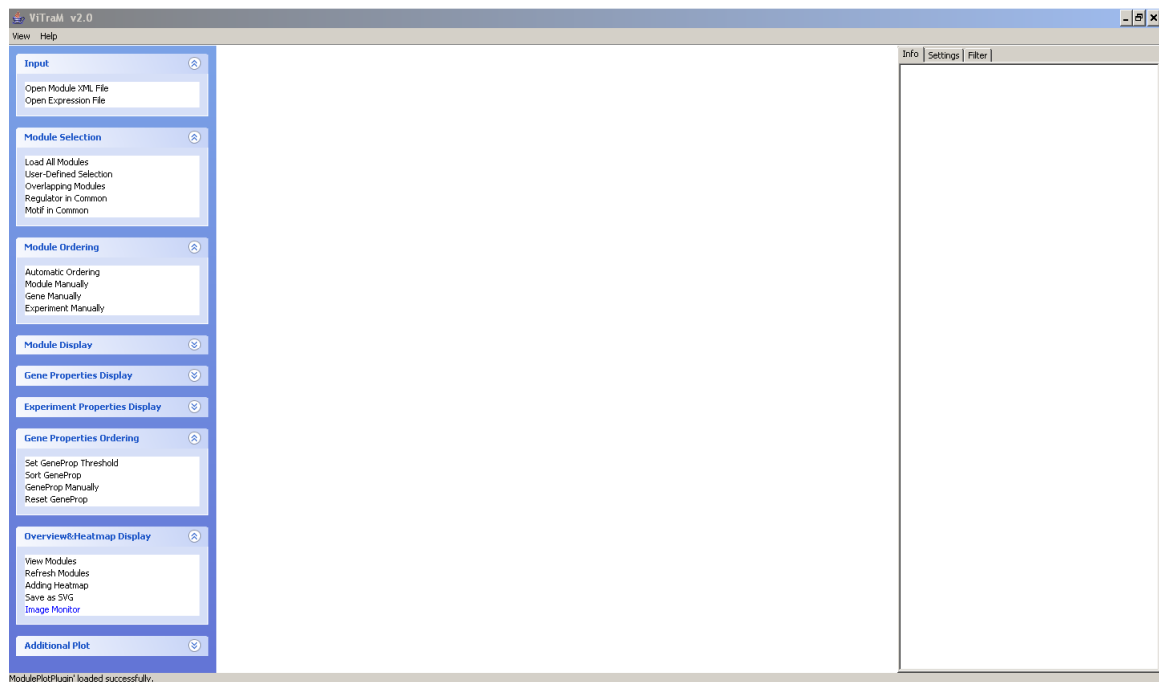

Figure 3: Start window of ViTraM. The different plug-ins "Input", "Module Selection", "Module ordering", "Module Display", "Gene Properties Display", "Experiment Properties Display", "Gene Properties Ordering", "Overview & Heatmap Display" and "Additional Plot" can be seen on the left panel. On the right panel, the "Info", "Settings" and "Filter" options are available.

2 INPUT FILES

ViTraM requires two input files:

- Module file (section 2.2)
- Expression data file (section 2.3)

2.1 XMLCreator

The XMLCreator will use the input and output data of DISTILLER [2], together with additional data to create the XML file and expression data file that are required for visualization by ViTraM. Although the XMLCreator currently only includes the possibility to derive the XML file from the output and input of the module detection tool DISTILLER [2], more algorithms will be included in the future.

You can download XMLCreator and find the helpfile on:

http://homes.esat.kuleuven.be/~kmarchal/ViTraM/Index.html.

2.2 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 2.3). The structure of the 'module file' is shown in**Error! Reference source not found.**. 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,…).

```xml
<?xml version="1.0" encoding="UTF-8"?>
<module_network>
  <xml_definition_version>0.1.2.1</xml_definition_version>
  <algorithm>DISTILLER</algorithm>
  <date>2008-11-13 17:22:21</date>
  <genes>
    <gene id="1" name="gene_name">
      <property id="R1">regulator_score</property>
      <property id="M1">motif_score</property>
      <property id="GO_category1">0</property>
      <property id="GO_category2">1</property>
      ...
    </gene>
    ...
  </genes>
  <experiments>
    <experiment id="1" name="exp_name">
      <property id="category1">1</property>
      <property id="category2">0</property>
      ...
    </experiment>
    ...
  </experiments>
  <regulators>
    <regulator id="R1" name="Regulator 1" />
    ...
  </regulators>
  <motifs>
    <motif id="M1" name="Motif 1" />
    ...
  </motifs>
  <gos>
    <go id="A" name="GO 1" />
    ...
  </gos>
  <modules>
    <module id="Module_1">
      <genes>
        <gene id="1" />
        <gene id="5" />
        ...
      </genes>
      <experiments>
        <experiment id="12" />
        <experiment id="541" />
        ...
      </experiments>
      <regulators>
        <regulator id="M1" />
      </regulators>
      <motifs>
        <motif id="M1" />
      </motifs>
    </module>
    ...
  </modules>
</module_network>
```

General information · Gene property information · Experiment property information · Gene properties list · Gene-experiment property information

Figure 4: 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.

8

- 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.

2.3 Expression data file

The microarray expression data should be loaded in case the user wants to plot the modules' expression profiles or heatmaps. The expression file is a tab-delimited file. Gene names (ids) or experiment names (ids) are mentioned in the file in the first column and first row, respectively.

2.4   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 (See Figure 5**Error! Reference source not found.**) will appear and the correct module or expression file can be selected and opened. If the module file is loaded, a message will be shown in the left bottom corner ("Expression Data loaded successfully") and will indicate when the expression file is loaded.

Initially, no data will be shown. The data can be visualized by first clicking on *"Load All Modules"* in the *"Module selection"* plug-in. Subsequently clicking on *"View Modules"*, *"View gene properties"* and *"View experiment properties"* in the *"Module Display"*, *"Gene Properties Display"* and *"Experiment Properties Display"* plug-in respectively will open three windows that show the data that was loaded. In section 3 we will further explain these three windows.
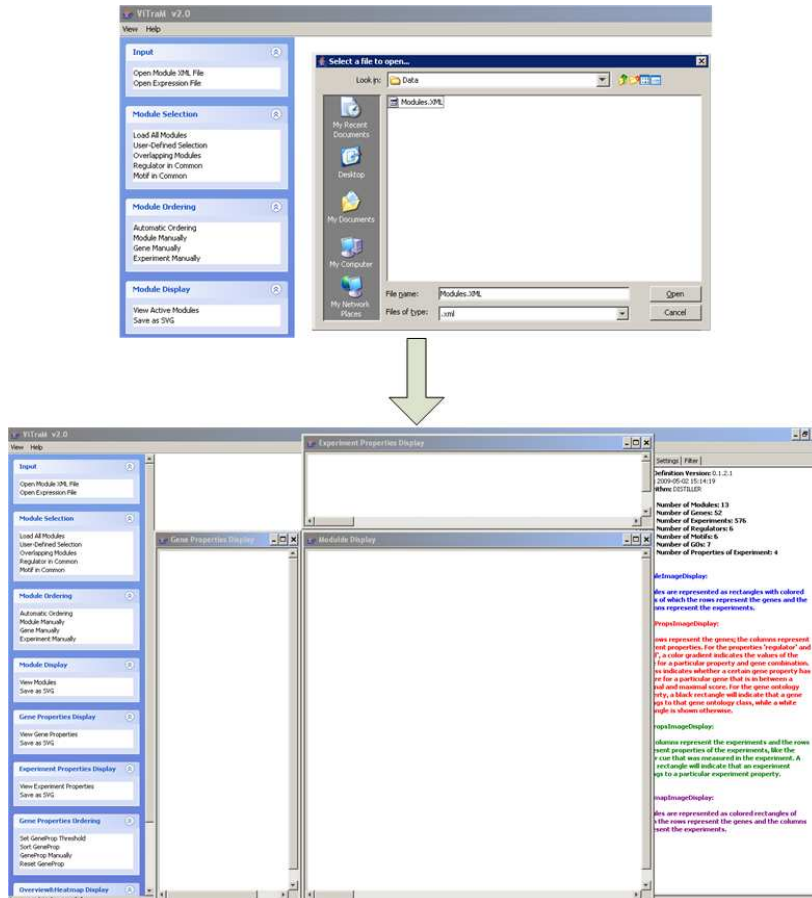

Figure 6: By clicking on "Open module XML file" in the "Input" panel, a pop-up window will appear. The module file will be loaded in ViTraM. Clicking on "View Modules" in the Module Display Panel, "View Gene Properties" in the Gene Properties Display Panel and "View Experiment Properties" in the Experiment Properties Display Panel, three windows will be displayed. Initially no data will be shown. Data can be loaded by selecting (all) modules from the "Module Selection" panel.

3 SOFTWARE ViTraM

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 ModuleDisplay. In the ModuleDisplay, 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). Initially the ModuleDisplay will not be opened. Clicking on "View Modules" in the Module Display Panel will open the ModuleDisplay. The modules that should be visualized can be selected via the Module Selection or Filter panel (see also section 3.4 and 3.3). 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 ModuleDisplay. A more optimal way of representing the modules can be obtained by ordering the modules (see section 3.5).

Displaying the additional information

In addition to the ModuleDisplay, two other displays are shown. The GenePropertiesDisplay and ExperimentPropertiesDisplay provide information on properties of the genes and the experiments, respectively (as extracted from the module input file) (see Figure 7). Clicking on "View Gene Properties" in the Gene Properties Display Panel and "View Experiment Properties" in the Experiment Properties Display Panel will open the GenePropertiesDisplay and ExperimentPropertiesDisplay respectively. The modules for which the properties should be visualized can be selected via the Module Selection or Filter panel (see also section 3.4 and 3.3).

The rows of the GenePropertiesDisplay 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 GenePropertiesDisplay 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 3.6.1). 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 ExperimentPropertiesDisplay, 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 ModuleDisplay.
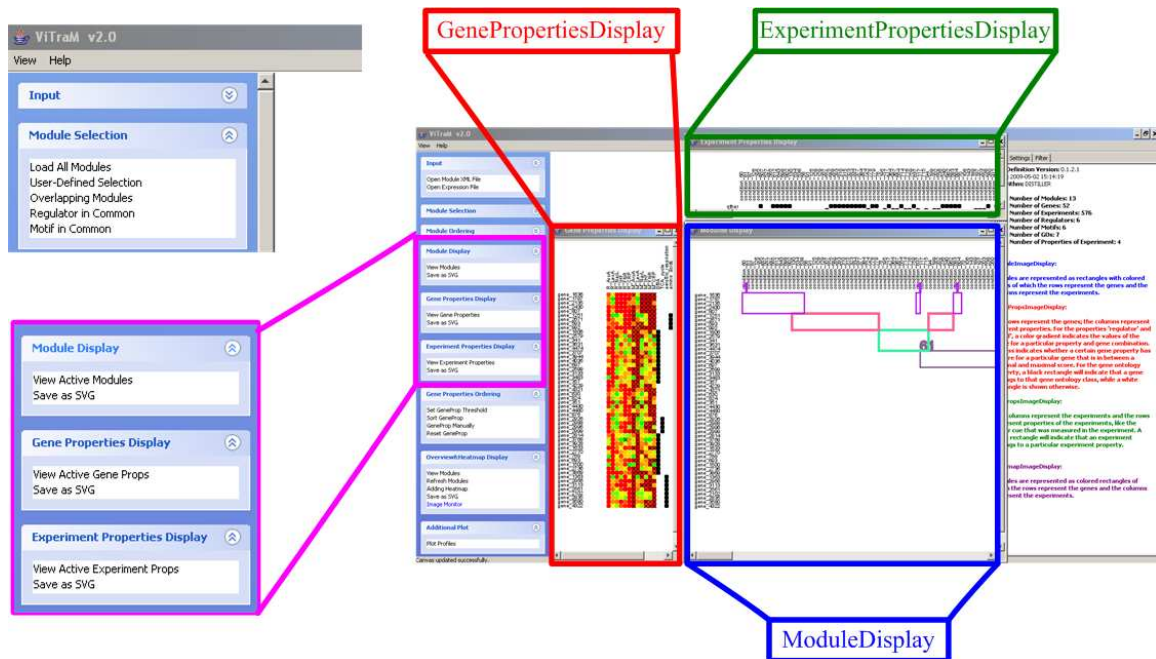

Figure 8: The three image displays are shown: the GenePropertiesDisplay, the ExperimentPropertiesDisplay and the ModuleDisplay. The GenePropertiesDisplay represents the genes in the rows, and the gene properties in the columns. Three gene properties are implemented at the moment: regulator, motif and 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 ExperimentPropertiesDisplay 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 ModuleDisplay represents the genes in the rows and the experiments in the columns. In the ModuleDisplay the modules, consisting of genes and experiments, are represented by transparent rectangles of different colors. The gene and experiment order in the GenePropertiesDisplay and ExperimentPropertiesDisplay are according to the order of the genes and experiments in the ModuleDisplay and the three displays are dynamically linked.

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 three sub-panels:

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

The panel on the left side consists of nine plugins:

- Module Display
- Gene Properties Display
- Experiment Properties Display
- Input (section 2)
- Module Selection (section 3.4)
- Module Ordering (section 3.5)
- Gene Properties Ordering (section 3.6)
- Overview & Heatmap Display (section 3.7)
- Additional Plot (section 3.8)

We will discuss these functionalities into more detail below.

3.1 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 (See Figure 9). In addition, the information that is shown by each of the three displays is explained in the panel.
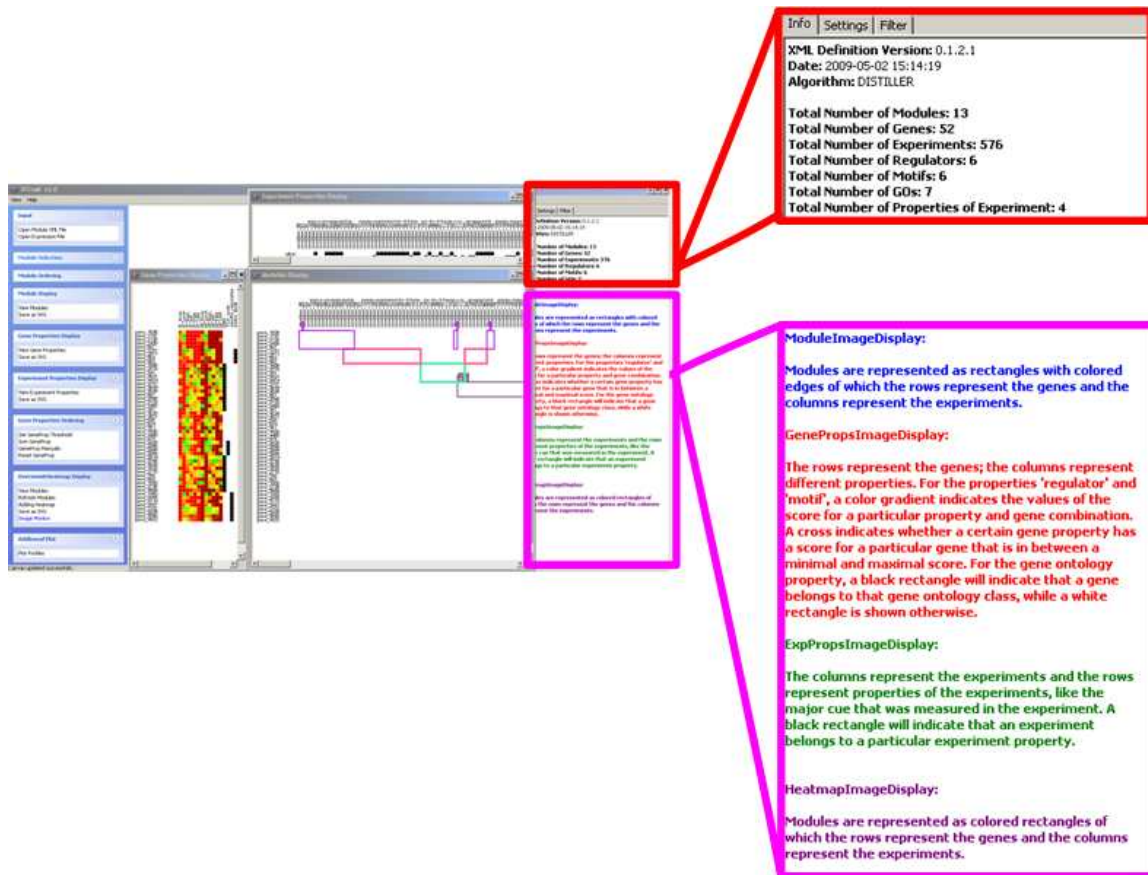

Figure 10: The info panel shows general information of the modules present in the module file (red). In addition, the info panel also gives information on what is visualized on the GenePropertiesDisplay, ModuleDisplay and ExperimentPropertiesDisplay (pink).

3.2   Settings panel

More specific information about each module that is selected and currently displayed in the ModuleDisplay can be obtained from the *"Settings panel"*. A list of currently displayed modules is available (see **Error! Reference source not found.**Figure 11). 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.
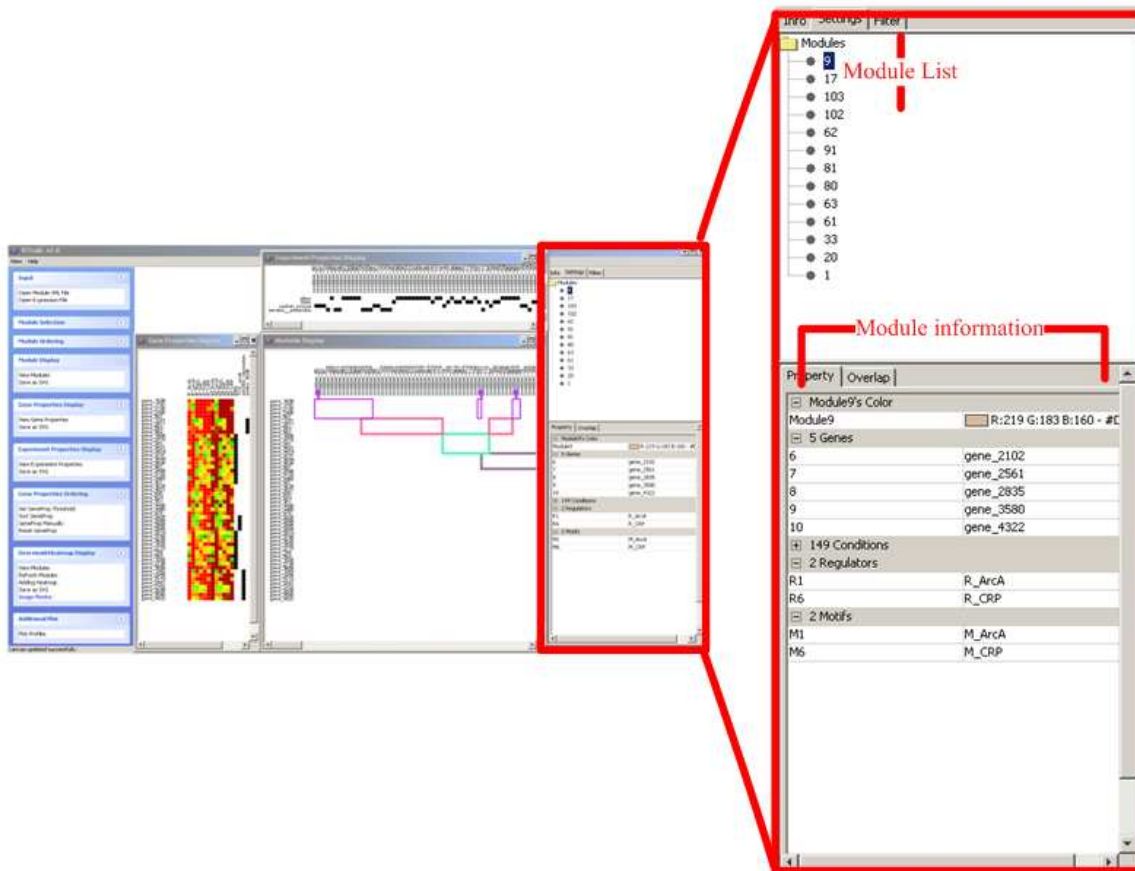

Figure 12: Information available from the *"Settings"* panel. The Settings panel contains information on the modules that are currently displayed in the ModuleDisplay. For each module separately, information is provided on its genes, experiments, motifs or regulators.

3.3 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 ModuleDisplay. It is thus possible to filter sequentially based on several criteria (in contrast to section 3.4).

3.3.1 Gene filter

Genes can be filtered based on their properties such as (see Figure 13):

- 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 (GenePropertiesDisplay, ExperimentPropertiesDisplay and ModuleDisplay). 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.

3.3.2 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 <u>all</u> 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 ModuleDisplay.
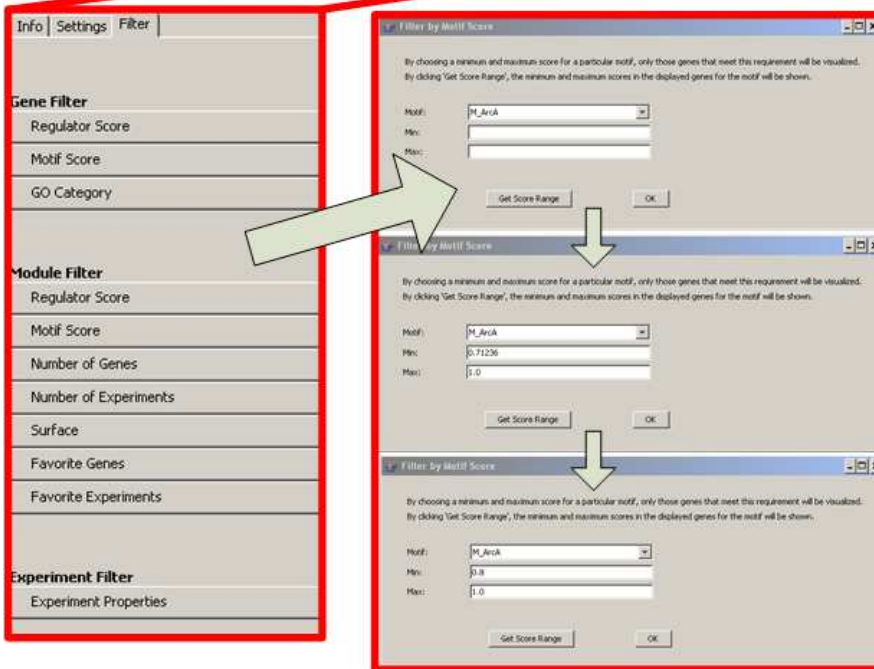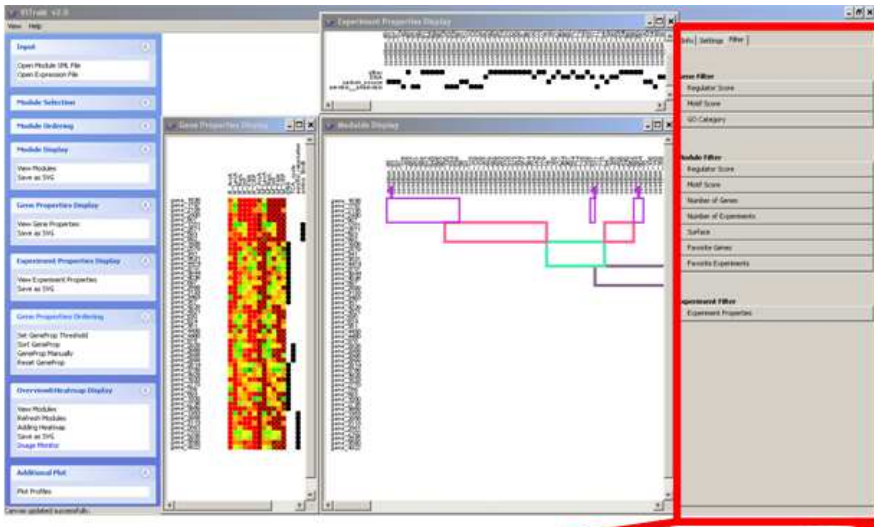
Figure 14: Modules, genes and conditions can be filtered by using the "Filter" panel of ViTraM. The filtering functions always operate on the currently displayed modules. Genes can be filtered based on their regulator score, motif score or whether they belong to a functional class. Modules can also be 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_ArcA. 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_ArcA 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 **Error! Reference source not found.**Figure 15). 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.
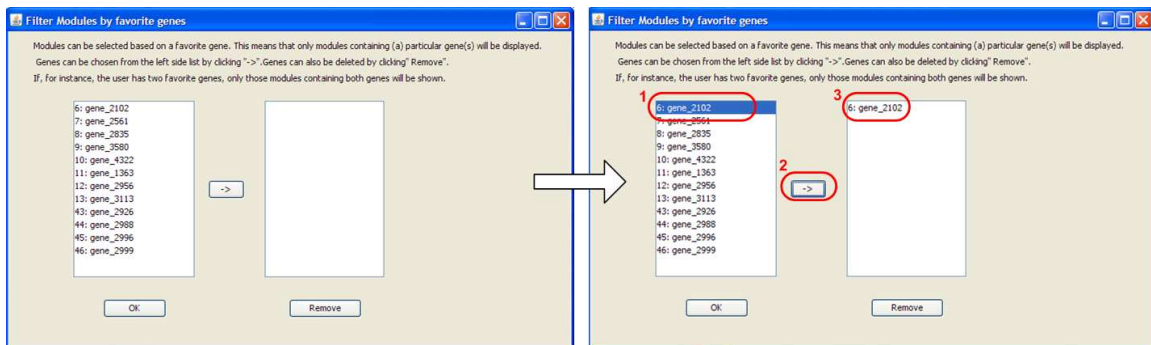

Figure 16: Example of how to filter modules containing a particular gene. The list on the left contains all genes (geneId: geneName). By selecting a gene (1) and clicking on the *"->"* button (2), a gene can be added in the list on the right (3). When clicking on *"OK"*, only modules containing the gene(s) 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.

3.3.3 Experiments filter:

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

3.4 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 <u>all</u> modules that are present in the module file (see Figure 17**Error! Reference source not found.**):

- Load all modules (section 3.4.1)
- User-defined selection (section 3.4.2)
- Overlapping modules (section 3.4.3)
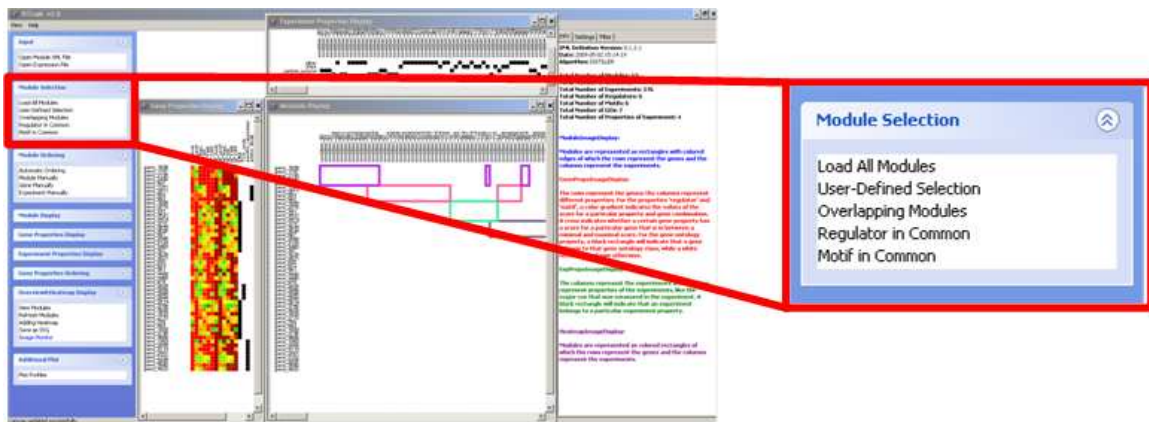- Motif or regulator in common (section 3.4.4)

Figure 18: The different possibilities for selecting a subset of modules.

3.4.1 Load all modules

After applying filtering (see section 3.3) and selection strategies (see section 3.4) on the modules, the user might want to obtain the original set of modules. This set of modules can be obtained by clicking on "Load All Modules" (see Figure 19**Error! Reference source not found.**).
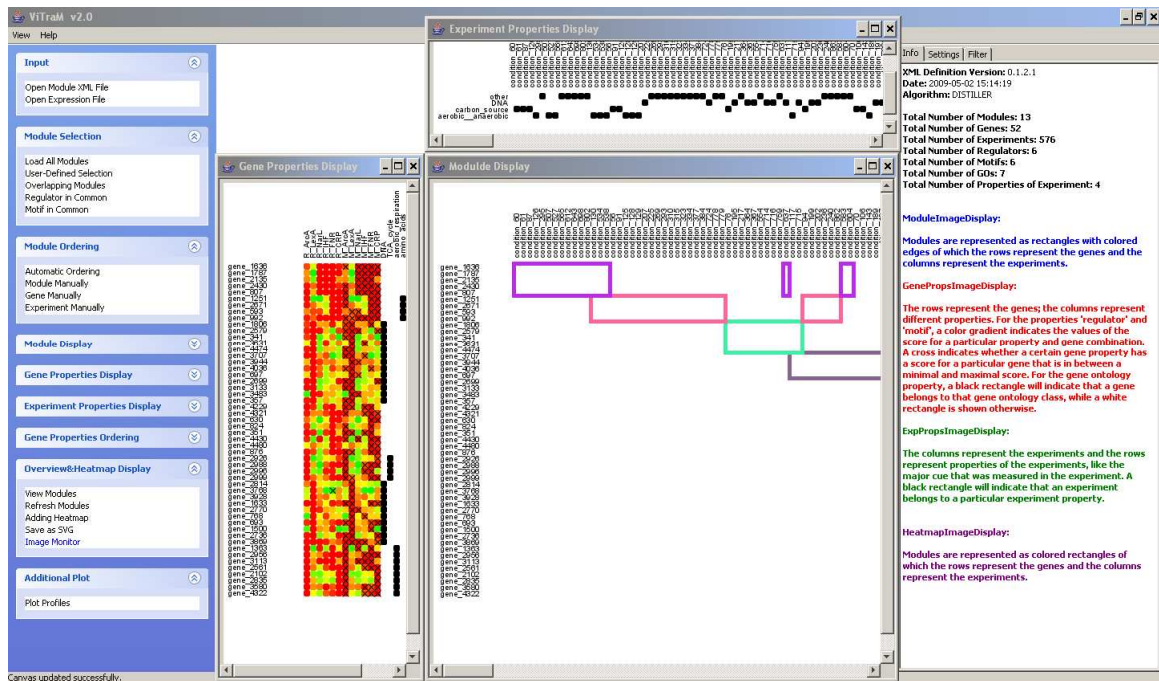

Figure 20: The "Load all Modules" function allows obtaining the original set of modules.

3.4.2 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 ModuleDisplay will then be updated and only the selected modules are shown (see Figure 21**Error! Reference source not found.**).
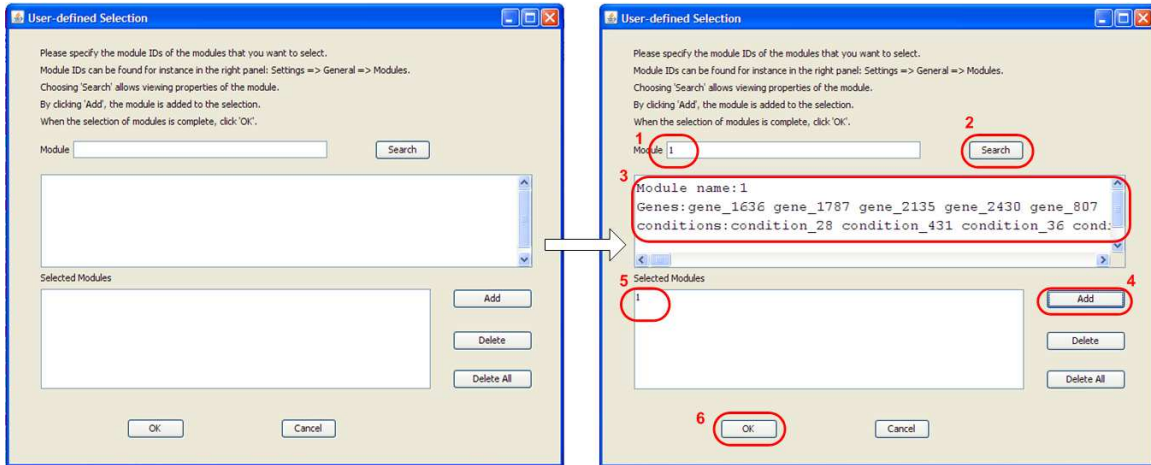
Figure 22: Pop-up window that appears when a user-defined selection is chosen. First type the name of the module (1). By clicking the *"Search"* button (2), the module's information will be shown in the box field (3). The module can be added by pushing the *"Add"* button (4 and 5). If you want to delete 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 ModuleDisplay will be updated according to the module selection after clicking the *"OK"* button (6).

3.4.3 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 23**Error! Reference source not found.**). 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 GenePropertiesDisplay, ExperimentPropertiesDisplay and ModuleDisplay will be updated with this new selection.
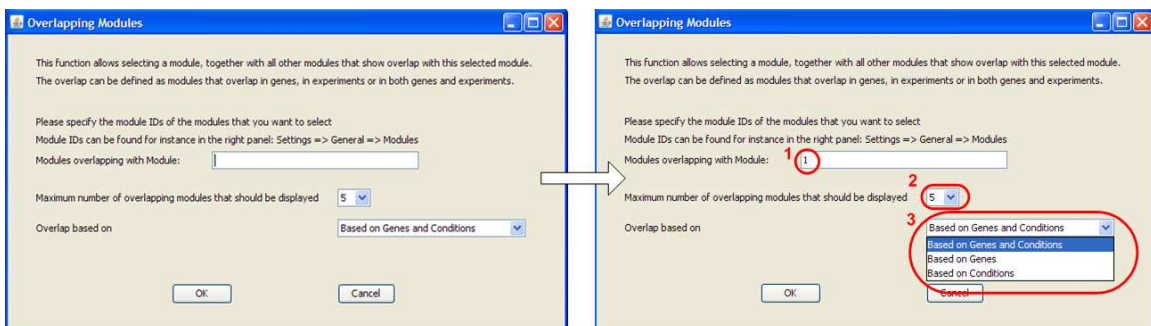

Figure 24: Selection of modules by *"Overlapping Modules"* function. After loading the data, modules of interest can be selected based on their overlap with a particular module of interest (1). The overlap can be based on genes, experiments or both genes and experiments (3). The modules that overlap 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).

3.4.4   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 25**Error! Reference source not found.**). 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 GenePropertiesDisplay, ExperimentPropertiesDisplay and ModuleDisplay 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.
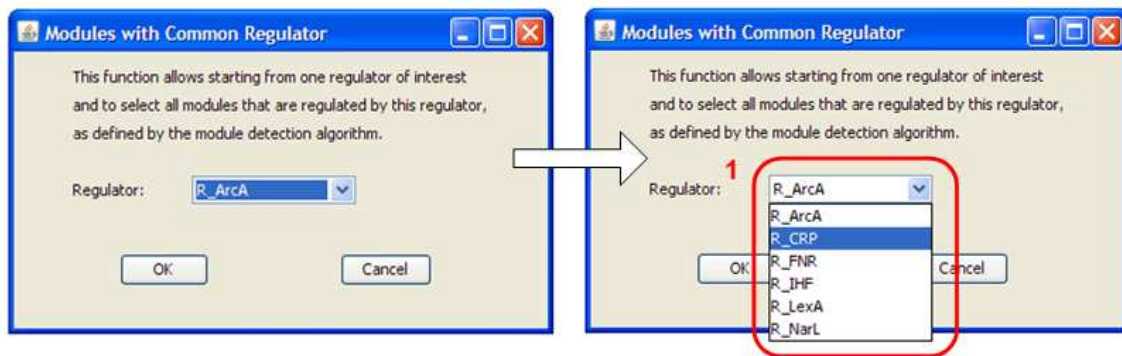

Figure 26: Selecting modules by using the "Regulator in common" function. A pop-up window will be shown after clicking the *"Regulator in Common"* button in the *"Module Selection"* panel. In this pop-up window the motif of interest can be chosen (1). After clicking the *"OK"* button, only the modules that have this motif will be visualized.

3.5   Optimizing the layout in the ModuleDisplay

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 (See Figure 15**Error! Reference source not found.**). Several reordering options are included:

- Automatic ordering
  - Overlap index (section 3.5.1.1)
  - Score function (section 3.5.1.2)

- User-defined ordering (section 3.5.2)
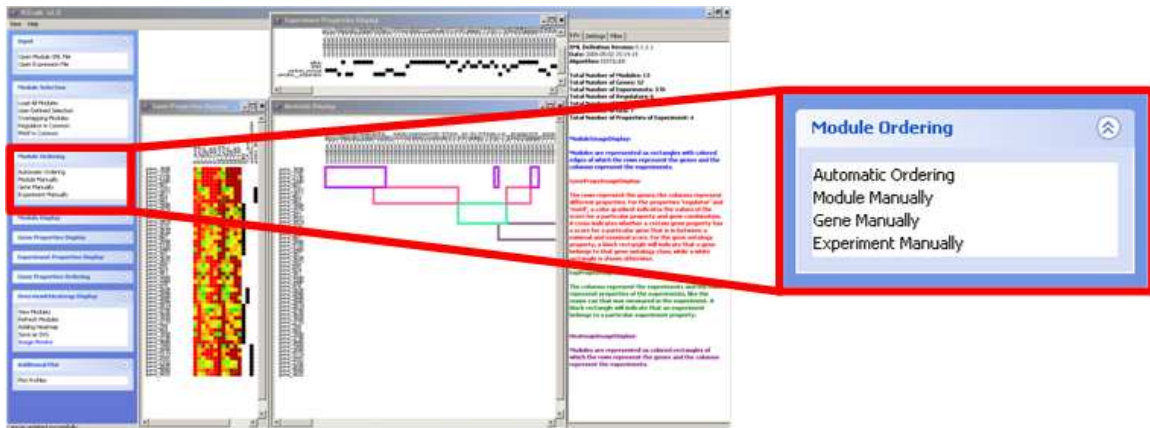 - Modules
  - Genes
  - Experiments

Figure 15: The different possibilities for sorting the modules.

3.5.1 Automatic ordering

3.5.1.1 Overlap Index

The overlap index method is an iterative procedure that places the modules that overlap with most other modules first in the ModuleDisplay. 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 ModuleDisplay. 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 ModuleDisplay, 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 3.5.4).

3.5.1.2 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

23

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 ModuleDisplay 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 ModuleDisplay 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 3.5.4).

3.5.2 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**Error! Reference source not found.**).

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 ModuleDisplay. 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 3.5.3). Changing the order of the genes and experiments can be done analogously.
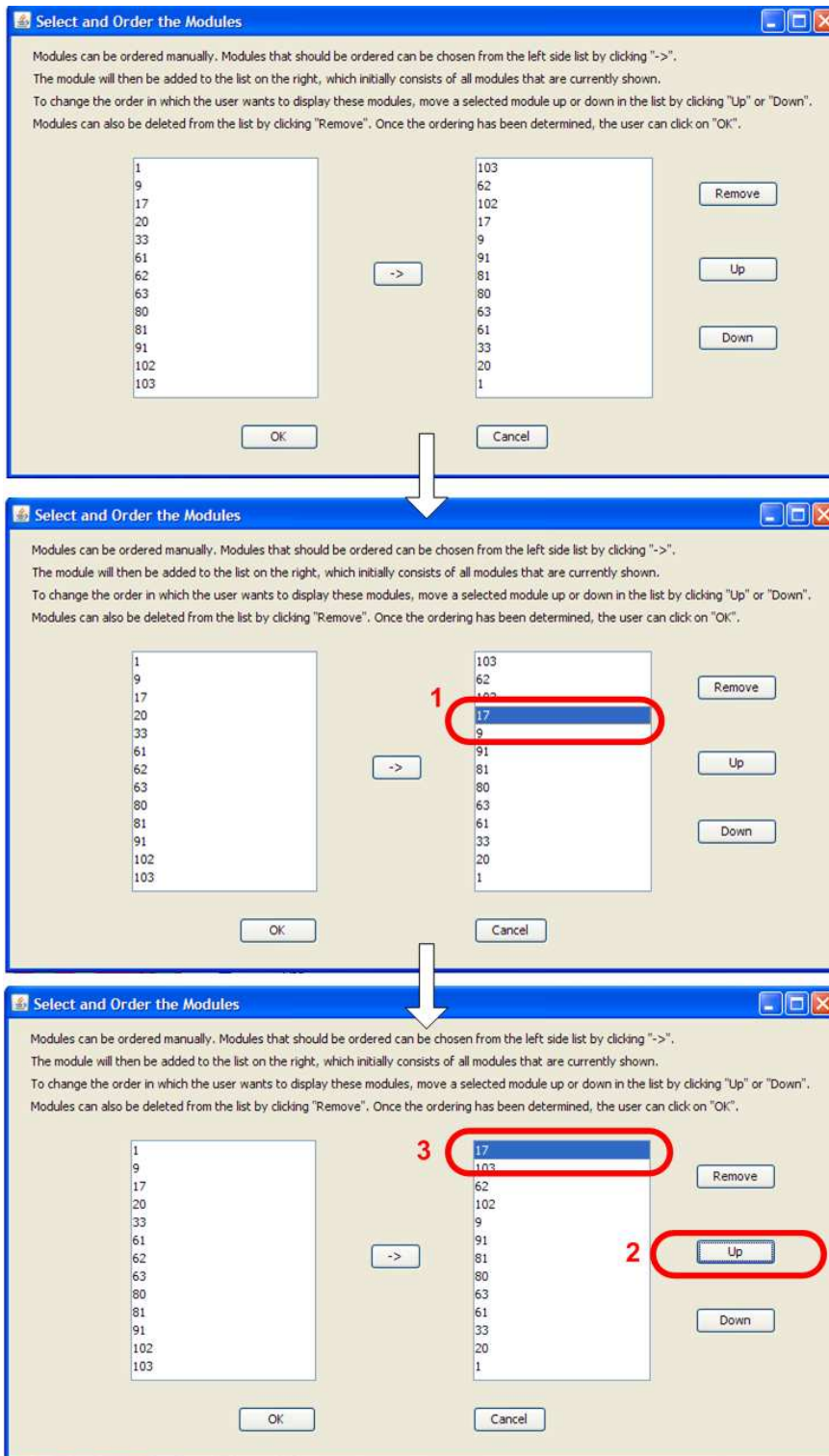
Figure 16: Pop-up window that appears when manually changing the order of the modules. Modules that should be ordered can be chosen from the left side list by clicking *"->"*. The module will then be added to the list on the right, which initially consists of all modules that are currently shown in the ModuleDisplay. 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"*.

3.5.3   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.
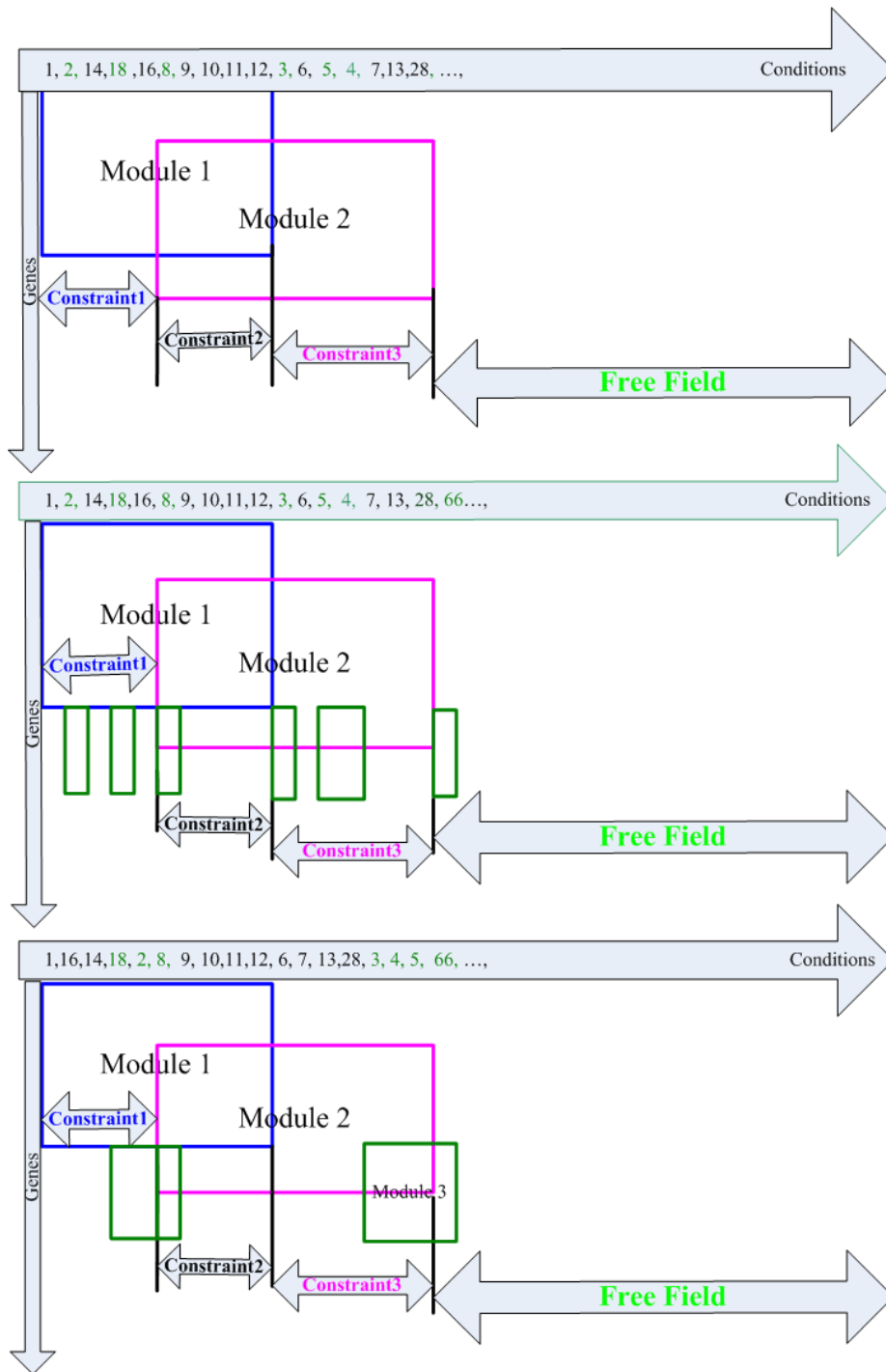
Figure 17: The "Constraint reordering of genes/experiments". Panel a; Module1 and Module2 are placed in an optimal way. Panel b: Adding module 3 using the current order of conditions will lead to a high fragmentation of module 3. Panel c: the fragmentation of module 3 can be reduced by reordering the 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.

3.6 Optimize layout in the GenePropertiesDisplay

The user can not only manipulate the ModuleDisplay, but also the GenePropertiesDisplay. 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**Error! Reference source not found.**):

- Choosing the regulator/motif score threshold (section 3.6.1)
- Sorting the regulators/motifs based on:
    - o Their scores (section 3.6.2)
    - o Their assignment to modules by the used module detection algorithm (section 3.6.3)
- User-defined sorting of the regulators/motifs (section 3.6.4)
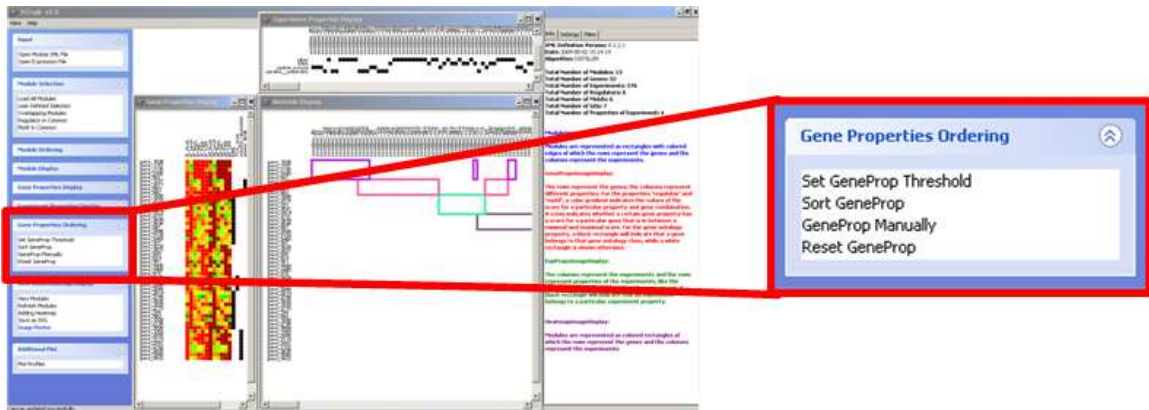- Resetting the ordering of the gene properties (section 3.6.5)

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

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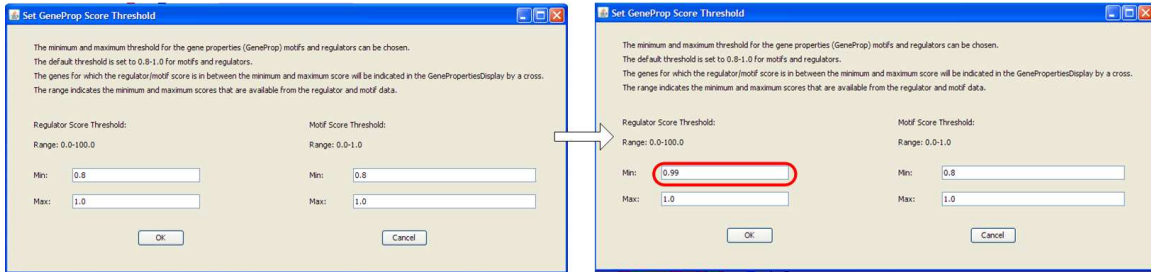
Figure 19: Pop-up window for setting the score thresholds of the gene properties. The ranges of the scores for regulators and motifs are shown. The default score will be shown in the text field when opening this window. A new minimum and/or maximum score can be chosen for both regulators and motifs.

3.6.2 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 GenePropertiesDisplay can be ordered based on their scores for the genes in the modules currently displayed in the ModuleDisplay. 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):

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

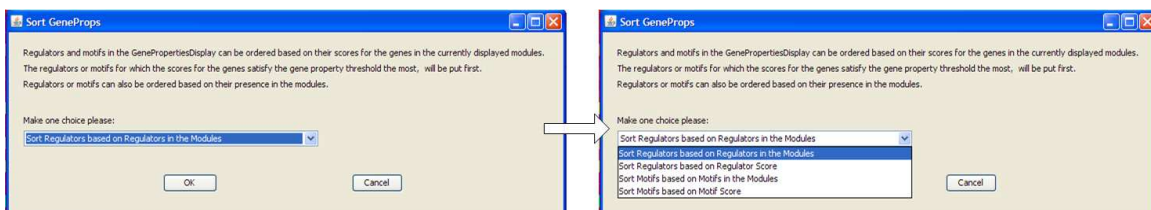The top 5 regulators/ motifs will be shown in the GenePropertiesDisplay.

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.

3.6.3 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 **Error! Reference source not found.** Figure 20):

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

3.6.4 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 **Error! Reference source not found.**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 GenePropertiesDisplay. 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 GenePropertiesDisplay, can be changed by moving the gene properties up or down in the list using the *"up"* or *"Down"* button.

3.6.5 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.
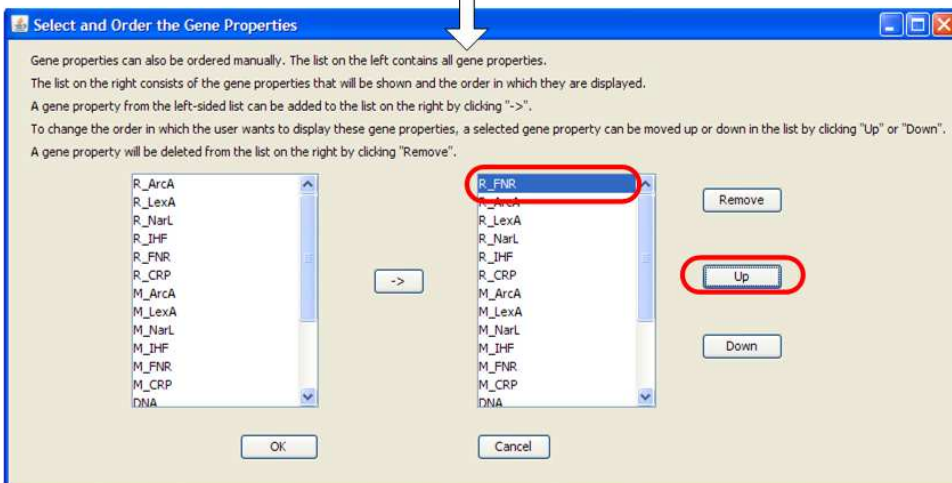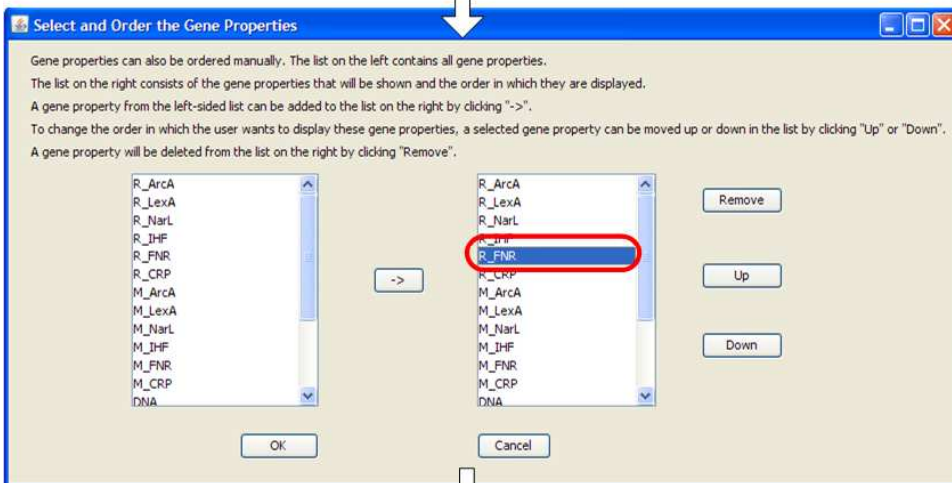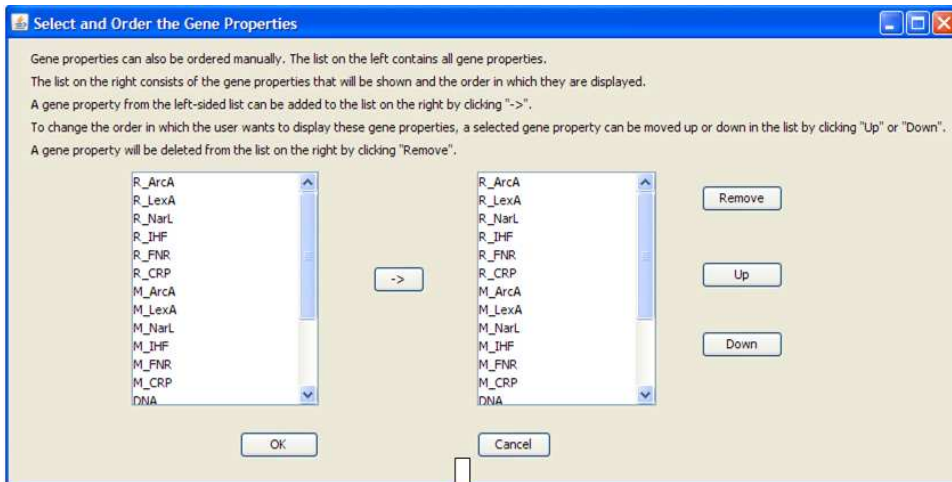
Figure 21: Manually change the order of the gene properties pop-up box. The list on the left contains all gene properties. The list on the right consists of the gene properties that will be shown and the order in which they are displayed. A gene property from the left-sided list can be added to the list 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"*.

3.7 Overview & Heatmap Display

In addition to the three image displays, ModuleDisplay, GenePropertiesDisplay and ExperimentPropertiesDisplay, ViTraM allows for some additional plots (see Figure 22**Error! Reference source not found.**):

- An overview of all modules (section 3.7.1)
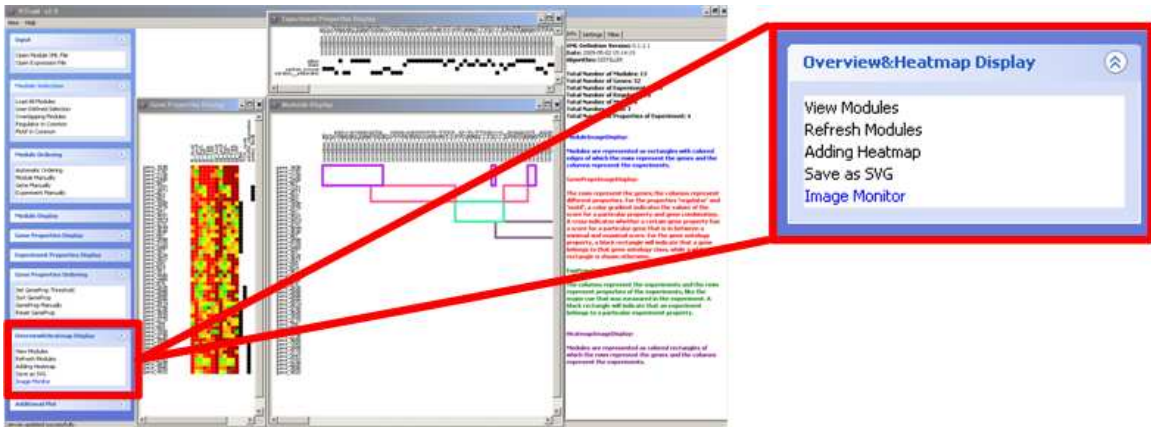- A heatmap of all modules (section 3.7.2)

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

3.7.1 Overview of all modules

Sometimes, too many modules are included in the ModuleDisplay 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 active Modules"* button in the *"Overview* & *Heatmap Display"* panel, the OverviewDisplay is shown (Figure 22). This display shows an overview on all modules in the current ModuleDisplay.

Whenever the content or layout of the ModuleDisplay changes, a novel overview of all currently displayed modules can be obtained by clicking on "Refresh Modules" in the "Overview & Heatmap Display".

The overview of the modules can also be saved as an SVG picture (see also section 4).

3.7.2   Heatmap

ViTraM allows visualizing also the expression profile of each gene separately in the modules. By clicking the *"Adding Heatmap"* button in the "Overview & Heatmap Display" panel, a heatmap will be added to the current OverviewDisplay (**Error! Reference source not found.**See Figure 23). 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. The image monitor will tell whether the heatmap plot is finished. The heatmap of the modules can also be saved as an SVG picture.
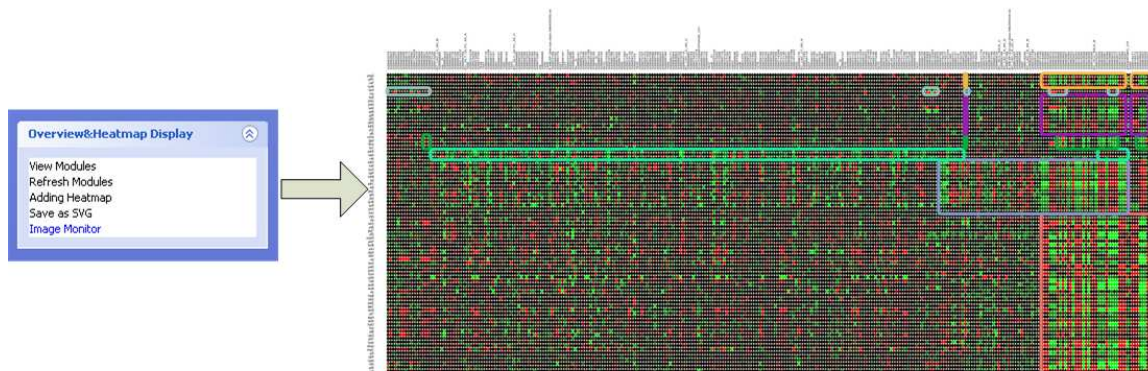

Figure 23: Plot of the heatmap of the modules currently displayed in the ModuleDisplay. First clicking on "View All Active Modules", an empty frame will be shown, clicking on "Refresh Modules", all of the active modules plotted will be shown on this frame. Clicking on "Adding Heatmap", ViTraM will start to plot the heatmap for these visualized active modules. A monitor will tell when the heatmap is ready. The X-axis represents the conditions. The order of the experiments corresponds to the order of the experiments in the ModuleDisplay. The Y-axis represents the genes. The order of the genes corresponds to the order of the genes in the ModuleDisplay. For each module, the expression values of all genes in the module are plotted for each experiment. Each module is represented by one colored rectangles.

3.8   Additional plot

The average expression profile of the genes in a module is shown after clicking the *"Plot Modules"* button in the "Additional Plot" panel (see**Error! Reference source not found.** Figure 24). 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 ModuleDisplay.
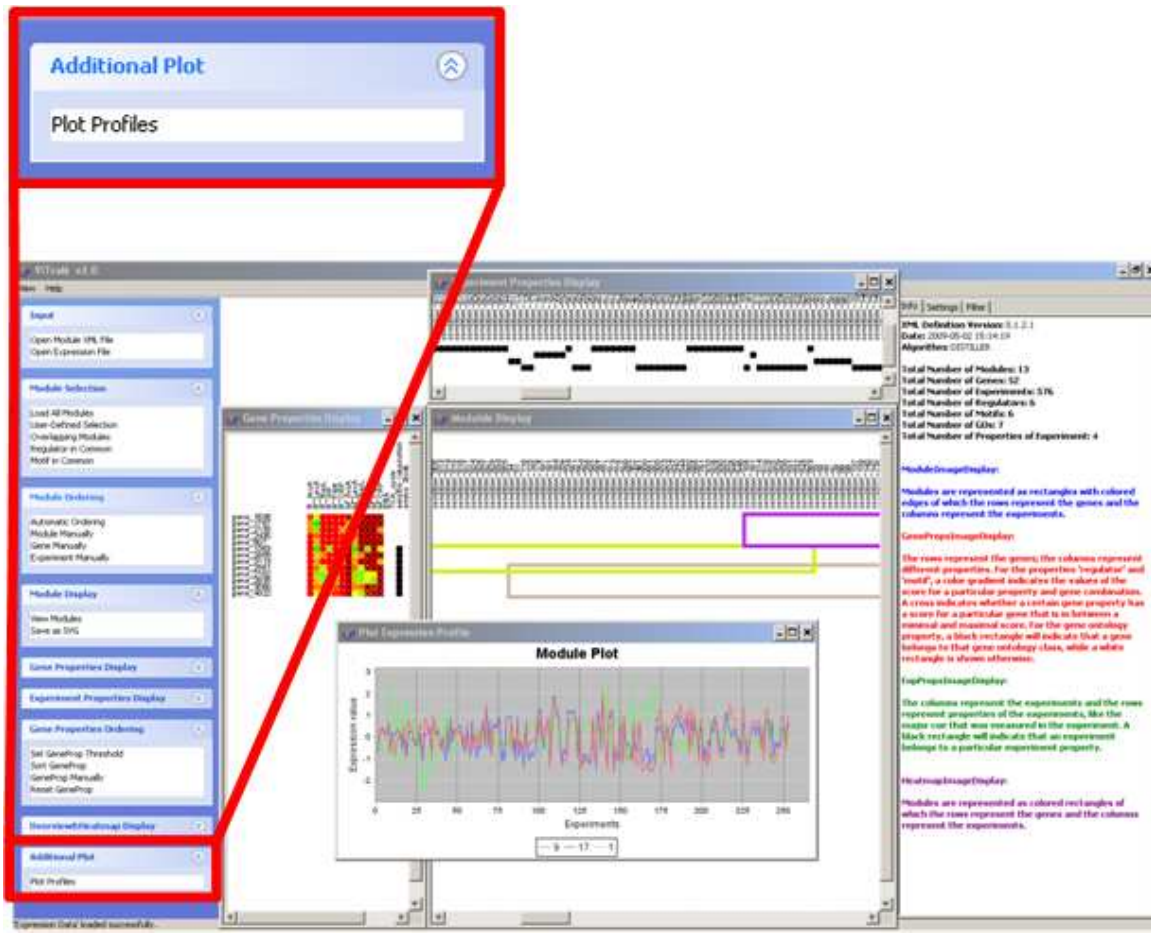
Figure 24: Plot of the average expression profile of the modules currently displayed in the ModuleDisplay. The X-axis represents the conditions. The order of the experiments corresponds to the order of the experiments in the ModuleDisplay. The Y-axis represents the expression values. For each module, the average expression values of all genes in the module are 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.

4 OUTPUT FILES

The resulting four images of "ModuleDisplay", "GenePropertiesDisplay", "ExperimentPropertiesDisplay" and "OverviewDisplay" can be saved as an SVG file (see Figure 25**Error! Reference source not found.**) by clicking on *Save as SVG >> SVG format*.

These SVG files can be loaded in software such as InkScape (http://www.inkscape.org/download/?lang=en) for saving them as high quality PNG files.
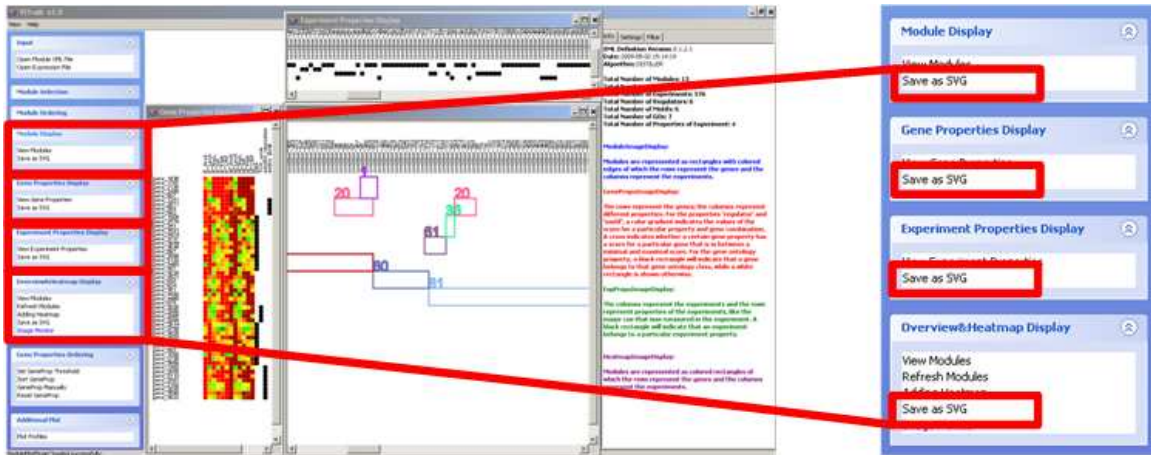

Figure 25: Several possibilities for exporting data from ViTraM are available.

5 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**. DISTILLER: a data integration framework to reveal condition dependency of complex regulons in Escherichia coli**. Genome Biology 2009, 10:R27.