Widgets for Functional Genomics

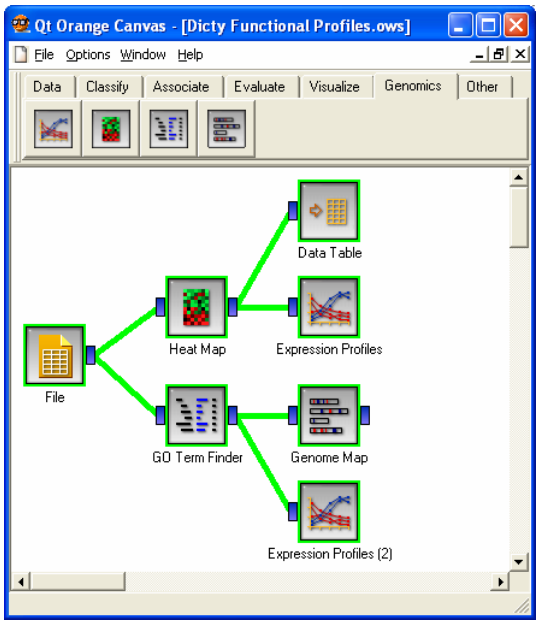
Use Orange functional genomics widgets to:

- View gene expression profiles;
- Visualize microarray data using heat maps;
- Find your genes on the genome;
- Browse through gene ontology, find which functions are characteristic for a specific group of genes;
- Combine them with other Orange widgets for clustering, supervised data analysis, and data visualization;

Orange and Orange Widgets

Orange is a general purpose machine learning and data mining suite. For machine learning researchers, it provides a flexible platform for development of new data analysis algorithms through Python scripting. For data miners, it offers a versatile component-based graphical user's interface that features visual programming and high degree of interactivity.

Orange Widgets are basic blocks that can be put together by means of visual programming to design powerful data exploration applications that fit one's needs. Widgets communicate, pass tokens that include data, attribute lists, models, etc. to each other, and often allow interactive selection of data subsets that are passed on to other widgets for further analysis.



About this Catalog

This catalog presents widgets that we have build to address some data analysis and visualization tasks in functional genomics and microarray data analysis. The widgets included in this set are: gene expression viewer, heat map visualizer, gene ontology browser and genome map. The catalog is a user's guide to the widgets, with the aim of giving an outline of the functionality.

Widgets are building blocks of data mining applications we call Orange Schemas. For this purpose, the catalog also provides some ideas on how to combine widgets into typical and less typical schemas.

Inside:

Heat Map 2

Expression Profiles 4

GO Term Finder 6

Genome Map 8

The Data 10

Other Widgets 14

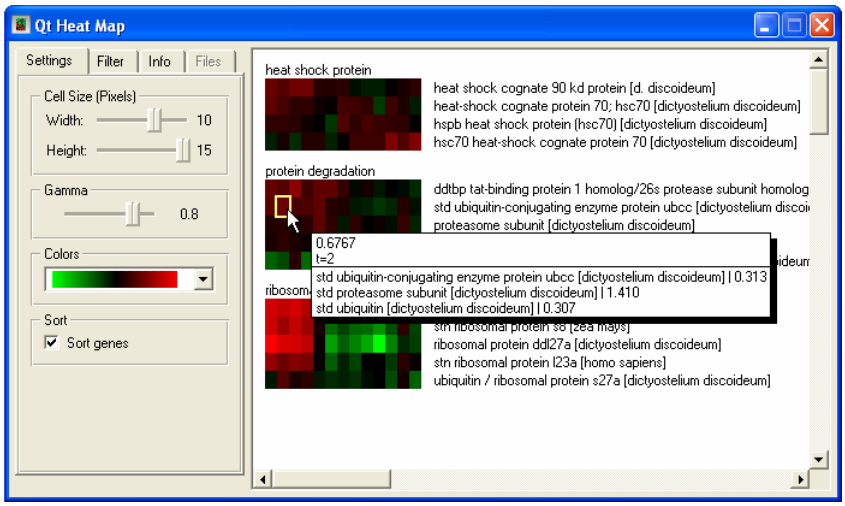
Widgets for Functional Genomics

Heat Map

Input: Examples

Output: Examples, Classified Examples

Heat Map Widget



This widget displays gene expression profiles as heat maps - a commonly used type of visualization. Several data sets can be displayed side by side, which can be very useful when comparing time series observed under different conditions or for different mutants.

One can control the dimension of generated graphics, choose the color palette or select if genes are to be sorted by the center of their expression (Settings tab; when sorted, genes that are overexpressed early will appear first in the list, and last will be those with late overexpression). When working with thousands of genes which do not all fit on screen, it is reasonable to merge the expressions and display a certain number of genes in a single row (Merge Rows in Filter tab).

The color palette can greatly influence the way we perceive data. The rate of color transition is controlled by Gamma parameter in the Settings tab. One can set lower and upper thresholds for gene expressions to display. Expressions outside this boundary are colored with the same minimal or maximal saturation color. All this is done in the Filter tab.

The parameters in the Info tab define what attribute to display for gene annotation alongside each row and what information to display in the tooltip when moving the mouse over heat map. Any descriptive, discrete data from input can be displayed, which usually includes: annotation, gene ID or gene function attributes.

The Files tab is used to manage more than one data input signal.

"Interactive gene subset selection is a powerful mechanism for data exploration, but requires another widget to intercept the output token and display the data."

Input Data, Data Subset Selection

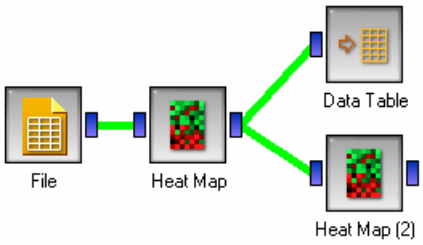
This widget takes one or many Examples of type ExampleTable - data tables containing gene expression data. Each example (gene) can be also assigned a class; in that case, genes of the same class will be displayed in the same heat map group. An example snapshot above, for instance, shows genes grouped within three different classes (heat shock protein, protein degradation, …). The data may also contain meta attributes, which can be either displayed for gene annotation or within a tooltip.

Click-and-drag over the heat map would select rows (genes) of interest. Use Shift key for multiple, disjoint selection. The selected subset is sent to widget's output, and the type ExampleTable (no class information assigned) or Classified Examples (in case where data selected belongs to several classes) is used. Interactive gene subset selection is a powerful way for data exploration, but requires another widget to intercept the output token and display the data.

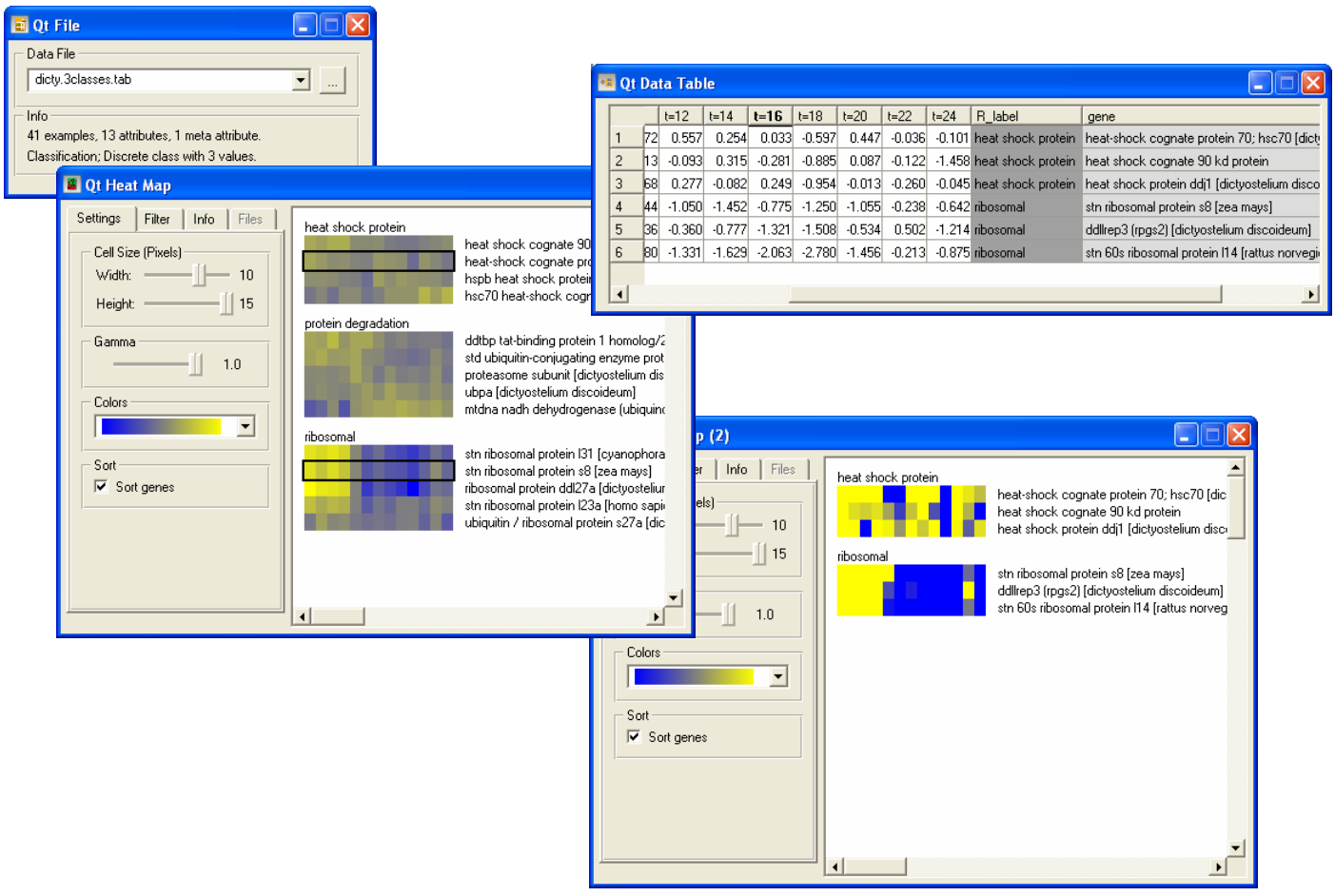
Widget Catalog

A Simple Schema With Heat Map Widget

Here is a simple schema that loads a gene expression data file by selecting it in the File widget. The particular data set we have used includes genes that are assigned to different classes. A heat map is generated for each class in the input data. Each gene is displayed in its own row, and the widget was instructed to display three gene per row (Merge Rows set to 3). We selected a row (three genes) genes from the "heat shock protein" and "ribosomal" class. Heat Map sends the selected data to the Data Table widget, which displays the data received in the tabular format. Like in any other Orange schema, any change in the selection of the widget immediately propagates through the schema. This effect can be of value for data exploration when having both Heat Map and Data Table widgets visible on the desktop.



Combination of two Heat Maps that we also use in the schema is particularly interesting from the view point of visual programming. It forms an effect we call a "magnifying glass". Genes selected in one heat map are sent to the other one, where rows can be enlarged to observe the details or particular genes. This combination may work particularly well when dealing with hundreds or thousands genes.



Widgets for Functional Genomics

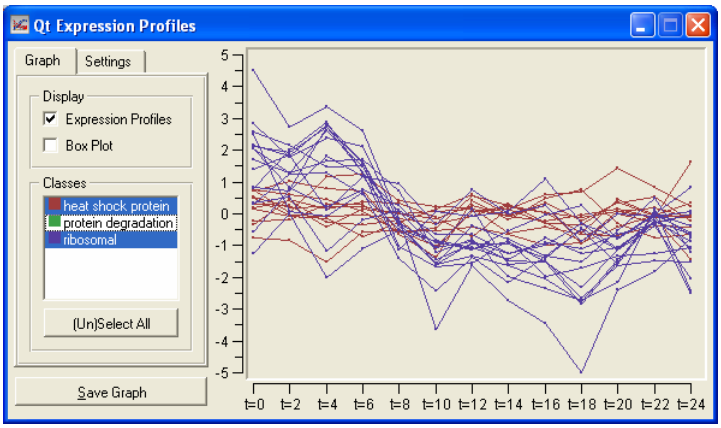
Expression Profiles

Input: Examples

Output: None

Expression Profiles Widget

This widget provides support for another type of visualization that is widely used for display of time series gene expression profiles. Gene expressions are displayed using a line graph, where each line represents the expression of a single gene in different time points or conditions.



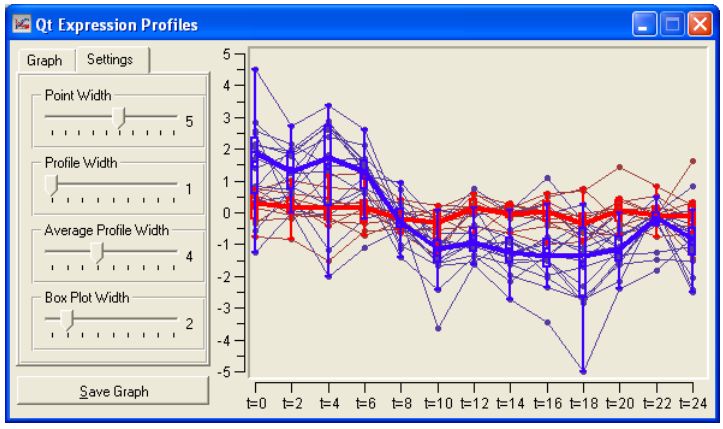
On its input the widget accepts Examples of type ExampleTable. Only the continuous attributes are displayed; if the data contains attributes of other types, these are ignored. The order on the x-axis corresponds to the order of the attributes (columns) in the input data set. If examples are classified, then one can select which classes to display.

Display Options, Zooming

Expression profiles can be display individually by checking the Expression Profiles option in the Graph Tab. Alternative presentation is through the display of an per-group average expression. These can be complemented with a box plot display that provides the information on the 1st and 3rd quartiles.

The width of lines representing the individual or averaged expression profiles, the width of lines of the box plot and point width can be set in the Settings tab.

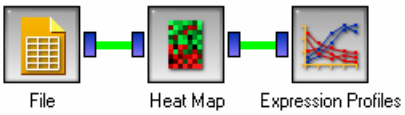
One can zoom to a part of the graph by mouse left-click-and-draw the rectangular region to be zoomed. This can be repeated on the already zoomed image, resulting in a larger zoom. A right click will undo the last zoom, until the whole graph is displayed.

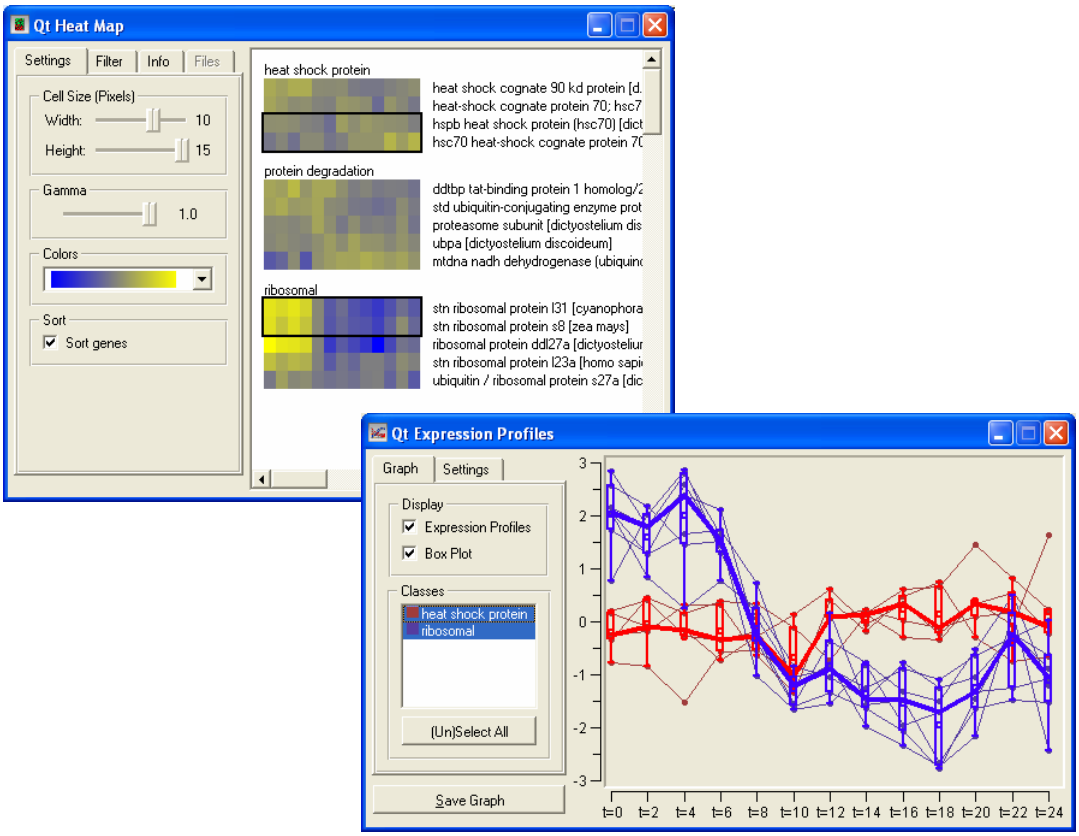


Widget Catalog

Combination with a Heat Map

The schema we present here is really simple, but it is also the one you may use often. The data read by a File widget is sent to the Heat Map. There, we have selected several genes and send it to Expression Profiles widget. Notice that the heat shock protein genes were rather grayish in the heat map, and their expression plot is rather constant and around zero. Differently, the expression of selected ribosomal genes drops from ovexpressed and finishes with underexpression, which is both visible at heat map and expression profile display.





When connected, widgets in Orange schema interchange tokens. For instance, any gene selection change in a Heat Map widget will immediately propagate to connected widgets.

Widgets for Functional Genomics

GO Term Finder

Input: Cluster Examples, Reference Examples

Output: Examples, Classified Examples

Gene Ontology Term Finder Widget

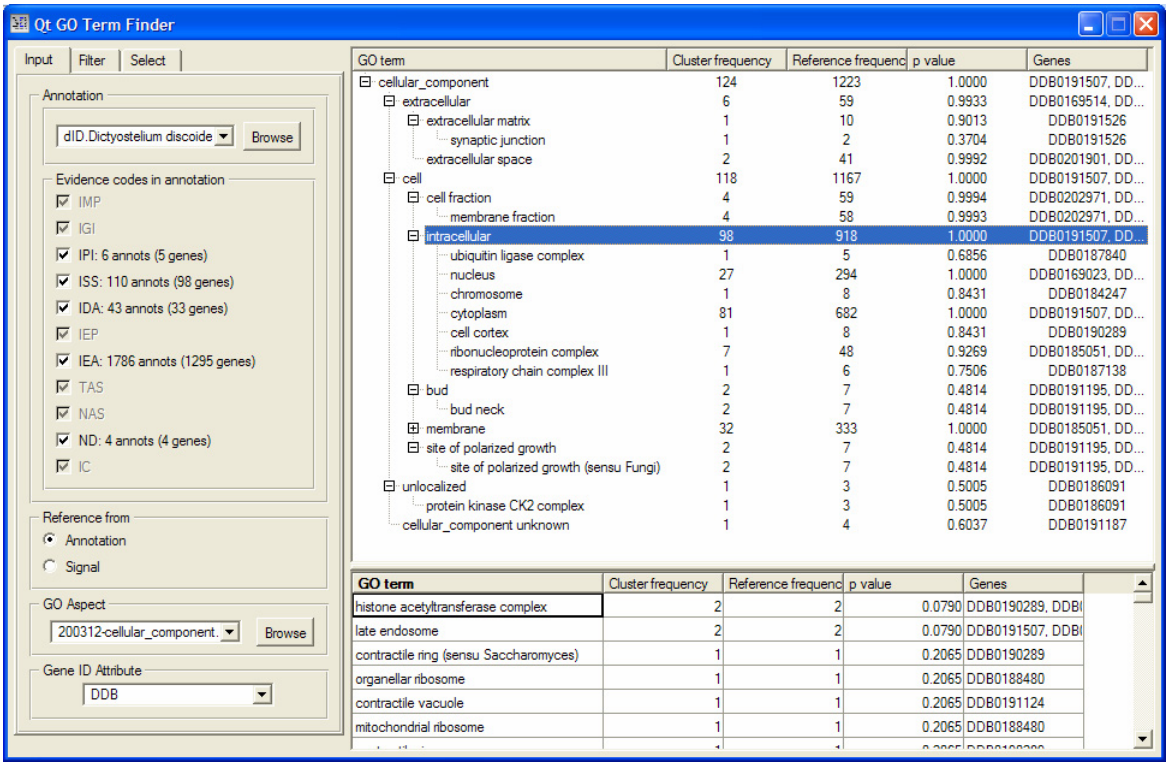
When given a set of genes one may want to determine the most significant GO (Gene Ontology) terms shared by them. Usually, the significance of shared GO terms is calculated in respect to the terms in the whole genome. In GO Term Finder, one can select an arbitrary set of genes as reference, thus enabling a group vs. group testing. For this reason, the widget takes two inputs of type ExampleTable: "Cluster Examples" holds a list of genes with gene IDs from the test set, and "Reference Examples" genes from the reference set (option "Reference from Signal"). If a reference set is not given, then whole genome is taken as reference (option "Reference from Annotation").

The only required attribute in input data set is gene ID; this is automatically detected and selected by the widget based on the contents of the attribute fields. If and when needed, one can choose a different attribute (option "Gene ID Attribute"). Any other attributes included in the data set are not considered; they are ignored in the presentation of this widget, and used only to form a data set based on the group of selected genes.

GO Term Finder Widget scans Genomics\GO and Genomics\Annotation directories in the Orange directory (e.g. c:\Python\Lib\Site Packages on Win32) for files holding GO and annotation data. These files are listed in the Annotation and GO Aspect combo boxes. The widget determines the most appropriate GO and annotation files based on gene IDs present in the input data. One can also use annotation and aspect files in other locations (Browse buttons).

Based on the annotation used, widget lists available evidence codes; the ones that are selected are those that will be used in the term finder.

Criteria for significant GO terms are set in the Filter tab: one can filter GO terms by p value, minimal cluster frequency, or GO depth. Significant GO terms are then listed in the table in the lower part of the widget, and the corresponding ontology is displayed in the top tree view.



Widget Catalog

Term-Based Gene Selection

Widgets supports the selection of GO terms in ontology graph. The data associated to the corresponding genes will be sent to the output. Corresponding genes are either those that are annotated directly to the term selected, or annotated with the related term upper in the ontology (direct or indirect annotation, Select tab). When selecting many GO terms, one can choose how to treat genes shared by many GO terms. One can omit them (option Disjoint) or place them in all of the corresponding GO terms (option Inclusive). Option "Add GO term as new class" enables one to append an additional attribute to output data, listing the selected GO terms belonging to individual genes.

Computing the Significance of GO Term Annotation

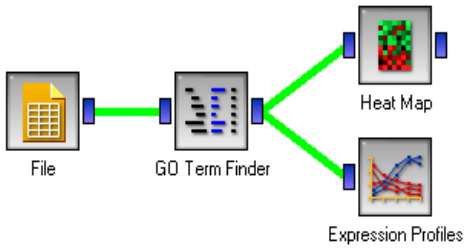
The significance (p value) of a GO term is calculated using the binomial distribution. If G is the number of reference genes annotated (directly or indirectly) to a GO term and N is the total number of genes in reference, then the probability of having x or more genes assigned to the same GO term by chance (out of all n genes in our test set) can be calculated with the following formula

$$p = \sum_{j=x}^{n} \binom{n}{j} f^{j} (1-f)^{n-j}$$

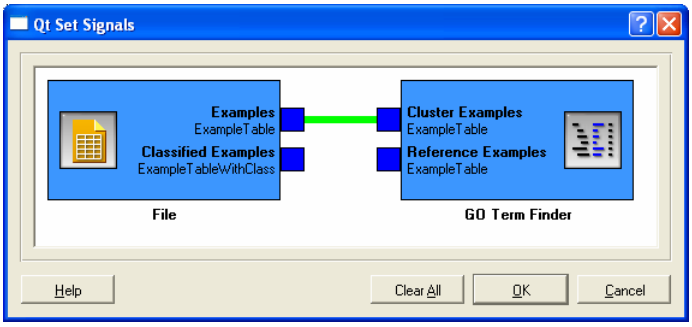
where $f = \frac{G}{N}$.

An Example Schema with GO Term Finder

The schema on the side shows a basic use of the GO Term Finder widget. Notice that a dialog will appear when connecting File and "GO Term Finder" widgets. This is because File widget outputs Examples of type ExampleTable, but "GO Term Finder" accepts two inputs of same type (Cluster Examples and Reference Examples). In such cases, Orange Canvas connects the output to all inputs of same type (in our case File's Examples to both GO Term Finder's inputs), and then asks the user to confirm or change the connections. One can invoke the dialog at a later time by double-clicking on the connection in the schema.



Since we want to test for significance against the whole genome, we connect to the output of a File Widget Cluster Examples only. In case we want to use a specific set of genes for reference, we can add another File widget to the schema and connect it to the Reference Examples of the GO Term Finder.



Following the schema, one first loads an expression data file by selecting it in the File widget. After the significant GO terms are determined in GO Term Finder widget, one can select the GO terms of interest. Only the input data (expression profiles) associated with genes annotated to the selected GO terms are then sent to the two subsequent widgets, where the data are displayed. Selecting different GO terms causes subsequent widgets to refresh automatically with new data.

In designing GO Term Finder, and in implementing its computational aspects, we aimed at replicating the functionality of a popular, similar, but stand-alone application at www.yeastgenome.org/help/goTermFinder.html.

Widgets for Functional Genomics

Genome Map

Input: Examples

Output: Examples, Classified Examples

Genome Map Widget

Genome Map widget can be used to explore the genomic position of genes and select a subset of genes from a genomic region.

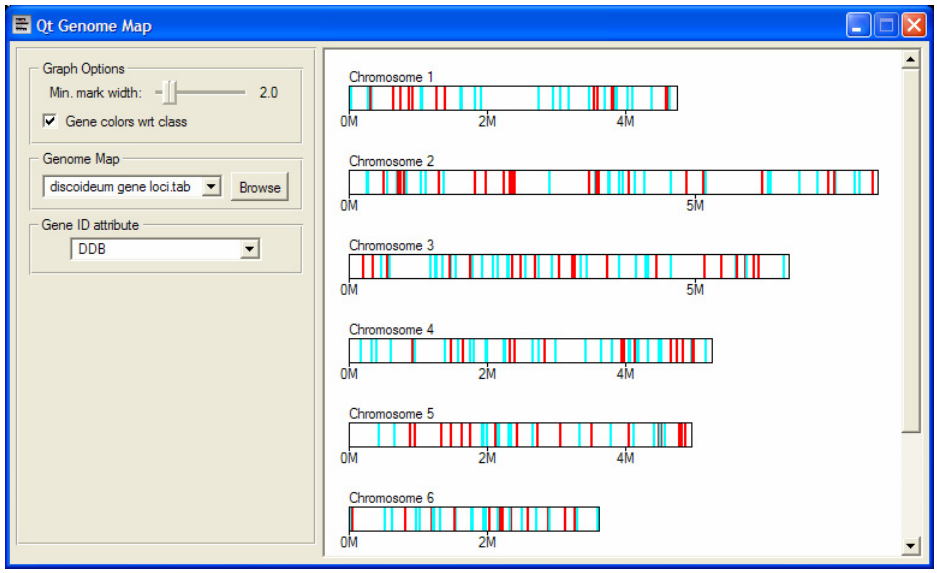
The widget accepts Examples (of type ExampleData), i.e. the data on genes which has to include an attribute with a gene ID; all other attributes are optional and are not used in the presentation of the widget.

To display gene position, the widget uses species-specific genome map files where the genomic coordinates for each gene are stored. Each time the widget accepts a new input data, the most appropriate map — the one that best matches the gene IDs — is selected automatically. User can change this selection in the combo box.

If genes in the input data are classified to different groups, a coloring is used to distinguish between group members. An alternative presentation is with all gene positions displayed in black (parameter "Gene colors wrt class"). User can also set a minimum pixel width when drawing a gene (parameter "Min. mark width").

Region of the genome is selected by a click-and-drag over a chromosome. More than one region can be selected by holding the Shift key, when selecting the next genomic region. The selected area is colored gray and only the input data with selected genes are sent to the output.

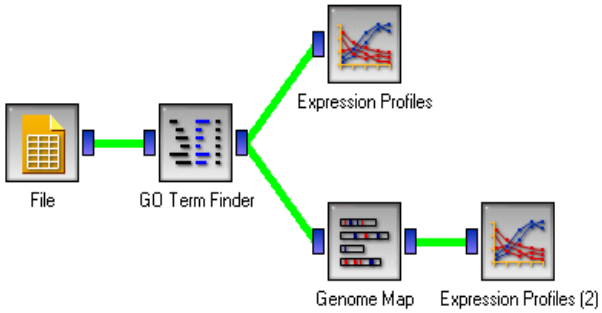
A region of the genome can be zoomed by a click-and-drag outside of chromosomes. A right click will bring the display back to the previous zoom, until the whole genome image is shown.

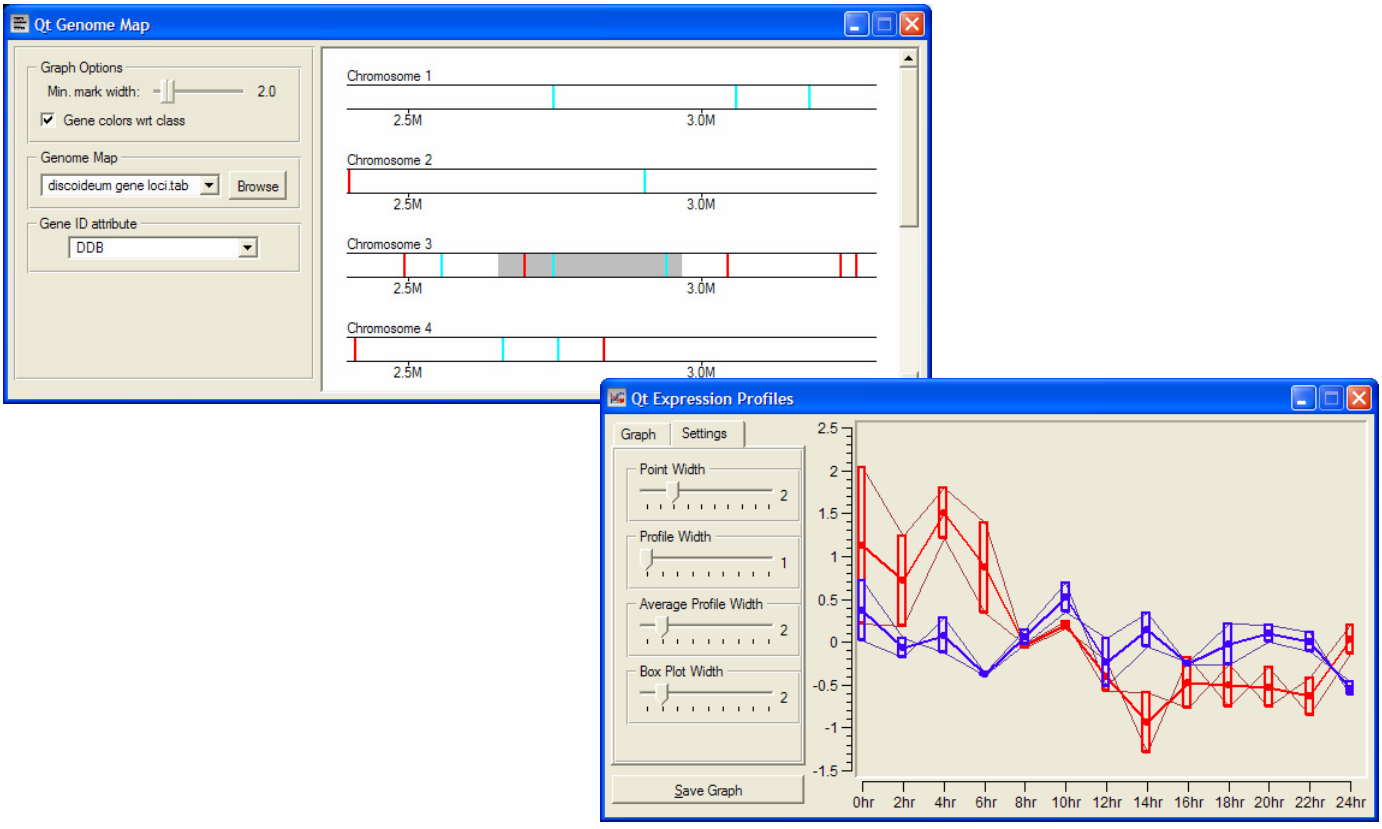


Widget Catalog

An Example Schema with Expression Profiles

The schema on the right loads gene expression data using the File widget. In the GO Term Finder we then select several significant GO terms, and the associated input data is then passed to the Genome Map and Expression Profiles widgets. In the Genome Map widget we zoom to a region, where we select two genes (the gray area). Their expression data is displayed in the Expression Profiles (2) widget.





Widgets for Functional Genomics

The Data

Orange widgets for functional genomics use several sources of information, including a user-defined data set and information on gene annotation, gene ontology and genome map files. Let us consider first the input data set. For Expression Profile and Heat Map widget, the main part of the data includes the information about gene expressions, and may additionally include the information on gene ID and gene groupings (classifications). Information on gene ID is necessary if we want to combine these two widgets with GO Term Finder or Genome Map.

Let us look at the example data file. The snapshot below shows a small data set as edited in Excel spreadsheet. It includes 11 genes, of which each expression was measured at 5 different time points. The first row is how we tell orange what attribute names are. The first five attributes are gene expressions, and their names correspond to time when the expression was measured. Following is an attribute with gene ID (a standard ID from www.dictybase.org is used here). We did cluster these genes into two groups, and the group labels are given in the next column. The final column gives some annotation (think of it for now as some comment on the gene, and don't mix it with the annotation that comes from GO Term Finder).

Two very important rows follow the first one. The second row tells about the type of the attribute, which can be in Orange, basically, either continuous, discrete or string. The third row contains optional flags, which tell Orange how to consider the attribute. For microarray data, use blanks for attributes with measurements, that is, for attributes you would like to show in the graphs. Meta attributes are never displayed, but hold additional information. In our case, we store gene ID and our annotation in a meta attribute. A class attribute is used to group the examples (in our case genes), and widgets that display such data would most often present examples of different classes with different colors.

What we have shown is a data set in a '.tab' file; this is because we store such data in the files that end with '.tab'. Orange recognizes other data file formats, and for those you should check the on-line reference manual.

"Apart from tab-delimited data with header rows, Orange recognizes a number of other data file formats. They are all listed in the on-line reference manual available on the Orange's web site or in the documentation that comes with Orange installation."

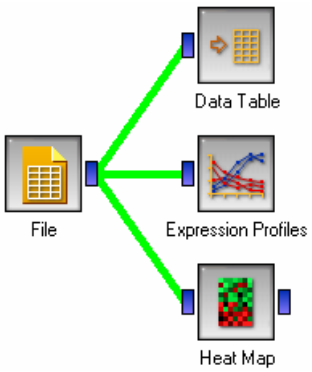
dicty small example.tab

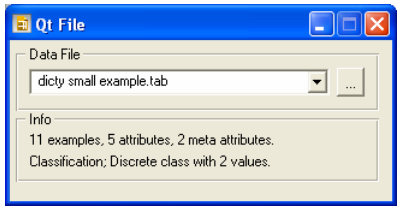
	A	B	C	D	E	F	G	H
1	0hr	6hr	12hr	18hr	24hr	gene id	cluster	annotation
2	continuous	continuous	continuous	continuous	continuous	string	discrete	string
3						meta	class	meta
4	0.7671	0.9505	-0.4704	-1.4279	0.8081	DDB0167249	1	metabolism
5	1.2263	0.7166	-0.1353	-0.6626	-0.3844	DDB0167445	1	metabolism
6	1.2676	1.1686	0.2319	-0.8947	0.7801	DDB0167402	1	metabolism
7	0.7957	0.4767	0.0403	-0.9461	0.6092	DDB0167598	1	metabolism
8	-0.6151	-0.8749	0.4597	0.5336	0.519	DDB0168111	2	metabolism
9	-0.0402	-0.574	-0.0606	0.0414	0.1311	DDB0168621	2	metabolism
10	0.7848	0.0258	-0.286	-0.5464	0.2289	DDB0168791	1	isocitrate metabolism
11	0.7479	0.4181	-0.3113	-0.1609	-0.0713	DDB0191663	1	protein folding
12	0.2429	-0.4371	-0.8938	-0.6535	-1.1846	DDB0191096	2	protein folding
13	0.7087	0.2086	0.8966	0.3686	0.4235	DDB0204297	2	regulation of cell cycle
14	0.0074	-0.8727	1.4157	-0.4335	-0.1141	DDB0186598	2	regulation of cell cycle
15								

dicty small example

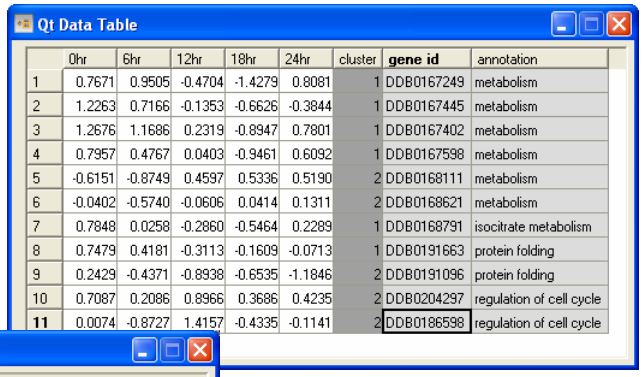
Widget Catalog

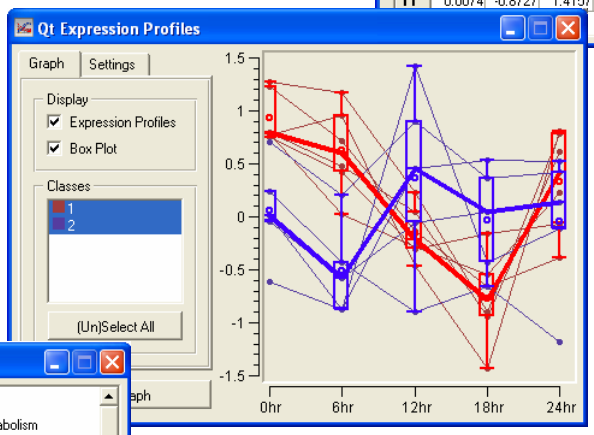
The schema on the right is just to show how such a data set would look when examined in the Data Table widget (a standard widget that comes with Orange) and in Expression Profiler. Notice that in the Data Table the class and meta attributes are marked with different shade.

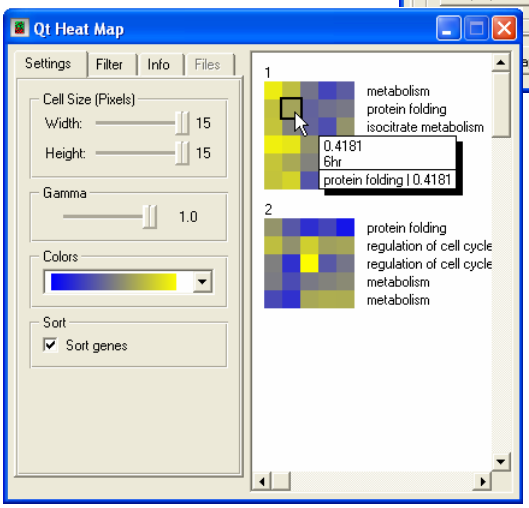




Notice that Expression Profiles widget would look just the same if we would remove the attributes on gene ID and annotation. But GO Term Finder would not function without the former, and the later can be additionally painted in the Heat Map (at a side of each row). We could explore all these variants in this documentation, but we won't. It is your turn now.







Widgets for Functional Genomics

Gene Ontology, Annotation and Genome Map Data

Orange widgets for functional genomics already come equipped with the latest data files for S. cerevisiae and D. Dicty-ostelium. To update the data or prepare it for other organisms, follow the steps mentioned on these pages. If this is too hard, contact us and we will be happy to provide you the information files for your data sets.

Two widgets, namely gene ontology term finder and genome map, need also other information besides input data from the user to function properly. GO Term Finder widget, for instance needs gene ontology and gene annotation data. Genome Map, on the other hand, needs to know how to map genes to the genome, i.e. use gene ID information and obtain chromosome coordinates. These data sets change frequently due to new updates and are for gene ontology and annotation available at www.godatabase.org. The mapping is, of course, species-specific; for instance, gene positions for yeast can be found at ftp.stanford.edu/pub/yeast/data_download/chromosomal_feature/ chromosomal_feature.tab.

Here we give short instructions on how to transform the needed data into a format that can be used in the GO Term Finder widget, and we also discuss on the format of the data file that stores the mapping from gene ID to the genome coordinates.

Preparing Gene Ontology Data

Latest Gene Ontology data can be downloaded from www.godatabase.org/dev/database/archive/latest/go_YYYYMM-termdb-tables.tar.gz. Here, YYYYMM is the version date. To convert such a file to the data file used by Orange, decompress the file in the *Orange-Widgets\Genomics* directory in Orange installation directory (on Win32 this would normally beat *C:\Python\Lib\site-packages\orange*). This will create a *go_YYYYMM -termdb-tables* subdirectory. Use shell (on Win32 use DOS shell), set your directory to *Orange-Widgets\Genomics*, start Python (just type "python" in the command shell), and once in Python type:

```
import GOlib
GOlib.txtGO2pickle\
('YYYYMM', './GO')
```

This will output some basic statistics and will create three new files in the *./GO* subdirectory, one for each GO aspect: YYYYMM-biological_process.go, YYYYMM-cellular_component-.go and YYYYMM-molecular_function.go. Of course one can instruct the procedure to create the files in a different location, but the advantage of putting files in the *./GO* subdirectory is that the GOTermFinder automatically makes them available for selection in the combo box. Files elsewhere must be opened by the "Browse" option.

Widget Catalog

Preparing the Annotation Data

While the GO data is a single description of gene products that can be applied to all species, the Annotation data is species specific. A list of all species is available at www.geneontology.org/GO.current.annotations.shtml.

Similarly as for GO data, place the uncompressed annotation file in the *orange\OrangeWidgets\Genomics* directory (e.g, for yeast you would download the file www.geneontology.org/cgi-bin/GO/downloadGOGA.pl/gene_association.sgd.gz and decompress it into gene_association.sgd). Then start Python in the *OrangeWidgets\Genomics* subdirectory of the Orange installation directory, and type:

```
import GOlib
GOlib.txtAnnotation2pickle
('gene_association.sgd',#
'./Annotation/Saccharomyces
cerevisiae.annotation')
```

This will create a new file in the *./Annotation* subdirectory. Be careful not overwrite an existing version you might still need.

Data Format for Gene Genomic Position

Genome map files should be stored in the *./Genome Map* subdirectory. The files use the syntax of Orange data file, and a typical composition is shown in the example data table below. This, for convenience of this example, includes only information about two chromosomes and four genes.

As other Orange data files, the first three rows are needed by Orange, they list attribute names (geneID, chromosome, start, stop) and types (string, discrete, continuous). In the first few lines of the data we then define the chromosomes (Excel rows 4 and 5). The chromosome name, as it will be displayed in the Genome Map widget, is given in the first column, followed by a chromosome ID number that will be used in all subsequent references to the chromosome, a starting position (this must be zero) and the chromosome length.

In the subsequent lines, we state the genomic position of all genes. These rows start with a gene name, followed by an ID of a chromosome where the gene is located, and two location numbers, the starting and ending position of the gene.

chromosomes.tab

	A	B	C	D
1	geneID	chromosome	start	stop
2	string	discrete	continuous	continuous
3				
4	Chromosome 1	1	0	230210
5	Chromosome 2	2	0	813140
6	TFC3	1	151170	158930
7	VPS8	1	143711	147535
8	EFB1	2	134334	136892
9	YPR202W	2	232440	235934

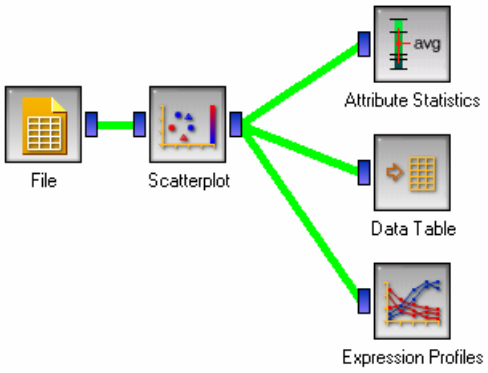
chromosomes

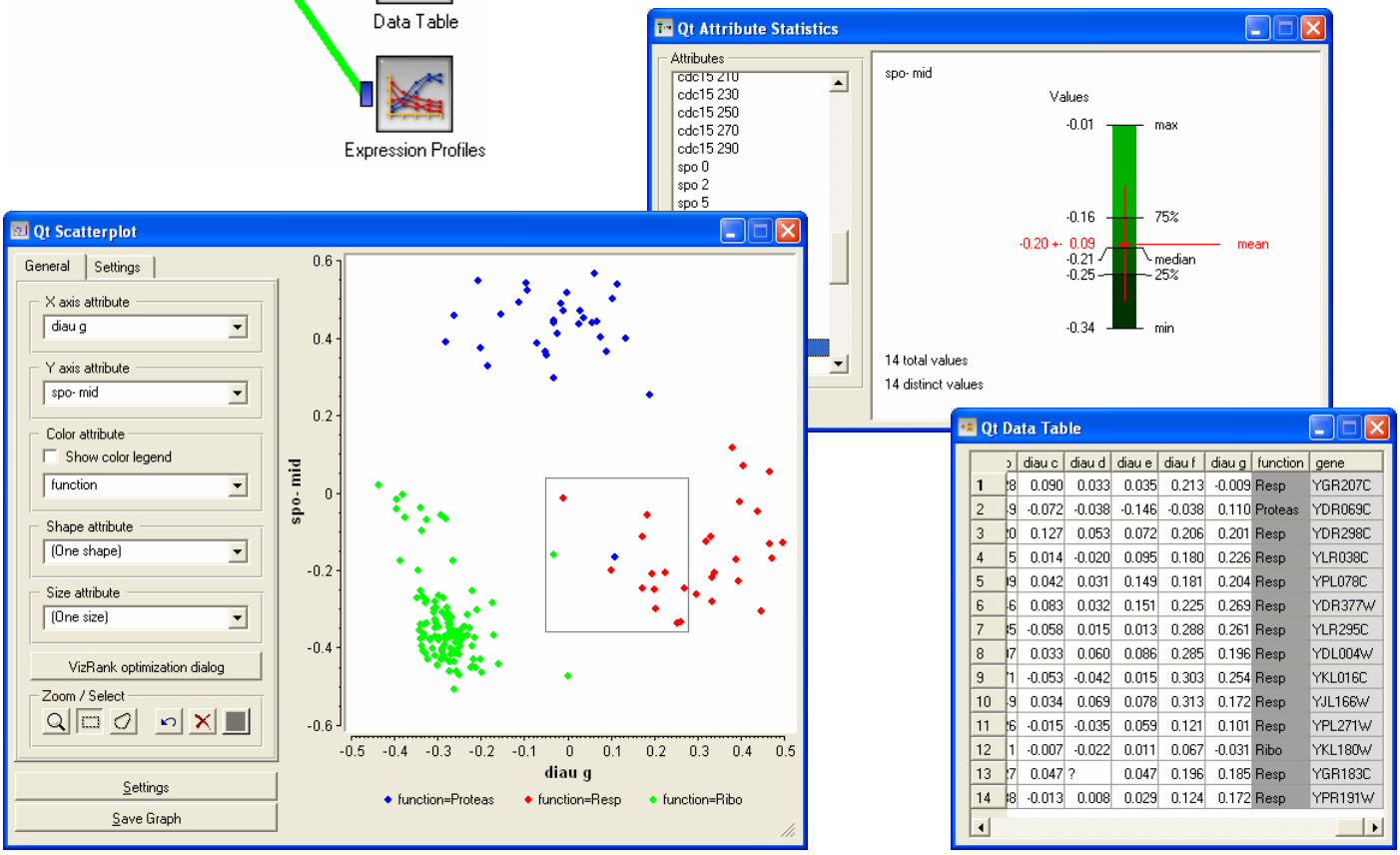
Widgets for Functional Genomics

Other Orange Widgets, and Do You Need Them?

As Orange is a general purpose data mining suite, it includes a number of other widgets, none of which is specialized for functional genomics. And of course you may need them to analyze your data further. In fact, we have already met two very useful and simple Orange widgets: the File widget for loading in your data, and Data Table widget to display the data in the tabular form. Here we just briefly mention some other widgets that may come handy. For the rest, please visit Orange's web site and download the documentation on widgets. You may also run Orange Canvas, and simply experiment with widgets listed in the toolbar — Orange will allow you to connect only the widgets that share compatible communication channels, so you really cannot go wrong in crafting your data analysis scheme.

There are several widgets that may be useful when analyzing microarray data. For instance, consider a schema below. We have used it to analyze the data gene expression data set on budding yeast *S. cerevisiae*. The data includes 79 different DNA microarray hybridization measurement experiments. Each gene in this data set is classified to one of six functional classes. This data has been, among others, studied by Brown et al. (PNAS 97: 262–267, 2000) to figure out if one can predict the functional class from expressional profile. In our data set presented here, we only use those on respiration (30 genes), cytoplasmic ribosomes (121 genes), proteasome (35 genes). An interesting scatterplot for this data set is shown below. In the scatterplot, we have selected a group of genes, displayed it in the table and expression profiler, and computed some basic statistics for attributes.

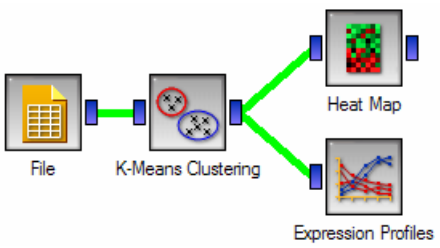


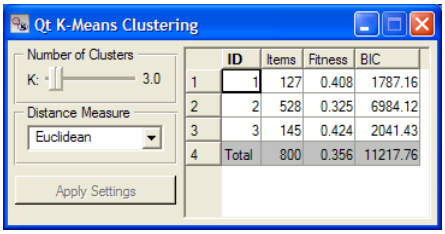


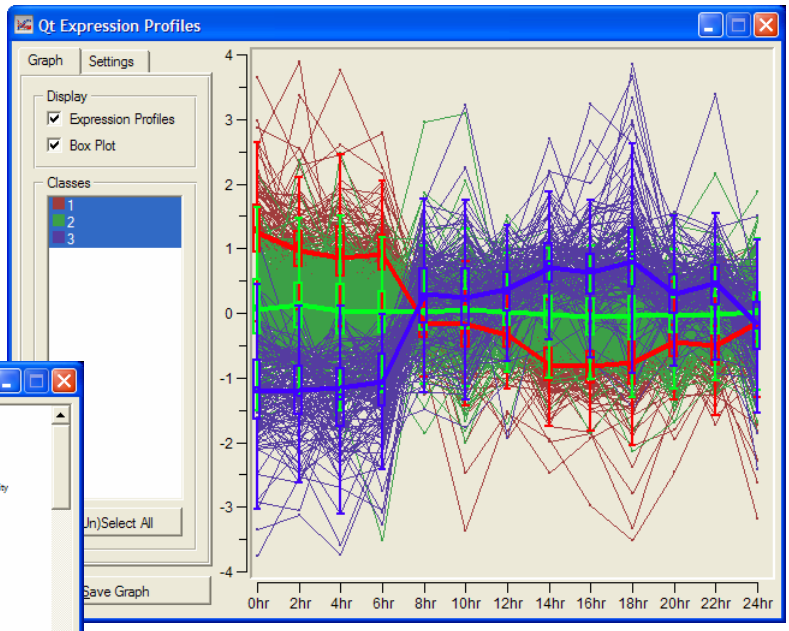
Widget Catalog

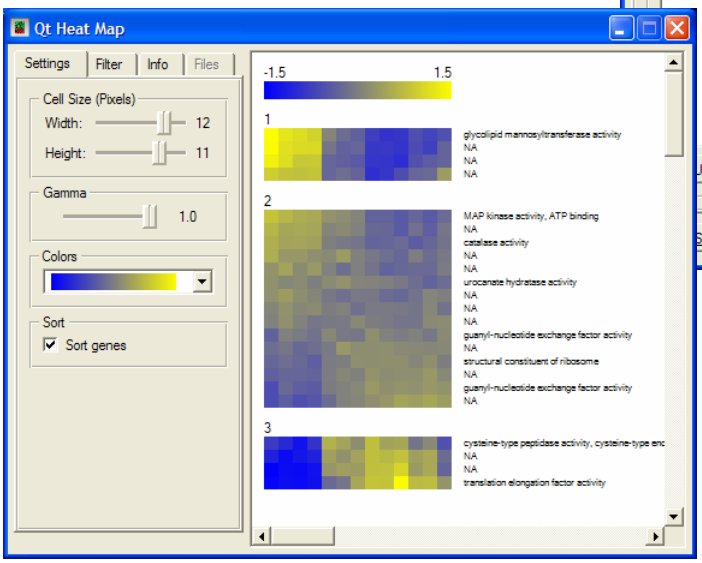
Here is another example schema. It involves k-means clustering. Notice that clustering in Orange is simple: connect a clustering widget to a source of data, and returned is the data plus a class attribute, which equals to the ID of a cluster. For the screenshots we have used 800 gene expression profiles as measured during the development of *D. Discoideum* (N. Van Driessche et al., Development, 129, 1543-52, 2002).

Notice that heat map and expression profiler nicely show different classes of genes as assigned by clustering.









Artificial Intelligence Laboratory, University of Ljubljana

Artificial Intelligence Laboratory
Faculty of Computer and Information Science
University of Ljubljana
Trzaska 25
SI-1000 Ljubljana, Slovenia

Further Reading

Curk T, Demsar J, Xu Q, Leban G, Petrovic U, Bratko I, Shaulsky G, Zupan B (2004) Microarray Data Mining with Visual Programming, Bioinformatics, *submitted.*

Demsar J, Zupan B: Orange: From Experimental Machine Learning to Interactive Data Mining, White Paper (www.ailab.si/orange), Faculty of Computer and Information Science, University of Ljubljana, 2004.

Orange's Web Site:

www.ailab.si/orange

Orange Functional Genomics Team

Orange widgets for functional genomics are a result of our collaboration with Departments of Molecular and Human Genetics and Biochemistry and Molecular Biology at Baylor College of Medicine, Houston, TX, and Jozef Stefan Institute, Ljubljana, Slovenia.

A number of people that specialize either in biology and genetics or in computer science participate in this project. The team at University of Ljubljana includes Tomaz Curk, Blaz Zupan, Janez Demsar, Gregor Leban and Peter Juvan, who specialize in data mining and visualization. Widgets for functional genomics were crafted in collaboration with biologists Gad Shaulsky, Nancy Van Driessche, Qikai Xu (Baylor) and Uros Petrovic (J. Stefan Institute). We here also acknowledge the support and help from other our colleagues from the participating institutions.

Widgets for functional genomics presented in this catalog live in a data mining environment called Orange. The development of this comprehensive data analysis suite has started in late 1990s by Janez Demsar and Blaz Zupan, and has since involved a number of people from different places around the world. See Orange's web site for updates, documentation, installation packages, and acknowledgements to participating researchers, institutions and granting agencies.