ermineJ help system

This help system is organized into the following main topics. You can also access these topics directly from the Help menu within ermineJ.

About ermineJ	What this software does and who it is for
Tutorials	Features and how-tos.
Data file formats	Information about the input files
FAQ	Frequently asked questions, with answers

ermineJ

Tutorials and topics

The following main step-by-step help pages are available:

- Notes on installing the software
- Startup and overview of the main software window
- Running an analysis. This topic is geared towards ORA analysis, but the other methods have very similar steps.
- Running a resampling analysis. This topic only covers the differences between ORA and the resampling method.
- Running a correlation analysis. This topic only covers the differences between ORA and the correlation method.
- Exploring analysis results.
- Viewing results in detail.
- Loading and saving results files.
- Manipulating gene sets.

In addition the following topics are available:

- How to locate a gene set
- How genes that occur more than once are handled.
- How multiple test correction works in ermineJ
- Some tricks for speeding up resampling
- An explanation of how gene set sizes are computed
- Using the ermineJ Application Programming Interface (API)
- The ermineJ command line interface (CLI)

ermineJ

By Paul Pavlidis, Homin Lee, William Braynen, Kiran Keshav.

Features of ermineJ

ErmineJ performs analyses of **gene sets** in **expression microarray data**. A typical goal is to determine whether particular biological pathways are "doing something interesting" in the data. The software is designed to be used by biologists with little or no informatics background.

Key features include:

- Implementation of several methods for gene set analysis:
    - Over-representation analysis
    - A resampling-based method that uses gene scores
    - A resampling-based method that uses correlation between gene expression profiles .
- Gene sets receive statistical scores (p-values), and multiple test correction is supported.
- Support of the Gene Ontology terminology
- User files use simple text formats.
- Users can modify gene sets or create new ones.
- The results can be visualized within the software and gene sets browsed as a tree or a table.
- It is simple to compare multiple analyses of the same data set with different settings.
- User-definable hyperlinks are provided to external sites to allow more efficient browsing of the results.
- For programmers, there is a simple application programming interface that can be used to plug ermineJ functionality into your own code

License information

Installation notes

WebStart

The simplest way to use ermineJ might be to use the webstart. This allows you to run ermineJ without running an installer.

Windows

Download the installer (ermineJ-XXXXX.exe), and double-click on it. The installation wizard will take you through the steps to install the software. After installation, ermineJ can be started by double-clicking on the desktop icon.

The only additional general information you should need is how the data directory is organized.

By default, ermineJ expects a data directory (folder) called ermineJ.data to be present in the installation directory (e.g., C:\Program Files\ermineJ\ermineJ.data). This directory is used to store information needed by ermineJ during analysis. This directory should be created during installation.

Custom gene sets created by the software go in the ermineJ/ermineJ.data/genesets folder. These are most easily created from within the ermineJ software, but you can add ones created externally by dropping them in ermineJ/ermineJ.data/genesets, or loading them from within ermineJ.

Other platforms

For other platforms, we provide a "generic bundle" that can be used to run ermineJ without an installer.

1. Unpack the distribution. You should end up with a directory called 'ermineJ-2.1' or something like that.
2. Set the enviroment variable ERMINEJ_HOME to this directory.
3. Add $ERMINEJ_HOME/bin to your path, if you want to make it easier to execute.
4. You can now execute ermineJ by running the script ermineJ.sh (Unix, should also work on Mac OSX) or ermineJ.bat (Windows) in the bin directory.

By default, executing the script will simply print usage instructions. To fire up the gui, use `ermineJ.sh -G`.

By using the other options, you can cause ermineJ to run an analysis non-interactively. The command line interface is described here

Using ermineJ for the first time

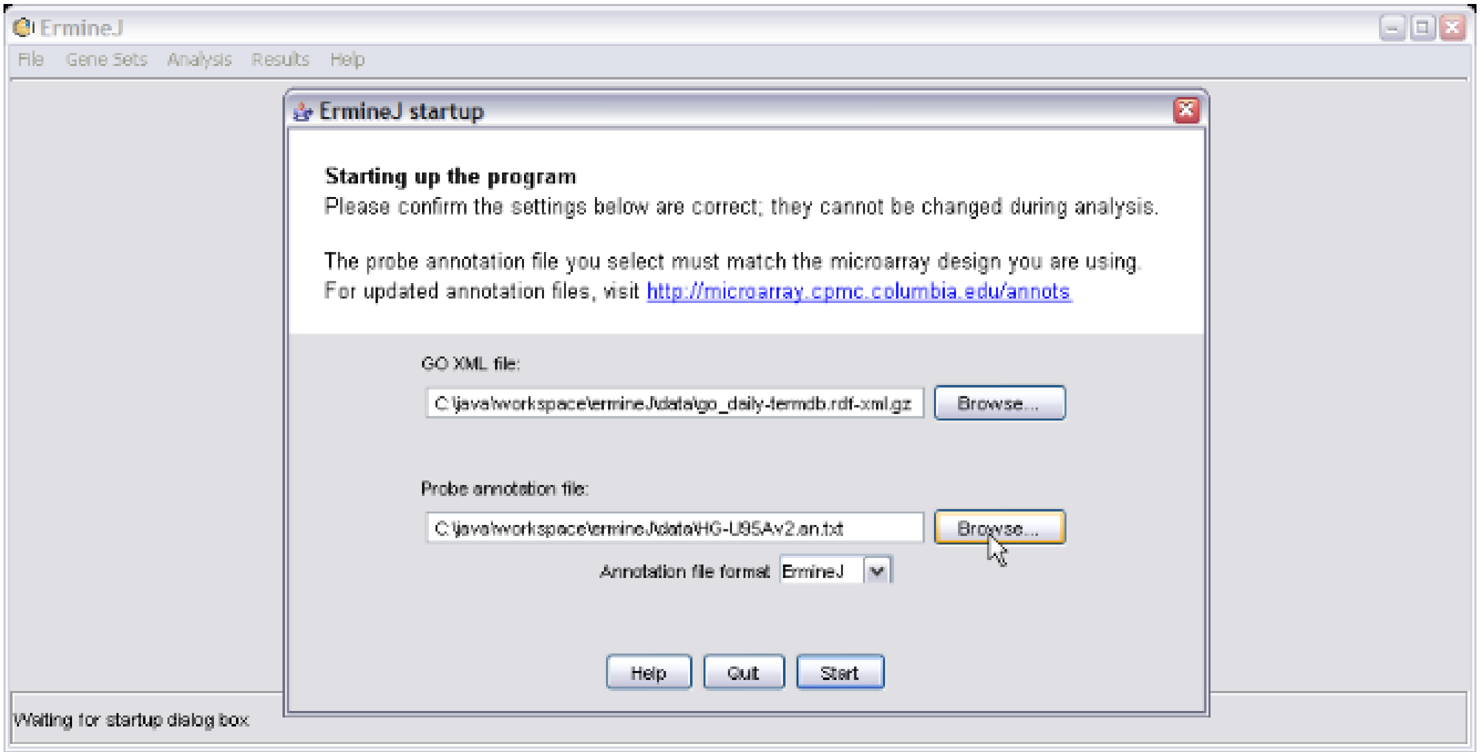
This page introduces what you will see when you first run ermineJ. Details on how to run an analysis, view the details of a gene set and other tasks are detailed elsewhere.

Before you begin you will need to get some files the software uses. First, you need a file that describes what gene sets are available. ErmineJ is set up to use the Gene Ontology; the necessary file is provided with the software but you may want to keep it up to date. The other file is one that describes the microarray design you are using: which of the Gene Ontology terms are actually represented on the microarray.

For more information on these input files, see the "Gene Ontology XML" topic and the "Gene Annotation" topic.

After you have obtained the files (unzipping them is not necessary), you can return to this page.

When you first start up the software, you will be presented with the following dialog box (Note that most of the screen shots in this manual have been scaled down to save space).



Starting up the program
Please confirm the settings below are correct; they cannot be changed during analysis.

The probe annotation file you select must match the microarray design you are using.
For updated annotation files, visit http://microarray.cpmc.columbia.edu/annots

GO XML file: C:\java\workspace\ermineJ\data\go_daily-termdb.rdf-xml.gz

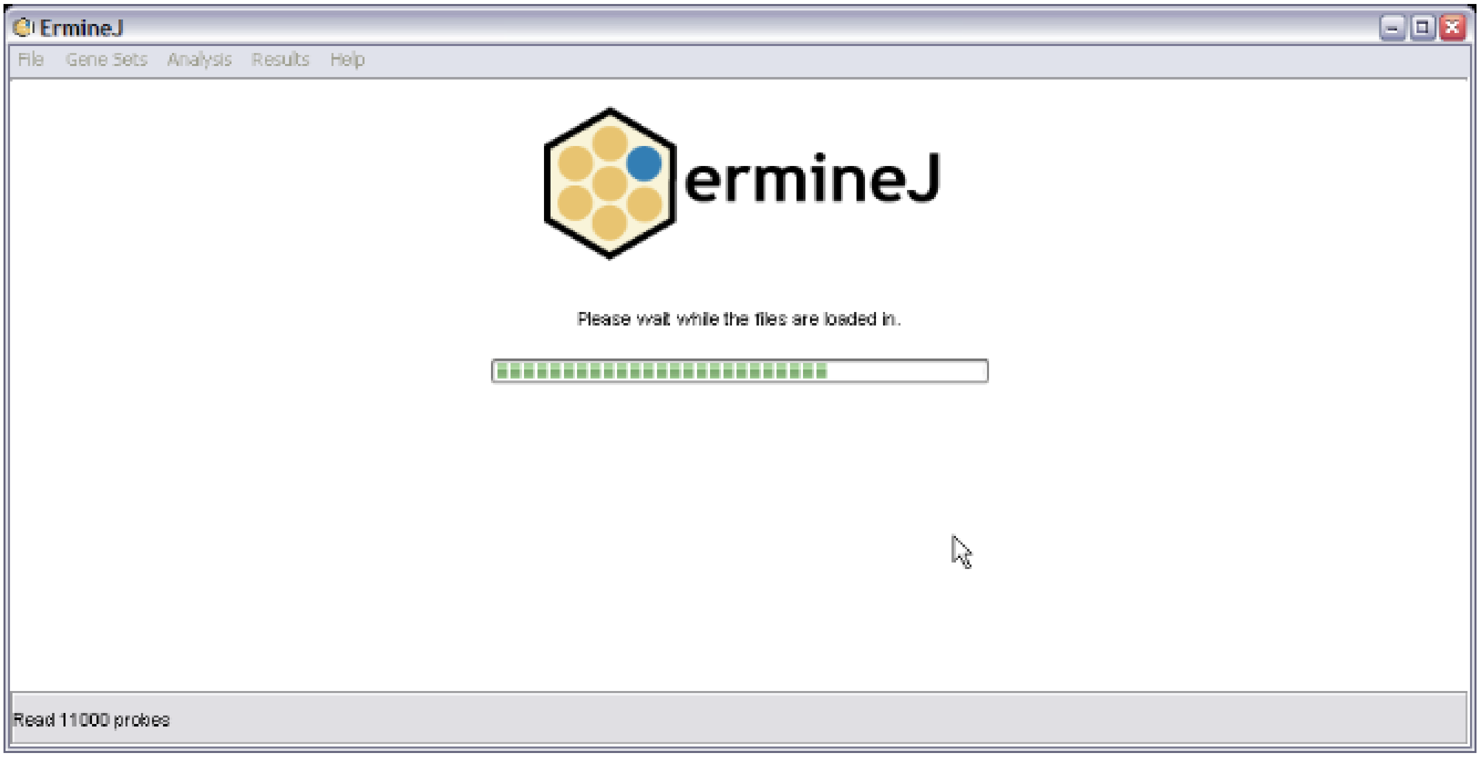
Probe annotation file: C:\java\workspace\ermineJ\data\HG-U95Av2.an.txt

Annotation file format: ErmineJ

Waiting for startup dialog box

Enter the GO XML file you are using, and the microarray annotation file, using the "Browse" buttons to locate them on your computer. Select the format of the annotation file from the pull-down menu. If you got the file from our web site, the format is "ErmineJ". The click on "Start".

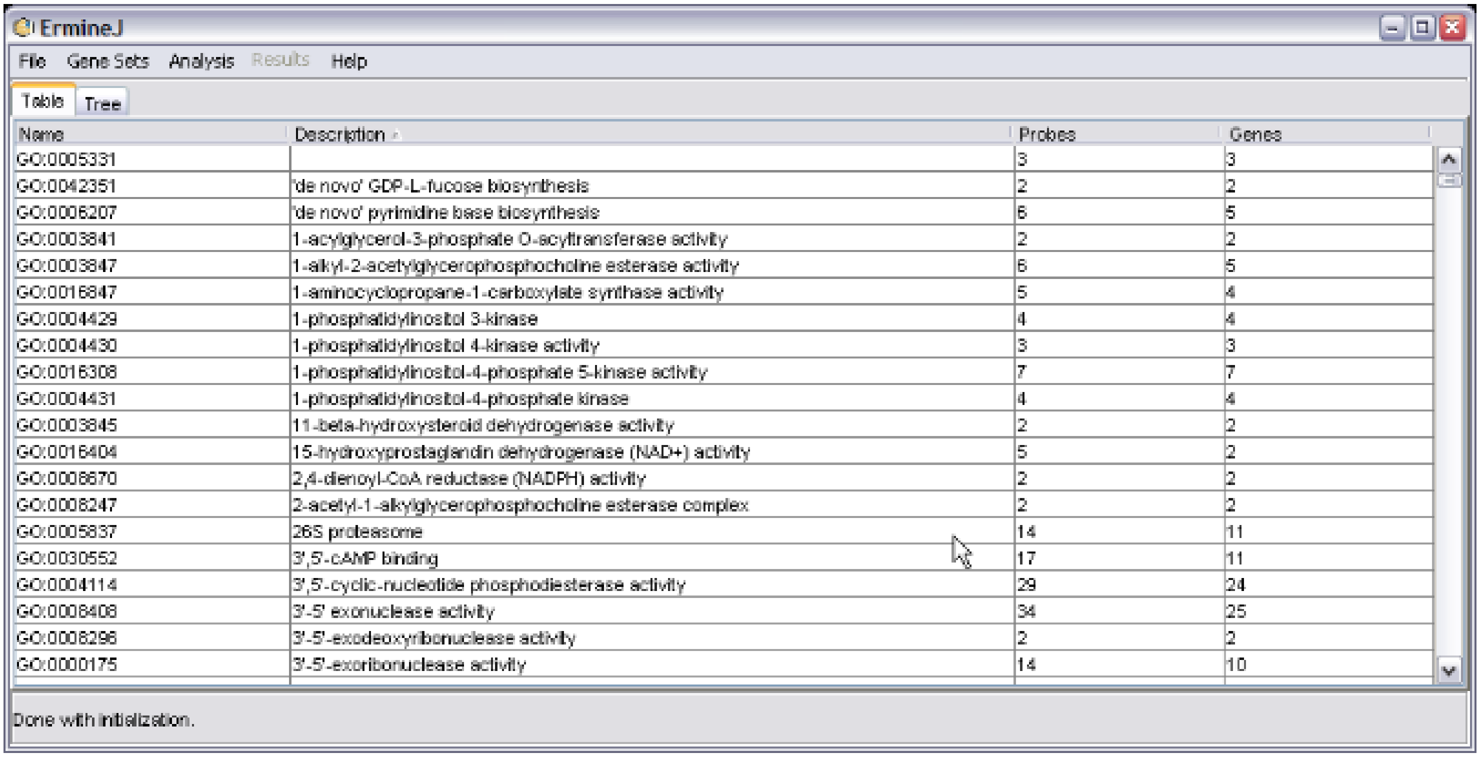
Then you will have a brief wait while the data files are loaded. These data are used for the remaining analysis and will only have to be loaded once. Here is what the loading screen looks like:



Please wait while the files are loaded in.

Read 11000 probes

Once loading is finished (~15-30 seconds) the following window is shown.



Name	Description	Probes	Genes
GO:0005331		3	3
GO:0042351	'de novo' GDP-L-fucose biosynthesis	2	2
GO:0006207	'de novo' pyrimidine base biosynthesis	6	5
GO:0003841	1-acylglycerol-3-phosphate O-acyltransferase activity	2	2
GO:0003847	1-alkyl-2-acetylglycerophosphocholine esterase activity	6	5
GO:0016847	1-aminocyclopropane-1-carboxylate synthase activity	5	4
GO:0004429	1-phosphatidylinositol 3-kinase	4	4
GO:0004430	1-phosphatidylinositol 4-kinase activity	3	3
GO:0016308	1-phosphatidylinositol-4-phosphate 5-kinase activity	7	7
GO:0004431	1-phosphatidylinositol-4-phosphate kinase	4	4
GO:0003845	11-beta-hydroxysteroid dehydrogenase activity	2	2
GO:0016404	15-hydroxyprostaglandin dehydrogenase (NAD+) activity	5	2
GO:0008670	2,4-dienoyl-CoA reductase (NADPH) activity	2	2
GO:0008247	2-acetyl-1-alkylglycerophosphocholine esterase complex	2	2
GO:0005837	26S proteasome	14	11
GO:0030552	3',5'-cAMP binding	17	11
GO:0004114	3',5'-cyclic-nucleotide phosphodiesterase activity	29	24
GO:0008408	3'-5' exonuclease activity	34	25
GO:0008296	3'-5'-exodeoxyribonuclease activity	2	2
GO:0000175	3'-5'-exoribonuclease activity	14	10

Done with initialization.

The three key parts of this window are:

- The "output panel" listing all the available gene sets either in a table or in a tree (see below)
- The menu bar, which is used to access analyses and other functions
- The "status bar" at the bottom of the window that is used to display pertinent information.

There are some additional features of the interface described below.

The Output panel always shows the four columns (and others, once you have done some analysis); clicking on a column header will re-sort the table according to the selected column.

The columns are:

- **Name** - the name or ID number of the gene set.
- **Description** - the description of the gene set
- **Probes** - how many probes on the array are in the gene set (We use the term "probe" to mean a "probe set" on Affymetrix arrays). This value is always the number of probes on the array design, even if your data set (which has not been entered yet) has filtered out some of the results. See this note .
- **Genes** - how many genes are in the gene set. This value is always less than or equal to the number of probes, because some genes are represented by more than one probe on the array. This value is always the number of genes represented in the array design, even if your data set (which has not been entered yet) has filtered out some of the results. See this note.

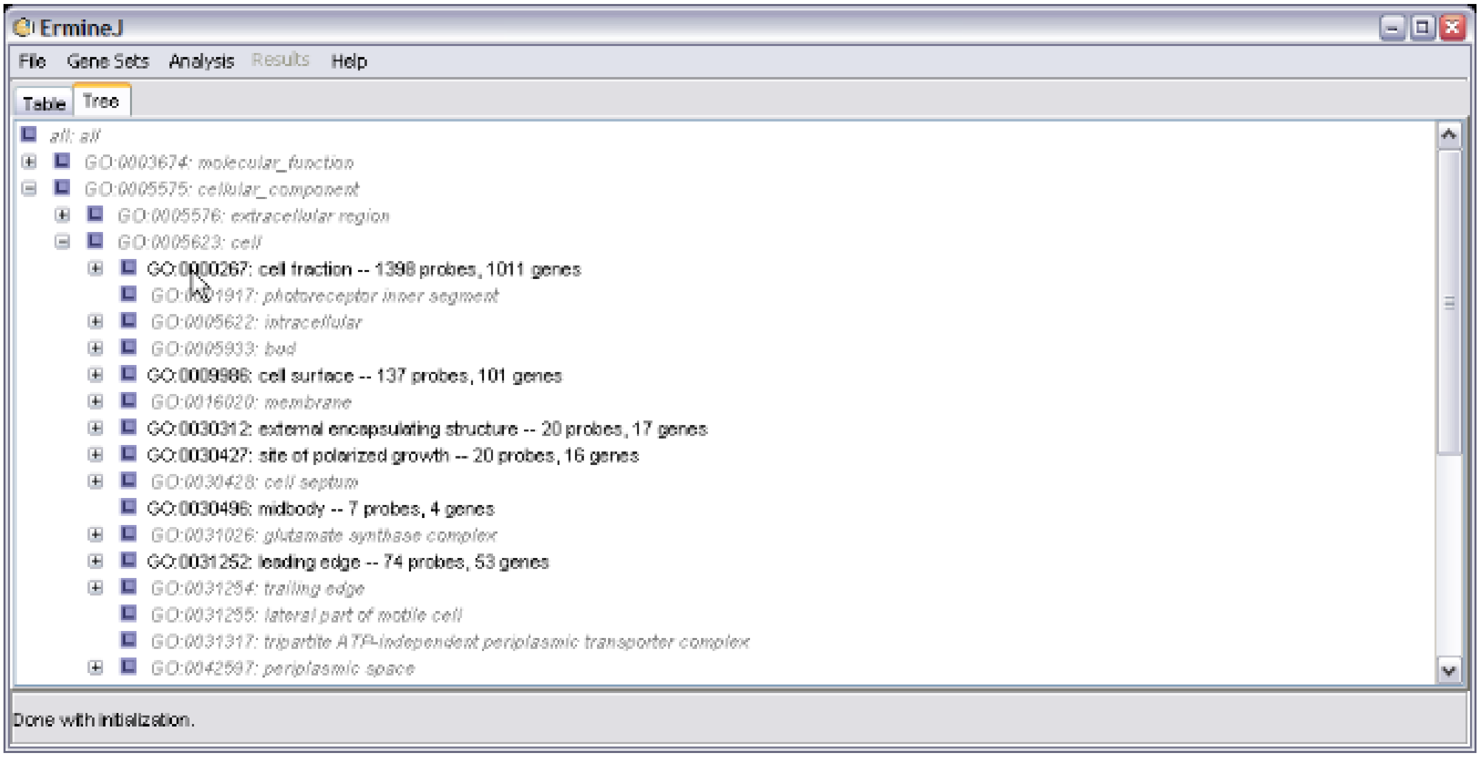
After an analysis has been done, additional columns will be added to this table. These are explained elsewhere.

The menus are:

- **File** - currently only used to quit the program.
- **Gene sets** - used to modify or create new gene sets as well as search for gene sets.
- **Analysis** - used to run analyses, load or save. Also to cancel an analysis while it is running.
- **Results** - Used to switch the results set shown in the tree view. It will be inactive if you are viewing the table view or have done fewer than 2 analyses.
- **Help** - used to reach these pages and view credits.

There are shortcut keys for many of the menu items.

The "tree" tab switches you to a view of the gene sets in a hierarchy. This view is linked to the table view in many ways. You can use either or both views, depending on what you are doing and which you prefer.



```
all: all
⊞ GO:0003674: molecular_function
⊟ GO:0005575: cellular_component
    ⊞ GO:0005576: extracellular region
    ⊟ GO:0005623: cell
        ⊞ GO:0000267: cell fraction -- 1398 probes, 1011 genes
          GO:0001917: photoreceptor inner segment
        ⊞ GO:0005622: intracellular
        ⊞ GO:0005933: bud
        ⊞ GO:0009986: cell surface -- 137 probes, 101 genes
        ⊞ GO:0016020: membrane
        ⊞ GO:0030312: external encapsulating structure -- 20 probes, 17 genes
        ⊞ GO:0030427: site of polarized growth -- 20 probes, 16 genes
        ⊞ GO:0030428: cell septum
          GO:0030496: midbody -- 7 probes, 4 genes
        ⊞ GO:0031026: glutamate synthase complex
        ⊞ GO:0031252: leading edge -- 74 probes, 53 genes
        ⊞ GO:0031254: trailing edge
          GO:0031255: lateral part of motile cell
          GO:0031317: tripartite ATP-independent periplasmic transporter complex
        ⊞ GO:0042597: periplasmic space
```

Done with initialization.

Additional functionality of the main panel

In addition to the obvious features shown above, there are other functionalities of the main panel:

Mouse button actions

- Double-clicking on a gene set in either the table or tree brings up the details view for that gene set.
- On both the tree and table view, there is a "popup" or "context-specific" menu (right-click on windows, option-click on Mac) for gene sets that makes it easier to use common functions and search the Gene Ontology web site. There is a different popup menu for different parts of the table and some functions are only available from the tree view or the table view.
- Clicking on the table headers resorts the table.

Tooltips

There are **"tooltips"** that are used to display extra information that would otherwise clutter up the interface. Hover the mouse over a field in the table or a line in the tree to see this. Specifically:

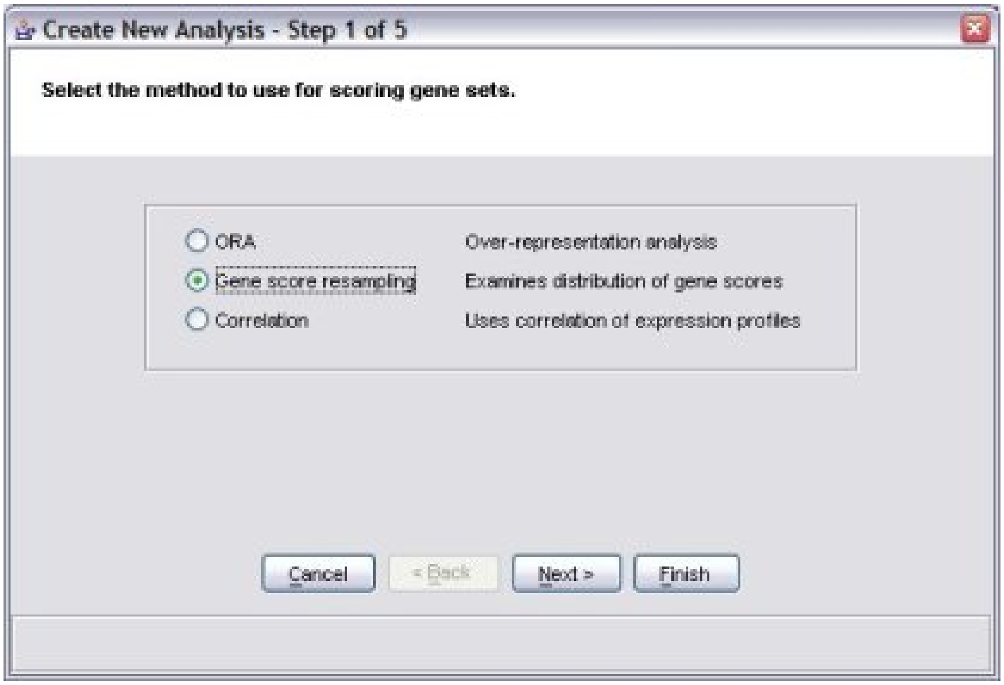
- On both the tree and table views, hovering over a gene set ID or name brings up a tooltip that shows the GO aspect and definition (if any).
- In the table view, hovering over a result p-value shows detailed statistics (when you first start the software there are no results columns, so this will make more sense once you do).
- In the table view, hovering over the header of a results column shows a summary of the settings that were used

ermineJ

Running an analysis and using the output panel

To run an analysis, choose "run analysis" from the analysis menu.

The first screen you are presented with is this one:



Create New Analysis - Step 1 of 5

Select the method to use for scoring gene sets.

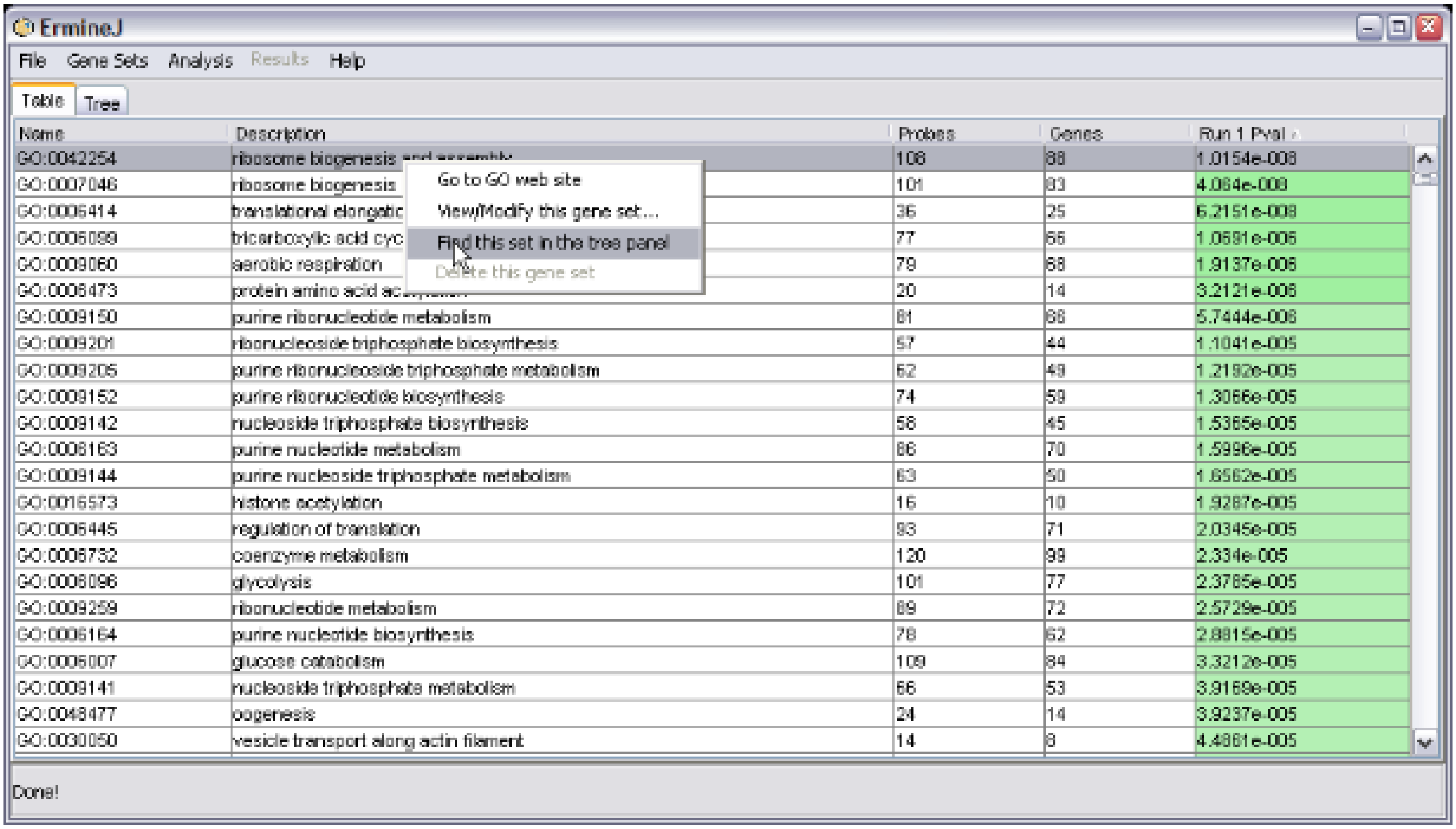
○ ORA	Over-representation analysis
● Gene score resampling	Examines distribution of gene scores
○ Correlation	Uses correlation of expression profiles

Choose the type of analysis you want to do. Directions on the specific types of analysis are available:

- ORA - start here to read general information about conducting an analysis.
- Gene score resampling
- Correlation

(the rest of this tutorial assumes you have run some analysis using the directions)

When you are done with an analysis, the output panel will show the results. Here is the output panel after an analysis. Note that the "Probes" and "Genes" columns show how many probe and genes are on the array design, not necessarily the number represented in your data. Please read this note for more explanation. To get the number used in the analysis, hover your the mouse pointer over the results column and read the sizes from the tool tip.



Name	Description	Probes	Genes	Run 1 Pval
GO:0042254	ribosome biogenesis and assembly	108	88	1.0154e-008
GO:0007046	ribosome biogenesis	101	83	4.084e-008
GO:0006414	translational elongatic	36	25	6.2151e-008
GO:0006099	tricarboxylic acid cyc	77	66	1.0691e-006
GO:0009060	aerobic respiration	79	68	1.9137e-006
GO:0006473	protein amino acid ac	20	14	3.2121e-006
GO:0009150	purine ribonucleotide metabolism	81	66	5.7444e-006
GO:0009201	ribonucleoside triphosphate biosynthesis	57	44	1.1041e-005
GO:0009205	purine ribonucleoside triphosphate metabolism	62	49	1.2192e-005
GO:0009152	purine ribonucleotide biosynthesis	74	59	1.3066e-005
GO:0009142	nucleoside triphosphate biosynthesis	58	45	1.5365e-005
GO:0006163	purine nucleotide metabolism	86	70	1.5996e-005
GO:0009144	purine nucleoside triphosphate metabolism	63	50	1.6562e-005
GO:0016573	histone acetylation	16	10	1.9287e-005
GO:0006445	regulation of translation	93	71	2.0345e-005
GO:0006732	coenzyme metabolism	120	99	2.334e-005
GO:0006096	glycolysis	101	77	2.3785e-005
GO:0009259	ribonucleotide metabolism	89	72	2.5729e-005
GO:0006164	purine nucleotide biosynthesis	78	62	2.8815e-005
GO:0006007	glucose catabolism	109	84	3.3212e-005
GO:0009141	nucleoside triphosphate metabolism	66	53	3.9169e-005
GO:0048477	oogenesis	24	14	3.9237e-005
GO:0030050	vesicle transport along actin filament	14	8	4.4881e-005

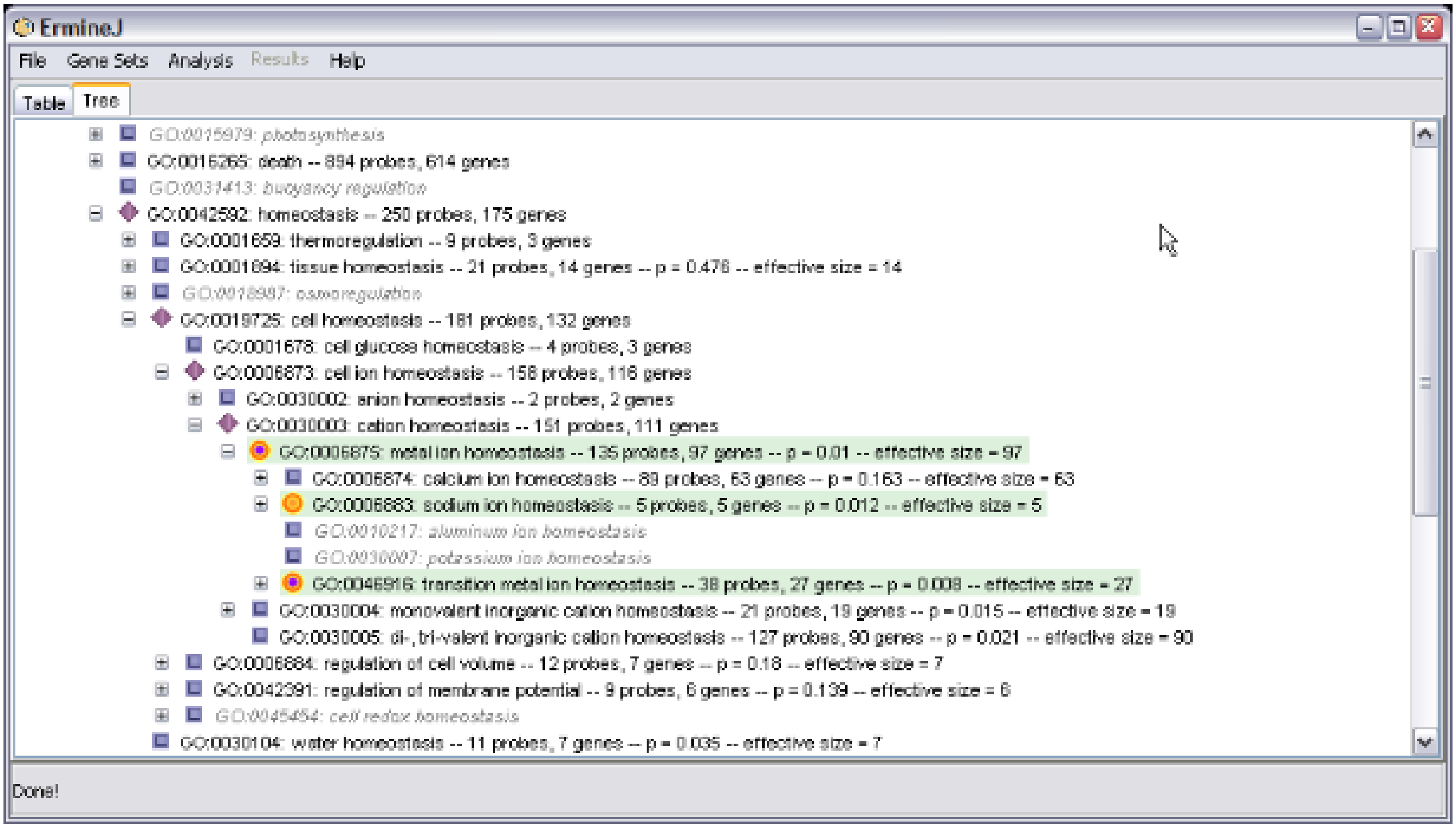
The results of each run are shown in columns after the standard four. Each value is the p value associated with the gene set for that run. Rows that have "good" pvalues are shown in color. (the specific meaning of the colors is subject to change: currently only gene sets that meet a FDR of 0.1 are in color.)

If you hold the mouse over a p-value for a gene set, a tool tip will be displayed that shows additional information, including the correct number of probes and genes for the gene set as represented in the data you used.

The colors indicate classes that passed a multiple test correction. Read details here.

The tree panel

Selecting the tree panel lets you view the results of the analysis in another way:



```
    GO:0015979: photosynthesis
    GO:0016265: death -- 894 probes, 614 genes
    GO:0031413: buoyancy regulation
    GO:0042592: homeostasis -- 250 probes, 175 genes
      GO:0001659: thermoregulation -- 9 probes, 3 genes
      GO:0001894: tissue homeostasis -- 21 probes, 14 genes -- p = 0.476 -- effective size = 14
      GO:0018987: osmoregulation
      GO:0019725: cell homeostasis -- 181 probes, 132 genes
        GO:0001678: cell glucose homeostasis -- 4 probes, 3 genes
        GO:0006873: cell ion homeostasis -- 158 probes, 116 genes
          GO:0030002: anion homeostasis -- 2 probes, 2 genes
          GO:0030003: cation homeostasis -- 151 probes, 111 genes
            GO:0006875: metal ion homeostasis -- 135 probes, 97 genes -- p = 0.01 -- effective size = 97
              GO:0006874: calcium ion homeostasis -- 89 probes, 63 genes -- p = 0.163 -- effective size = 63
              GO:0006883: sodium ion homeostasis -- 5 probes, 5 genes -- p = 0.012 -- effective size = 5
              GO:0010217: aluminum ion homeostasis
              GO:0030007: potassium ion homeostasis
              GO:0046916: transition metal ion homeostasis -- 38 probes, 27 genes -- p = 0.008 -- effective size = 27
            GO:0030004: monovalent inorganic cation homeostasis -- 21 probes, 19 genes -- p = 0.015 -- effective size = 19
            GO:0030005: di-, tri-valent inorganic cation homeostasis -- 127 probes, 90 genes -- p = 0.021 -- effective size = 90
        GO:0006884: regulation of cell volume -- 12 probes, 7 genes -- p = 0.18 -- effective size = 7
        GO:0042391: regulation of membrane potential -- 9 probes, 6 genes -- p = 0.139 -- effective size = 6
        GO:0045454: cell redox homeostasis
      GO:0030104: water homeostasis -- 11 probes, 7 genes -- p = 0.035 -- effective size = 7
```

Done!

Gene sets that appear in grey are not represented on your microarray design, or have been excluded by a search.

The green color-coding has the same meaning as for the table panel - it indicates gene sets that have "good" pvalues. (the specific meaning of the colors is subject to change: currently only gene sets that meet a FDR of 0.1 are in color.)

The icons also have specific meaning to help you find interesting reuslts more quickly (Note: the icons may change in future versions of the software. This description is meant to be general:

- Yellow spots indicate that the gene set has a "good" pvalue (it would be shown in color).
- Purple diamonds indicate that the gene set has a child that has a "good" pvalue.
- Yellow spots with a purple center indicate that the gene set has a good pvalue AND it has a child that has a good pvalue.
- Blue squares indicate a gene set that has no children with good pvalues.

Selecting the analysis in the tree view

After an analysis is run, the results of the new analysis are shown in the tree. To switch to a different analysis, use the "Results" menu and select the run you would like to view. If you have done fewer than two analyses, the "Results" menu will be greyed out.

Deleting a run

You can **delete a run by right-clicking on its column heading** and selecting "delete this run". **Note**: A known issue is that the table will resort itself after doing this; just click on a header to resort it the way you want.

Renaming a run

You can **Rename a run** by right-clicking on its column heading and selecting "rename this run". This can be convenient if you have a lot of runs.

Viewing the details of a gene set

You can **double-click on a row** to get the details for a gene set. For example, to view the details for run 2, "response to hypoxia", click on the cell in the "Run 2 Pval" column on the "response to hypoxia" row. The window you get is explained here.

View information about a gene set on the GO web site

By **right-clicking on a row**, you will get a pop-up menu. The first item on this menu should open a web browser showing the Amigo web page for the gene set selected.

The other menu item allows you to view or modify the gene set.

ermineJ

Tutorial: General analysis steps and Over-representation analysis (ORA)

We suggest that you read this page first, as it explains some aspects of analysis that are common to all the methods.

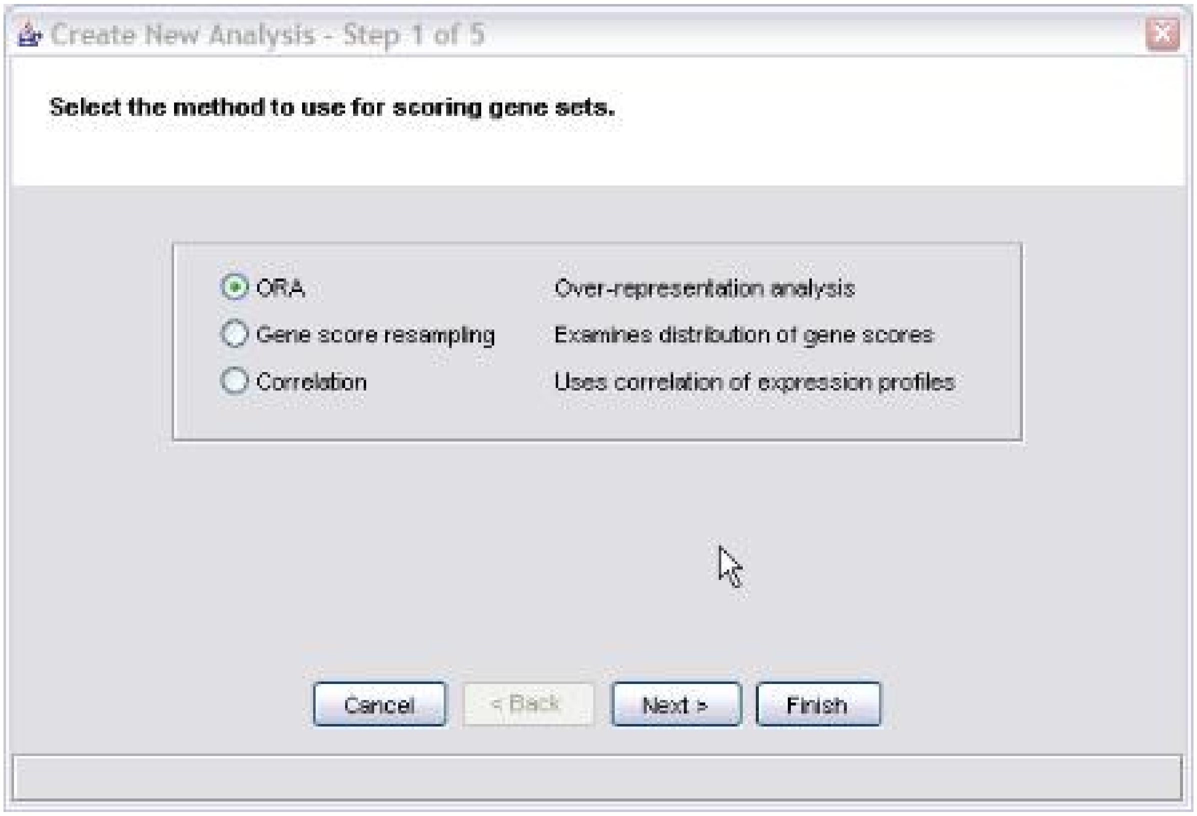
Overview of the method

Over-representation analysis (ORA) examines the genes that meet a selection criterion and determines if there are gene sets which are statistically over-represented. This method differs from the resampling-based analysis in that you must set a gene score threshold for gene selection. The software does not support entering a list of selected genes. Instead, you input the list of all genes along with their scores, and set the threshold.

The probabilities produced by ermineJ ORA are computed using the binomal approximation to the hypergeometric distribution.

Walkthrough

Step 1 is to choose the type of analysis. We select ORA:



Create New Analysis - Step 1 of 5

Select the method to use for scoring gene sets.

● ORA — Over-representation analysis

○ Gene score resampling — Examines distribution of gene scores

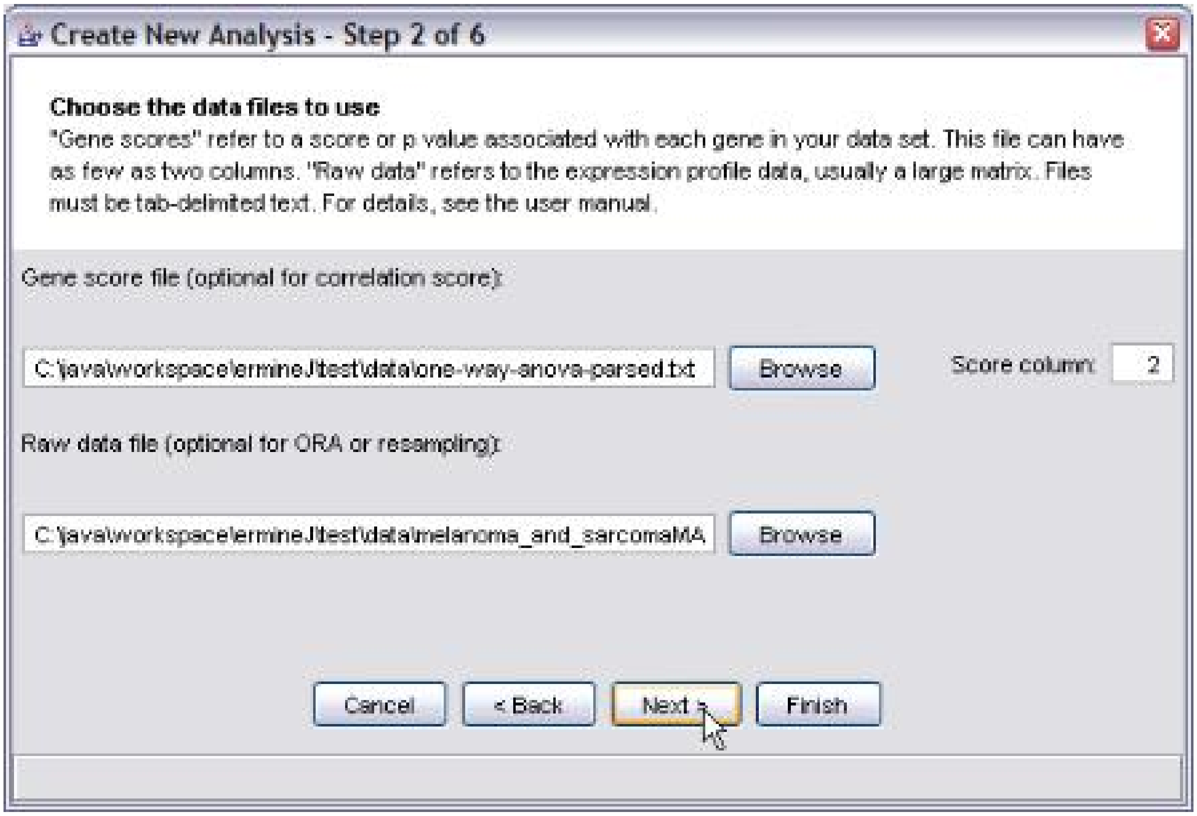
○ Correlation — Uses correlation of expression profiles

Step 2: choose input file(s)

The next window is common to all analysis methods. Two data files are requested. For ORA, only the Gene Score File is required. However, entering the raw data file will allow you to visualize the results later.

The Gene Score file format is explained here. In this panel, you must also select the score column. The first column in the gene score file contains the probe identifiers; the gene scores themselves are in the second or higher column. If your gene score file only has two columns, just use the default value of 2.

The Raw data file format is explained here .



Create New Analysis - Step 2 of 6

Choose the data files to use

"Gene scores" refer to a score or p value associated with each gene in your data set. This file can have as few as two columns. "Raw data" refers to the expression profile data, usually a large matrix. Files must be tab-delimited text. For details, see the user manual.

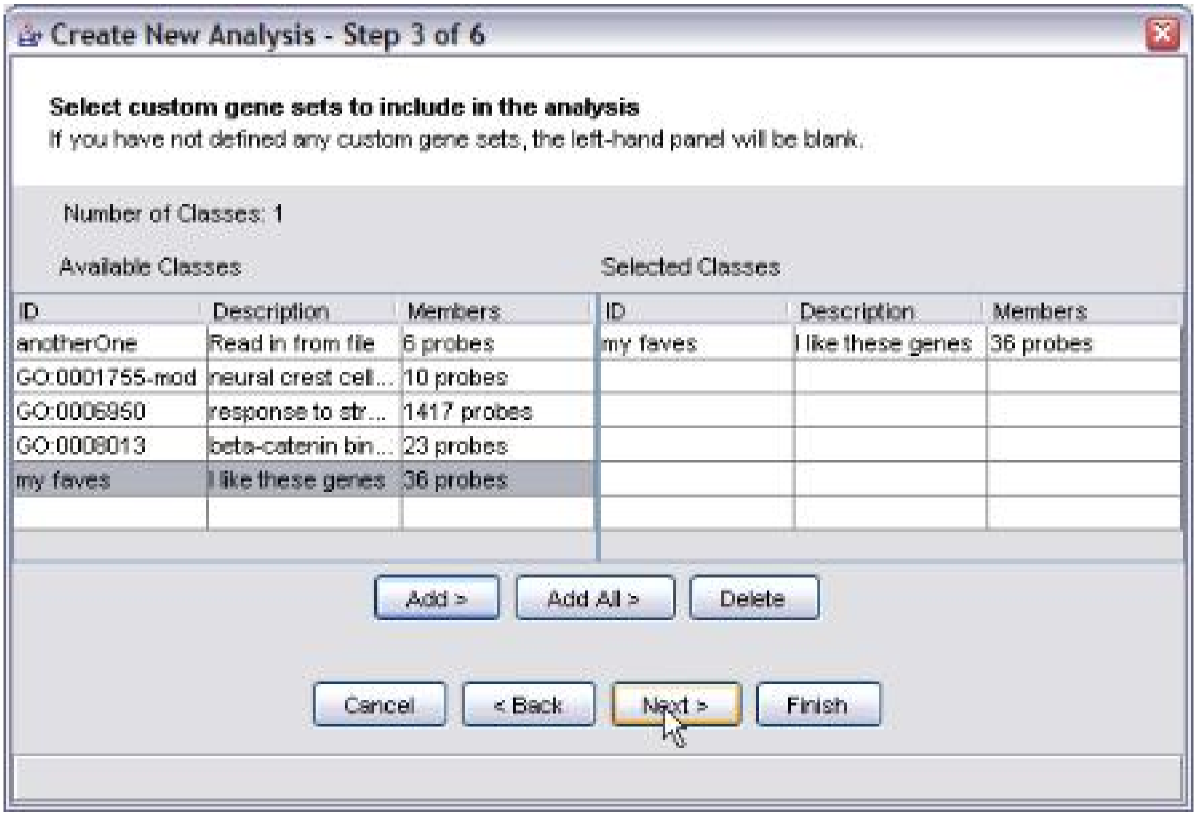
Gene score file (optional for correlation score): C:\java\workspace\ermineJ\test\data\one-way-anova-parsed.txt

Score column: 2

Raw data file (optional for ORA or resampling): C:\java\workspace\ermineJ\test\data\melanoma_and_sarcomaMA

Step 3: Choose custom gene sets to include

This, also common to all analysis methods, asks you to add any custom gene sets you may have defined.



Create New Analysis - Step 3 of 6

Select custom gene sets to include in the analysis
If you have not defined any custom gene sets, the left-hand panel will be blank.

Number of Classes: 1

Available Classes

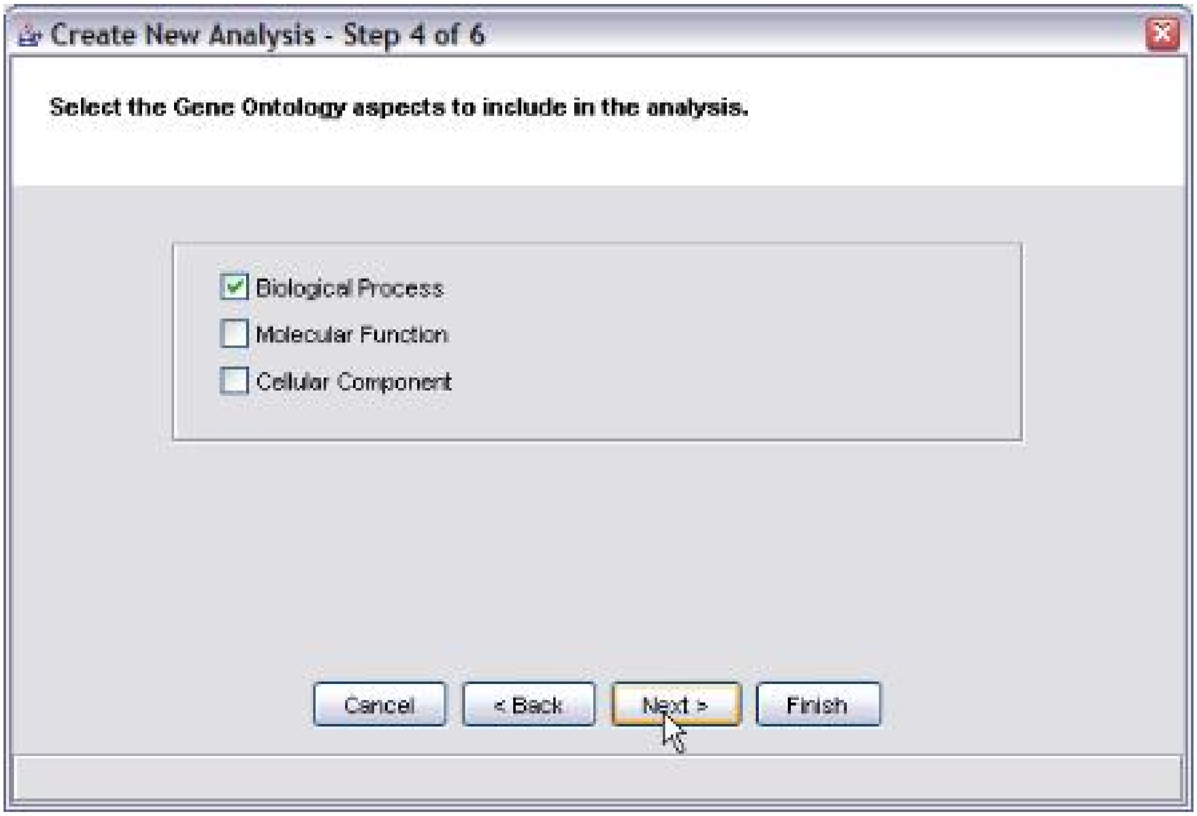
ID	Description	Members
anotherOne	Read in from file	6 probes
GO:0001755-mod	neural crest cell...	10 probes
GO:0006950	response to str...	1417 probes
GO:0008013	beta-catenin bin...	23 probes
my faves	I like these genes	36 probes

Selected Classes

ID	Description	Members
my faves	I like these genes	36 probes

Step 4: Choose GO aspects

Step 4 is also common to all the methods. Select the GO aspects you want to include in the analysis.



Create New Analysis - Step 4 of 6

Select the Gene Ontology aspects to include in the analysis.

☑ Biological Process
□ Molecular Function
□ Cellular Component

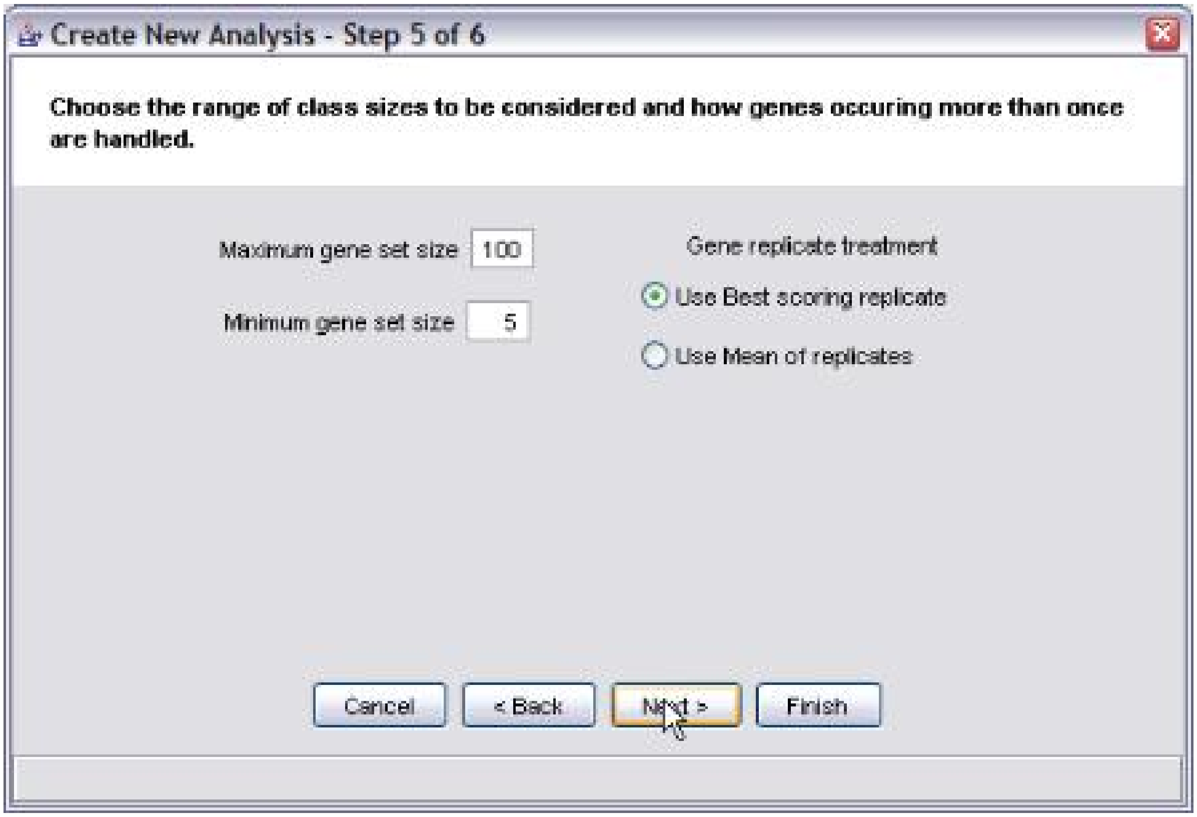
Step 5 is also common to all the methods.

The **maximum and minimum gene set sizes** determine the range of gene set sizes that will be considered. We recommend avoiding the use of very small or very large gene sets. There are several reasons to avoid using extremes:

- The more gene sets you examine, the worse the multiple testing issue.
- Using more gene sets increases computation time for the resamping-based methods (not an issue for ORA).
- Very large or very small gene sets are less informative:
  - Very large gene sets are rather non-specific and tend not to tell you as much.
- Very small gene sets defeat the purpose of examining genes in groups.

The "**Gene replicate treatment** " refers to what is done when a gene occurs more than once in the dataset.

There are two options, "Best" and "Mean". The application of these methods differs for the different methods. There is more information here



Create New Analysis - Step 5 of 6

Choose the range of class sizes to be considered and how genes occuring more than once are handled.

Maximum gene set size: 100

Minimum gene set size: 5

Gene replicate treatment

● Use Best scoring replicate

○ Use Mean of replicates

Step 6: ORA-specific settings

This step is specific to the ORA analysis.

The gene score threshold determines how genes are "selected". Genes that meet the threshold requirement are considered "good". ErmineJ operates differently from some other available tools in that you do not enter the "good" genes yourself: the software selects them based on the criteria you set here.

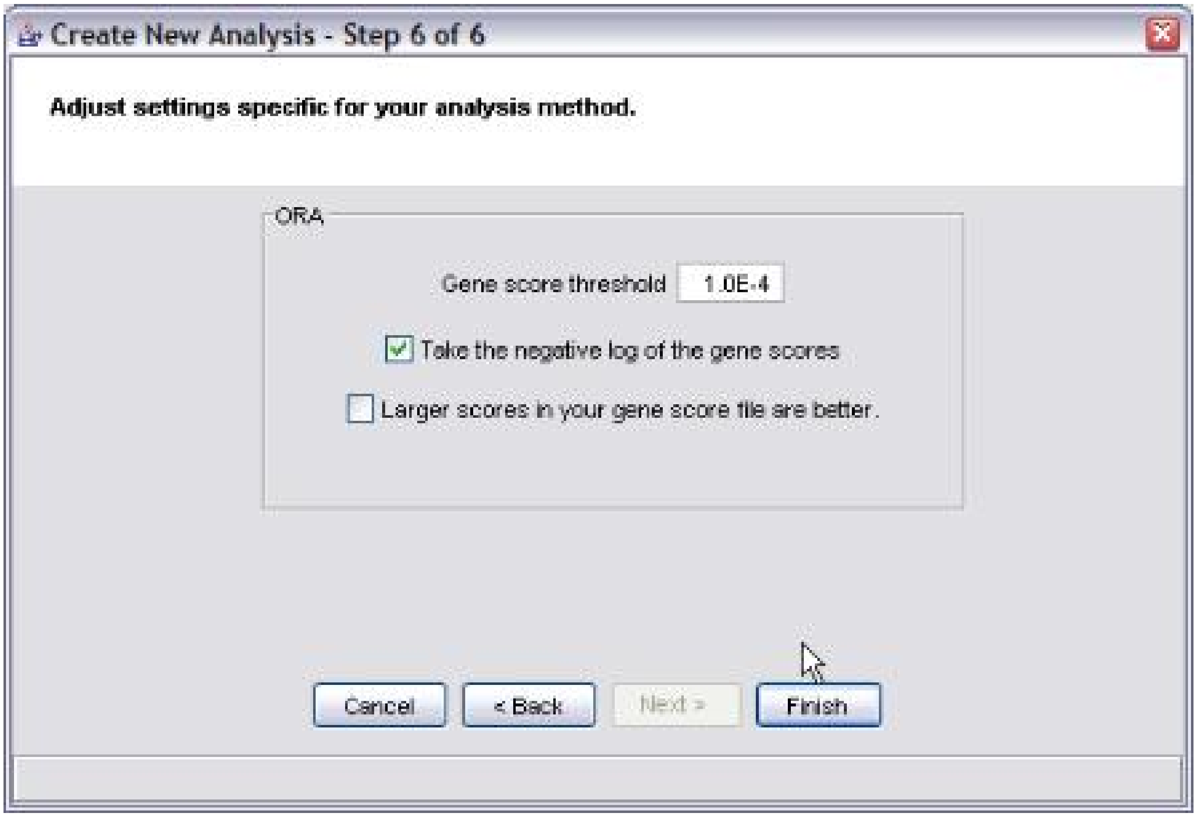
Tip: If you are using raw p values as your gene scores, make sure your threshold is a value between 0 and 1 (e.g., 0.0001), check the "log transform" box, and leave the "larger scores are better" box unchecked. This is because the "larger is better" choice relates to the original threshold, not the log-tranformed threshold. On the other hand, if your p-values are already -log-transformed, you should use the exact opposite settings.

Log transformation

If you negative log-transform the gene scores, then your input gene scores are transformed according to the function f(x) = $-log_{10}$(x). This option is provided as a convenience as the most common type of gene score is a p-value. The transform f(x) puts the p values on a more useful scale. However, you should still leave the "larger scores are better" box unchecked, as this refers to your original data.

Are larger scores better?

If you are using fold change as your gene scores, you may want to check the "larger scores are better" box. This assumes that you have taken the absolute value of the fold change values before entering them into ermineJ. That way, changes up and down will be considered equally. Alternatively, you could focus on changes up or down by retaining the sign on the fold change values and setting this option depending on which direction of change you are interested in analyzing.



Create New Analysis - Step 6 of 6

Adjust settings specific for your analysis method.

ORA

Gene score threshold 1.0E-4

☑ Take the negative log of the gene scores

□ Larger scores in your gene score file are better.

After hitting finish, you will rapidly get a new result set in the output (results) table. This is explained here

ermineJ

Tutorial: Gene score resampling (GSR)

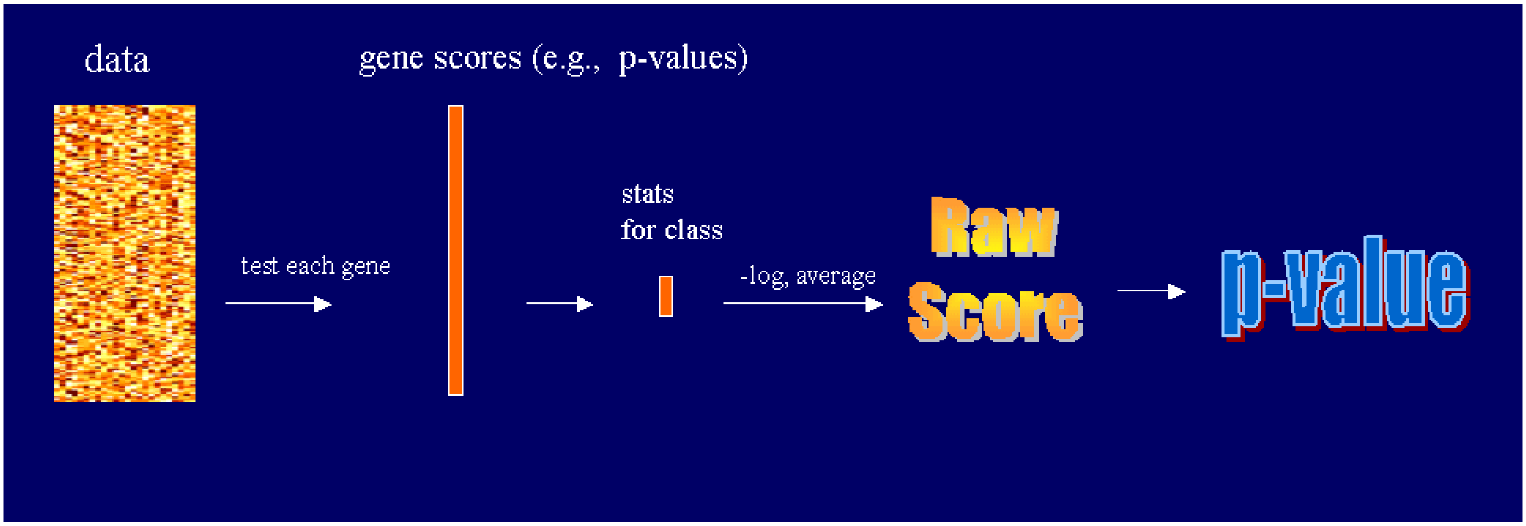
Overview of the method

The goal of this method is the same as for ORA: to provide a p value for each gene set. The key difference lies in that ORA requires that you select a threshold for "gene selection", whereas GSR does not.

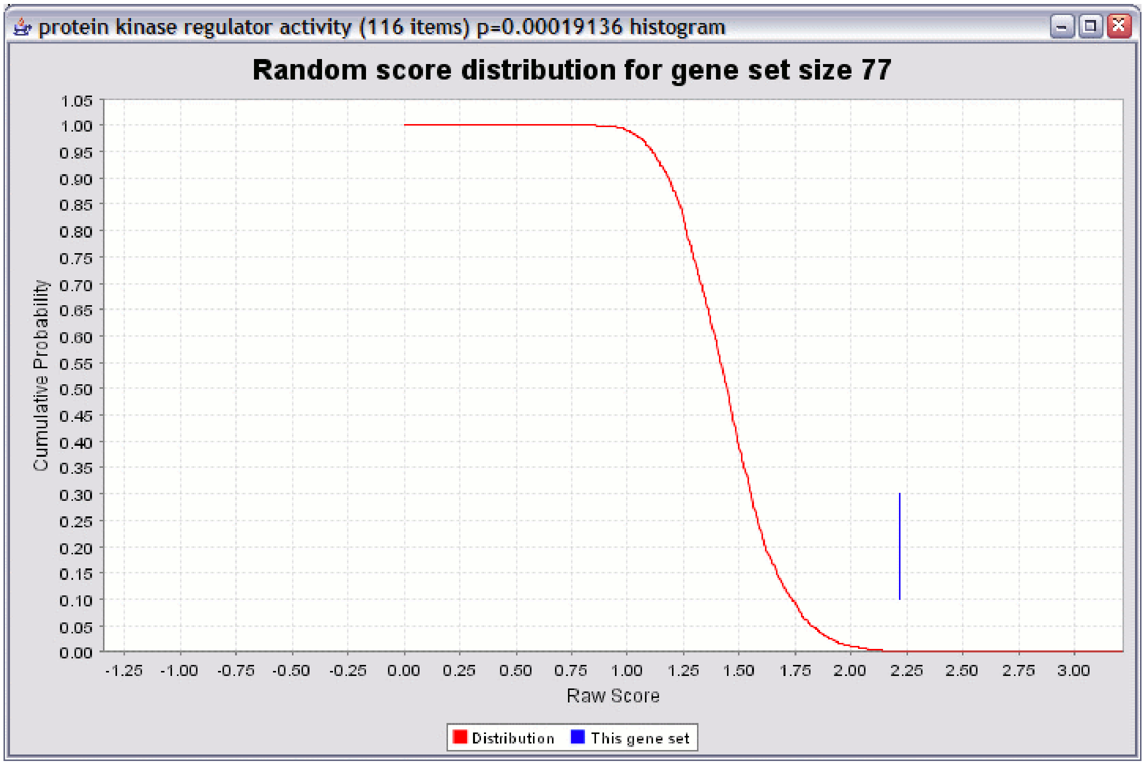
GSR uses all the gene scores for the genes in a gene set to produce a score. This means that genes that do not meet a statistical threshold for selection can contribute to the score. In addition, more information contained in the gene scores is preserved than in ORA, because ORA is essentially rank-based, whereas GSR uses the gene scores themselves.

In practice, ORA and GSR can yield similar results; however, we have found that GSR tends to be more robust than ORA (because there is no threshold to set) and can give interesting results in situations where ORA doesn't work as well (when no genes meet the predetermined selection threshold).

A high-level overview of the procedure is depicted below.



An illustration of the distribution obtained by resampling, and how a p value is computed, is shown below:



The red line is a cumulative probability distribution for the randomly sampled gene sets. You can see that raw scores over 2.00 are quite unlikely. In this example the gene set "protein kinase regulator activity" had a raw score of about 2.2. This yields a p value of about 0.00019 (the height of the red graph at that point). If the gene set had a score of 1.5, the p value would be poor (about 0.4).

Walk-through

The steps for the resampling analysis are the same as for ORA, except for the last step.

As for ORA, you must decide whether to take the negative logarithm of the gene scores or not, and whether "larger" genes scores are considered "good".

Tip: If you are using raw p values as your gene scores, check the "log transform" box, and leave the "larger scores are better" box unchecked. This is because the "larger is better" choice relates to the original threshold, not the log-tranformed threshold. On the other hand, if your p-values are already -log-transformed, you should use the exact opposite settings.

Log transformation

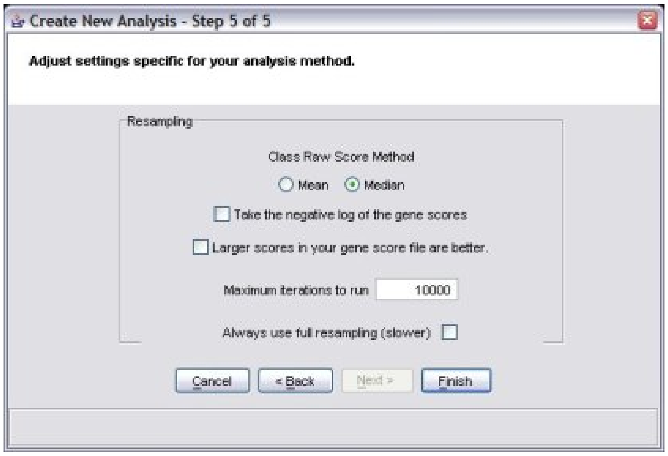
If you negative log-transform the gene scores, then your input gene scores are transformed according to the function f(x) = $-log_{10}(x)$. This option is provided as a convenience as the most common type of gene score is a p-value. The transform f(x) puts the p values on a more useful scale. However, you should still leave the "larger scores are better" box unchecked, as this refers to your original data.

Are larger scores better?

If you are using fold change as your gene scores, you may want to check the "larger scores are better" box. This assumes that you have taken the absolute value of the fold change values before entering them into ermineJ. That way, changes up and down will be considered equally. Alternatively, you could focus on changes up or down by retaining the sign on the fold change values and setting this option depending on which direction of change you are interested in analyzing.

How many iterations?

Unlike ORA, you must choose how many iterations to run. To speed things up, you can uncheck the "Always use full resampling" checkbox, which enables some approximations . Alternatively, we suggest a starting value of 10,000 iterations. When you decide on parameters you like, we recommend a larger number of iterations (perhaps 200,000 or more). This is to get sufficient precision in the p-values to make multiple-test correction work correctly.



Create New Analysis - Step 5 of 5

Adjust settings specific for your analysis method.

Resampling

Class Raw Score Method

○ Mean ● Median

□ Take the negative log of the gene scores

□ Larger scores in your gene score file are better.

Maximum iterations to run 10000

Always use full resampling (slower) □

After pressing the "finish" button, the analysis will run. Unlike ORA, the results will take a short while to compute. As the

---

ermineJ

Tutorial: Correlation resampling

Method Overview

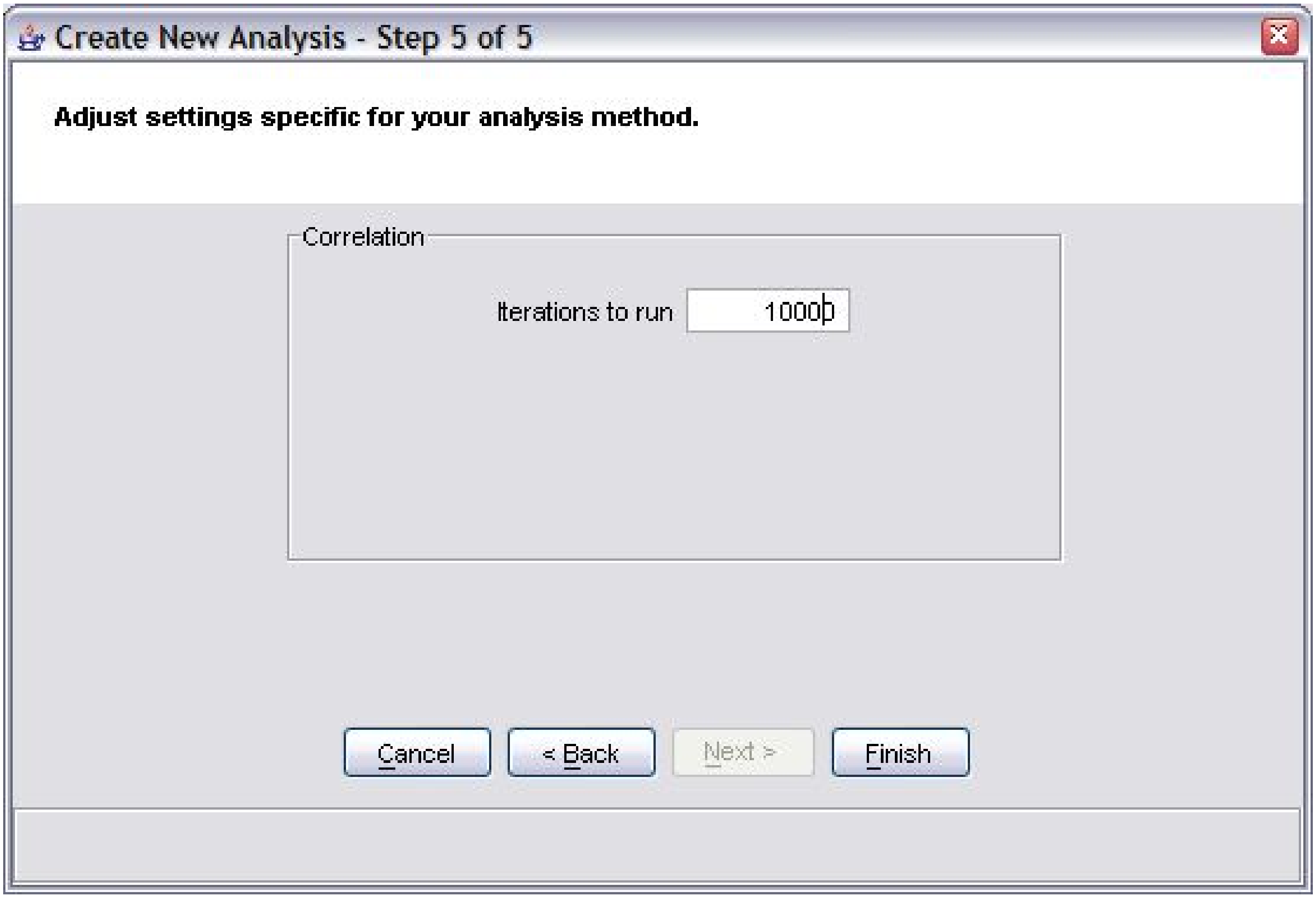
Unlike ORA and Gene Score Resampling, this method examines the gene expression profiles themselves, not the gene scores for each gene. A score is computed for a gene set based on how correlated the expression profiles are.

This can be thought of as a measure of how well the genes in the set cluster together, but they need not all be in the same cluster. Thus a gene set that contains two coherent clusters that encompass most of the genes in the set will tend to get a good score (though not as good as a gene set that is just one big cluster).

Correlation resampling is the most computationally intensive method implemented by ermineJ. For this reason we recommend setting the number of iterations lower, perhaps only 10,000. In addition, running larger class sizes takes longer than smaller ones, a consideration you might use when setting the parameters.

Walkthrough

If you have read the ORA page (please do), you will recognize the first 5 steps of the wizard. The only different one is the last step:



Create New Analysis - Step 5 of 5

Adjust settings specific for your analysis method.

Correlation

Iterations to run: 10000

Note that the "gene replicate treatment" choice in step 5 does not apply to correlation analysis at this time. All correlation analyses use the "mean" method of weighting multiple comparisons among genes. Comparisons of a gene to itself are always skipped. That is, if there are two probes for a gene, they are not compared.

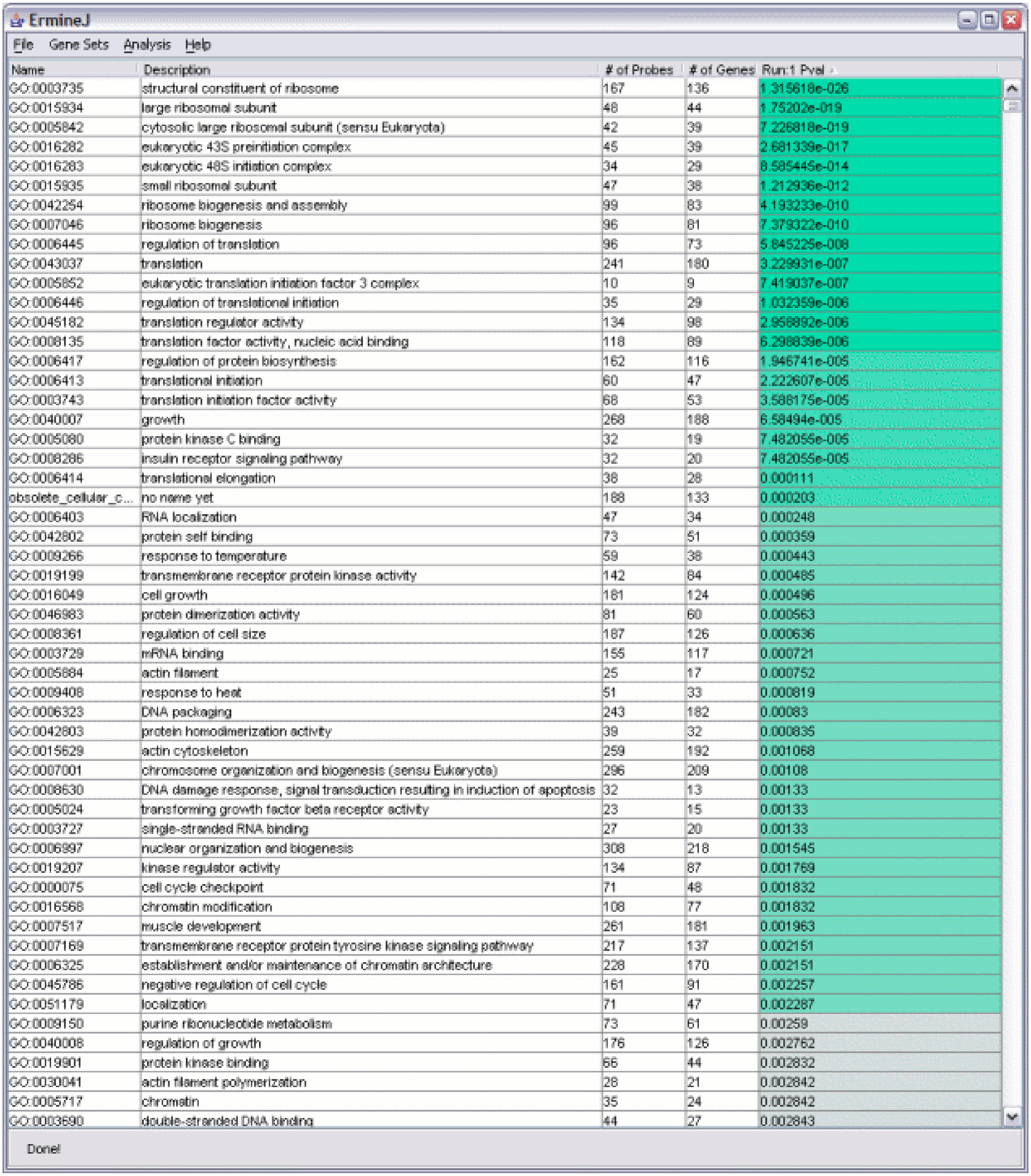
Multiple Test Correction in ermineJ

ErmineJ uses Benjamini-Hochberg correction of p values to determine which gene sets are selected with a false discovery rate (FDR) of 0.05. Such sets are colored shades of green in the pval column, as shown in the figure below. Gene sets which to not meet this criterion are not colored.

Brighter shades of color indicate even lower FDRs. Currently the colors indicate FDRs of 0.001, 0.01, 0.05 and 0.1. If you have trouble figuring out which color is which, just look at the tooltip you get when you point the mouse at a result, or save the results to a file - the corrected p-values are saved in their own column.

Note that the color reflects the FDR when the p values for the gene set indicated is used as the threshold. Thus it is possible for the FDR to go up and down as you go down the list. It is not uncommon for the FDR at very stringent thresholds to be above 0.05, while at less stringent thresholds is can be lower. This counterintuitive result is due to the way the FDR is computed. If you want to control the FDR at 0.05, you should pick the lowest-ranked 'blue' gene set, and all gene sets listed above would be selected.

If you feel using the FDR is not appropriate, other multiple test correction methods such as Bonferroni can be accessed from the command line interface. You can also work with the results file you save to compute other types of corrected p values from your data.



Name	Description	# of Probes	# of Genes	Run:1 Pval
GO:0003735	structural constituent of ribosome	167	136	1.315618e-026
GO:0015934	large ribosomal subunit	48	44	1.75202e-019
GO:0005842	cytosolic large ribosomal subunit (sensu Eukaryota)	42	39	7.226818e-019
GO:0016282	eukaryotic 43S preinitiation complex	45	39	2.681339e-017
GO:0016283	eukaryotic 48S initiation complex	34	29	8.585445e-014
GO:0015935	small ribosomal subunit	47	38	1.212936e-012
GO:0042254	ribosome biogenesis and assembly	99	83	4.193233e-010
GO:0007046	ribosome biogenesis	96	81	7.379322e-010
GO:0006445	regulation of translation	96	73	5.845225e-008
GO:0043037	translation	241	180	3.229931e-007
GO:0005852	eukaryotic translation initiation factor 3 complex	10	9	7.419037e-007
GO:0006446	regulation of translational initiation	35	29	1.032359e-006
GO:0045182	translation regulator activity	134	98	2.958892e-006
GO:0008135	translation factor activity, nucleic acid binding	118	89	6.298839e-006
GO:0006417	regulation of protein biosynthesis	162	116	1.946741e-005
GO:0006413	translational initiation	60	47	2.222607e-005
GO:0003743	translation initiation factor activity	68	53	3.588175e-005
GO:0040007	growth	268	188	6.58494e-005
GO:0005080	protein kinase C binding	32	19	7.482055e-005
GO:0008286	insulin receptor signaling pathway	32	20	7.482055e-005
GO:0006414	translational elongation	38	28	0.000111
obsolete_cellular_c...	no name yet	188	133	0.000203
GO:0006403	RNA localization	47	34	0.000248
GO:0042802	protein self binding	73	51	0.000359
GO:0009266	response to temperature	59	38	0.000443
GO:0019199	transmembrane receptor protein kinase activity	142	84	0.000485
GO:0016049	cell growth	181	124	0.000496
GO:0046983	protein dimerization activity	81	60	0.000563
GO:0008361	regulation of cell size	187	126	0.000636
GO:0003729	mRNA binding	155	117	0.000721
GO:0005884	actin filament	25	17	0.000752
GO:0009408	response to heat	51	33	0.000819
GO:0006323	DNA packaging	243	182	0.00083
GO:0042803	protein homodimerization activity	39	32	0.000835
GO:0015629	actin cytoskeleton	259	192	0.001068
GO:0007001	chromosome organization and biogenesis (sensu Eukaryota)	296	209	0.00108
GO:0008630	DNA damage response, signal transduction resulting in induction of apoptosis	32	13	0.00133
GO:0005024	transforming growth factor beta receptor activity	23	15	0.00133
GO:0003727	single-stranded RNA binding	27	20	0.00133
GO:0006997	nuclear organization and biogenesis	308	218	0.001545
GO:0019207	kinase regulator activity	134	87	0.001769
GO:0000075	cell cycle checkpoint	71	48	0.001832
GO:0016568	chromatin modification	108	77	0.001832
GO:0007517	muscle development	261	181	0.001963
GO:0007169	transmembrane receptor protein tyrosine kinase signaling pathway	217	137	0.002151
GO:0006325	establishment and/or maintenance of chromatin architecture	228	170	0.002151
GO:0045786	negative regulation of cell cycle	161	91	0.002257
GO:0051179	localization	71	47	0.002287
GO:0009150	purine ribonucleotide metabolism	73	61	0.00259
GO:0040008	regulation of growth	176	126	0.002762
GO:0019901	protein kinase binding	66	44	0.002832
GO:0030041	actin filament polymerization	28	21	0.002842
GO:0005717	chromatin	35	24	0.002842
GO:0003690	double-stranded DNA binding	44	27	0.002843

Done!

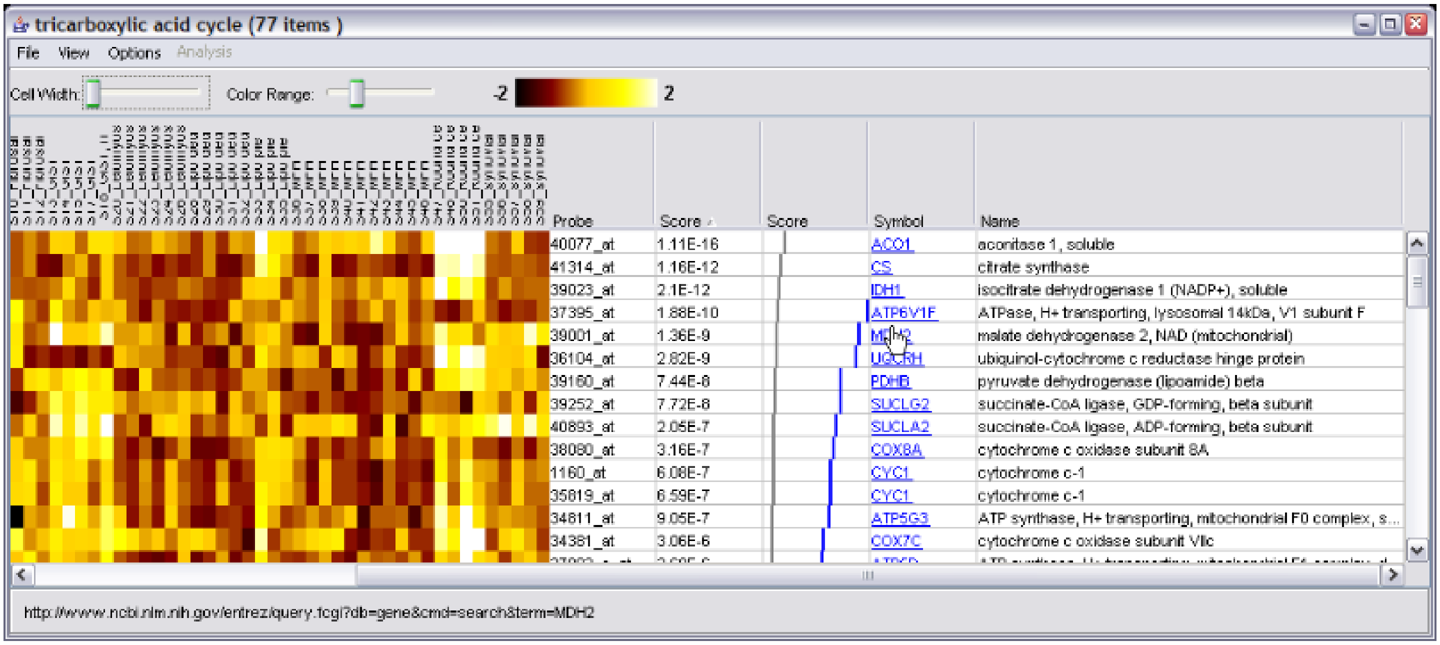
This screen shot is not up-to-date

ermineJ

Exploring gene set details

From the Output Panel (table or tree), you can double-click on a gene set row to pop up a new window showing the details for the gene set. If you have done multiple analyses, **click on the p-value for the specific gene set you wish to view**. If you have not done any analysis, you will still see the window. After double-clicking, you will see a new window.

A typical window will look like this



Probe	Score	Score	Symbol	Name
40077_at	1.11E-16		ACO1	aconitase 1, soluble
41314_at	1.16E-12		CS	citrate synthase
39023_at	2.1E-12		IDH1	isocitrate dehydrogenase 1 (NADP+), soluble
37395_at	1.88E-10		ATP6V1F	ATPase, H+ transporting, lysosomal 14kDa, V1 subunit F
39001_at	1.36E-9		MDH2	malate dehydrogenase 2, NAD (mitochondrial)
36104_at	2.82E-9		UQCRH	ubiquinol-cytochrome c reductase hinge protein
39160_at	7.44E-8		PDHB	pyruvate dehydrogenase (lipoamide) beta
39252_at	7.72E-8		SUCLG2	succinate-CoA ligase, GDP-forming, beta subunit
40893_at	2.05E-7		SUCLA2	succinate-CoA ligase, ADP-forming, beta subunit
38080_at	3.16E-7		COX8A	cytochrome c oxidase subunit 8A
1160_at	6.08E-7		CYC1	cytochrome c-1
35819_at	6.59E-7		CYC1	cytochrome c-1
34811_at	9.05E-7		ATP5G3	ATP synthase, H+ transporting, mitochondrial F0 complex, s...
34381_at	3.06E-6		COX7C	cytochrome c oxidase subunit VIIc

http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene&cmd=search&term=MDH2

There are a few things to note:

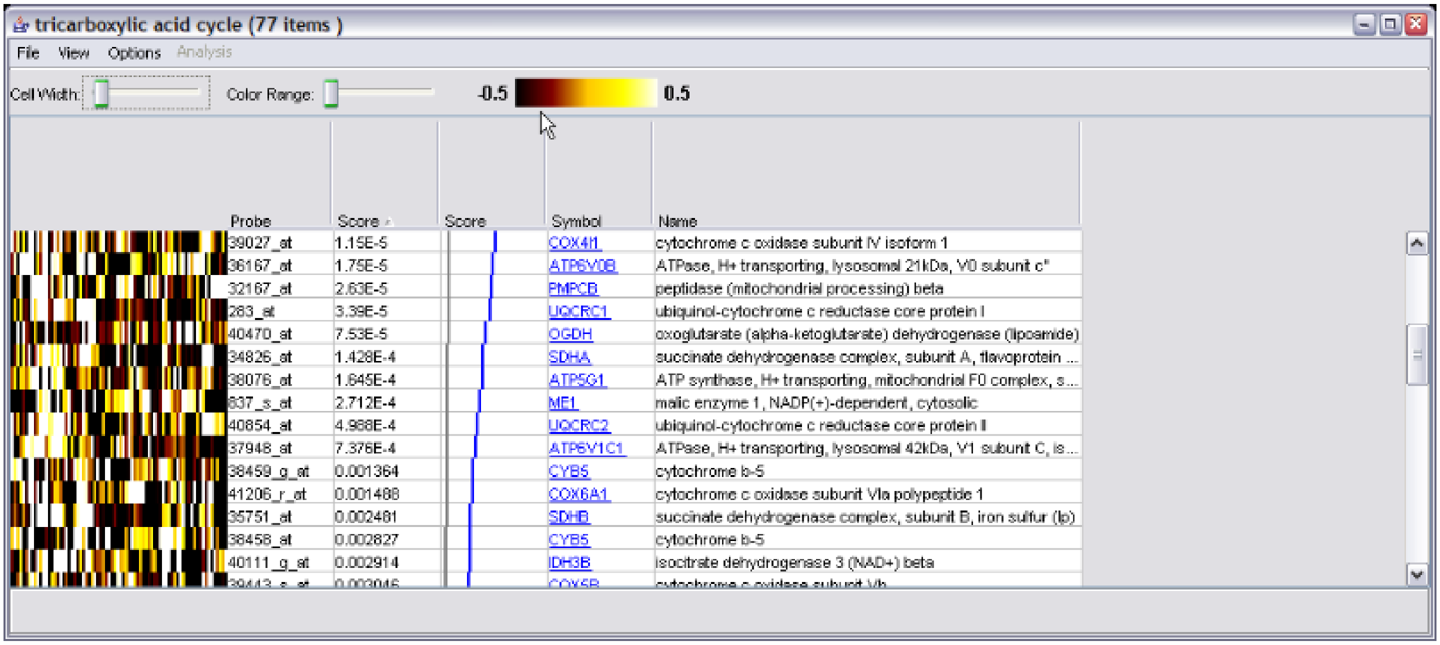
- If you have done an analysis, then the relevant gene score file will be defined, and the probe scores for the probes in the gene set will be displayed. If you have no current analyses, then the last gene score file you used will be selected for showing probe scores. If you have not selected a gene score file, no gene scores will be displayed (you will see "NaN" for "not a number" instead).
- If you have previously loaded your raw expression data, the window will include a visualization of the data. If you have not, you will be prompted to choose a data file. The file format must follow the rules described here. If you choose not to load a data file at this time, you will not be asked again during this session. If you want to see raw data, you can set the data set from a menu option.
- You can open as many visualization windows as you like, including multiple windows for the same gene set.
- You can switch the data set to view. This setting is remembered across sessions. If you set a raw data file in the Analysis wizard, that setting will be used.
- The title of the window shows the gene set name, size and the p-value, if there is one

The non-data columns are as follows:

- Probe - the identifier supplied by the array manufacturer, read in from the annotation file. In this example, this is an Affymetrix array.
- Score - the user-supplied gene score (if you have supplied one). This value is loaded from the "gene scores" file. Probes which were present in the array design, but not used for analysis due to lack of a gene score, are shown with a score of "NaN" (not a number).
- Score (visual) - a graph showing the scores graphically. The blue line represents the scores. The grey line shows the expected distribution under the null hypothesis and assuming independence (this only works if the values are p values).
- Symbol - the official gene symbol as supplied in the annotation file, with hyperlinks to a web site of your choosing. You can change the target URL as described here .
- Name - the gene name as supplied in the annotation file.

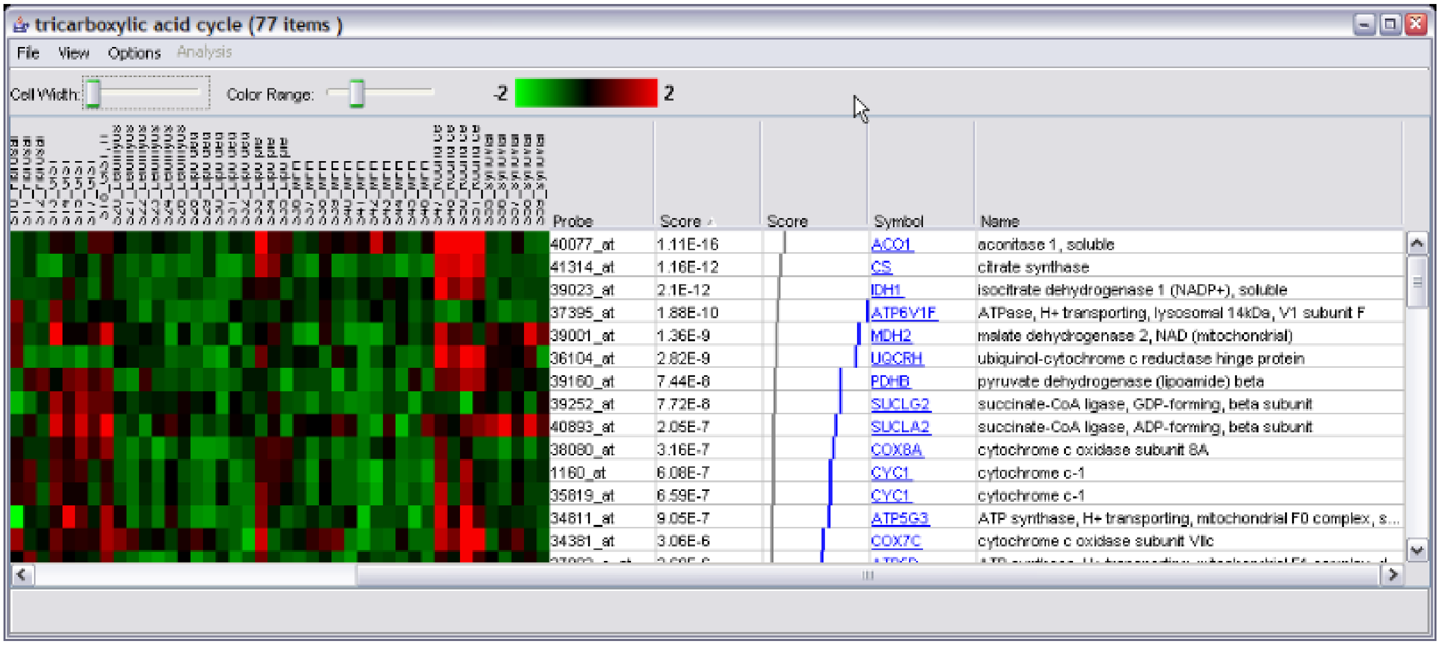
Image controls

- **Sorting**: Clicking on any column (including the raw data column) headers result in sorting of the view.
- **Contrast**: You can adjust the color range (contrast) and cell width for the color map.

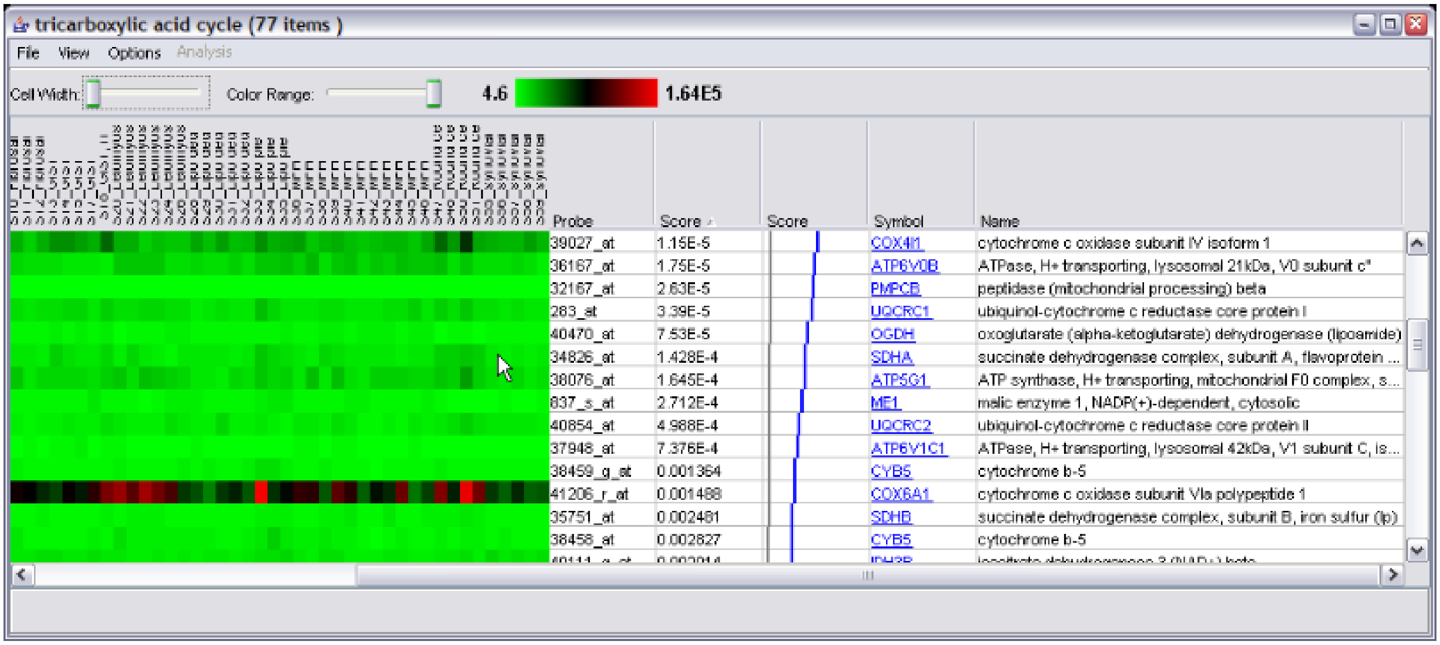


Probe	Score	Score	Symbol	Name
39027_at	1.15E-5		COX4I1	cytochrome c oxidase subunit IV isoform 1
36167_at	1.75E-5		ATP6V0B	ATPase, H+ transporting, lysosomal 21kDa, V0 subunit c"
32167_at	2.63E-5		PMPCB	peptidase (mitochondrial processing) beta
283_at	3.39E-5		UQCRC1	ubiquinol-cytochrome c reductase core protein I
40470_at	7.53E-5		OGDH	oxoglutarate (alpha-ketoglutarate) dehydrogenase (lipoamide)
34826_at	1.428E-4		SDHA	succinate dehydrogenase complex, subunit A, flavoprotein ...
38076_at	1.645E-4		ATP5G1	ATP synthase, H+ transporting, mitochondrial F0 complex, s...
837_s_at	2.712E-4		ME1	malic enzyme 1, NADP(+)-dependent, cytosolic
40854_at	4.988E-4		UQCRC2	ubiquinol-cytochrome c reductase core protein II
37948_at	7.376E-4		ATP6V1C1	ATPase, H+ transporting, lysosomal 42kDa, V1 subunit C, is...
38459_g_at	0.001364		CYB5	cytochrome b-5
41206_r_at	0.001488		COX6A1	cytochrome c oxidase subunit VIa polypeptide 1
35751_at	0.002481		SDHB	succinate dehydrogenase complex, subunit B, iron sulfur (Ip)
38458_at	0.002827		CYB5	cytochrome b-5
40111_g_at	0.002914		IDH3B	isocitrate dehydrogenase 3 (NAD+) beta
39443_s_at	0.003046		COX5B	cytochrome c oxidase subunit Vb

- Resize the columns in the image view. Use this to "zoom out" to make the image fit horizontally on your screen.
- Using the "View" menu, you can switch between different preset color maps and choose normalize or non-normalized data for viewing.



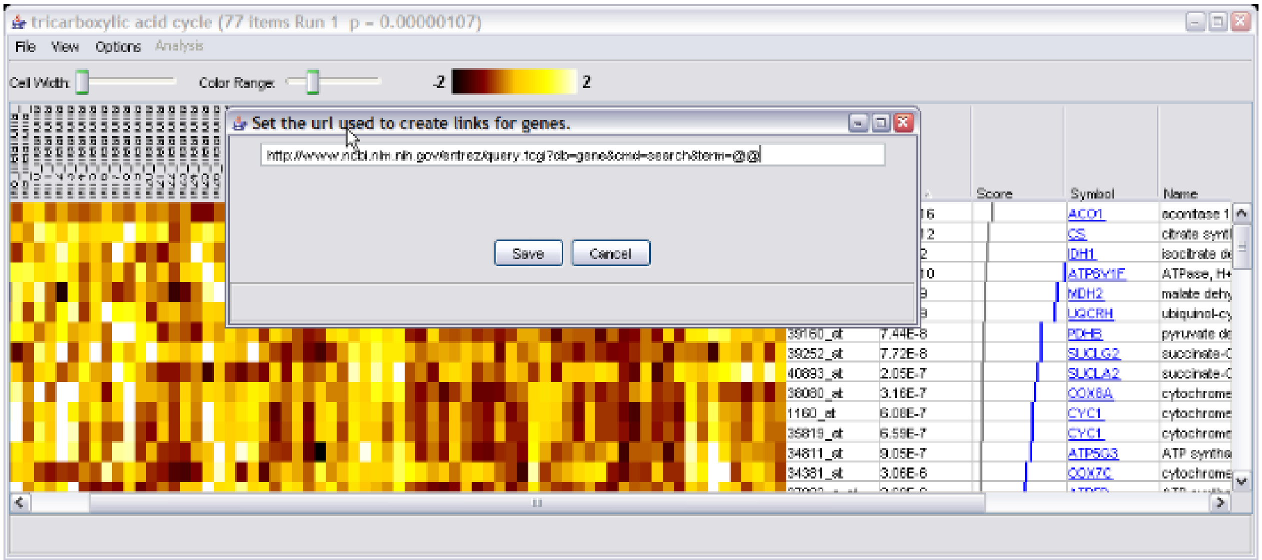
Probe	Score	Score	Symbol	Name
40077_at	1.11E-16		ACO1	aconitase 1, soluble
41314_at	1.16E-12		CS	citrate synthase
39023_at	2.1E-12		IDH1	isocitrate dehydrogenase 1 (NADP+), soluble
37395_at	1.88E-10		ATP6V1F	ATPase, H+ transporting, lysosomal 14kDa, V1 subunit F
39001_at	1.36E-9		MDH2	malate dehydrogenase 2, NAD (mitochondrial)
36104_at	2.82E-9		UQCRH	ubiquinol-cytochrome c reductase hinge protein
39160_at	7.44E-8		PDHB	pyruvate dehydrogenase (lipoamide) beta
39252_at	7.72E-8		SUCLG2	succinate-CoA ligase, GDP-forming, beta subunit
40893_at	2.05E-7		SUCLA2	succinate-CoA ligase, ADP-forming, beta subunit
38080_at	3.16E-7		COX8A	cytochrome c oxidase subunit 8A
1160_at	6.08E-7		CYC1	cytochrome c-1
35819_at	6.59E-7		CYC1	cytochrome c-1
34811_at	9.05E-7		ATP5G3	ATP synthase, H+ transporting, mitochondrial F0 complex, s...
34381_at	3.06E-6		COX7C	cytochrome c oxidase subunit VIIc



Probe	Score	Score	Symbol	Name
39027_at	1.15E-5		COX4I1	cytochrome c oxidase subunit IV isoform 1
36167_at	1.75E-5		ATP6V0B	ATPase, H+ transporting, lysosomal 21kDa, V0 subunit c"
32167_at	2.63E-5		PMPCB	peptidase (mitochondrial processing) beta
283_at	3.39E-5		UQCRC1	ubiquinol-cytochrome c reductase core protein I
40470_at	7.53E-5		OGDH	oxoglutarate (alpha-ketoglutarate) dehydrogenase (lipoamide)
34826_at	1.428E-4		SDHA	succinate dehydrogenase complex, subunit A, flavoprotein ...
38076_at	1.645E-4		ATP5G1	ATP synthase, H+ transporting, mitochondrial F0 complex, s...
837_s_at	2.712E-4		ME1	malic enzyme 1, NADP(+)-dependent, cytosolic
40854_at	4.988E-4		UQCRC2	ubiquinol-cytochrome c reductase core protein II
37948_at	7.376E-4		ATP6V1C1	ATPase, H+ transporting, lysosomal 42kDa, V1 subunit C, is...
38459_g_at	0.001364		CYB5	cytochrome b-5
41206_r_at	0.001488		COX6A1	cytochrome c oxidase subunit VIa polypeptide 1
35751_at	0.002481		SDHB	succinate dehydrogenase complex, subunit B, iron sulfur (Ip)
38458_at	0.002827		CYB5	cytochrome b-5

Changing the hyperlinks

When you click on a gene symbol, your web browser will be opened to a web site of your choosing. The preset is the NCBI "Gene" database, but you can change this using the "Options->Change gene name URL" menu item:


tricarboxylic acid cycle (77 items Run 1 p = 0.00000107)
File
View
Options
Analysis
Cell Width:
Color Range:
-2
2
Set the url used to create links for genes.
Save
Cancel

http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene&cmd=search&term=@@

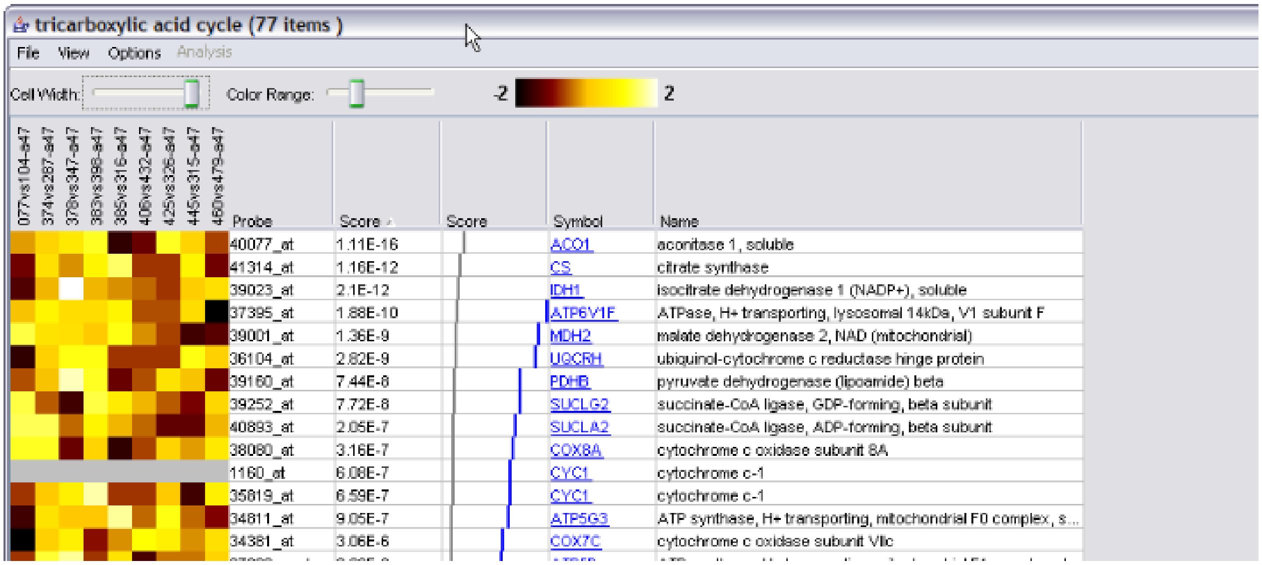
		Score	Symbol	Name
	16		ACO1	aconitase 1
	12		CS	citrate synt
	2		IDH1	isocitrate de
	10		ATP6V1F	ATPase, H+
	9		MDH2	malate dehy
	9		UQCRH	ubiquinol-cy
39160_at	7.44E-8		PDHB	pyruvate de
39252_at	7.72E-8		SUCLG2	succinate-C
40893_at	2.05E-7		SUCLA2	succinate-C
36080_at	3.16E-7		COX8A	cytochrome
1160_at	6.08E-7		CYC1	cytochrome
35819_at	6.59E-7		CYC1	cytochrome
34811_at	9.05E-7		ATP5G3	ATP syntha
34381_at	3.06E-6		COX7C	cytochrome

Using this requires that you know the URL (web address) for a gene. Just replace the term you would normally put in the URL with two "at" symbols ("@@"). They will be replaced with the Gene Symbol.

For details and some examples, see this page.

Changing the data set

You can switch data sets (or define one if you have never loaded one) viewed in the heatmap by selecting the "Options->Change data set". Here's an example where we switched:



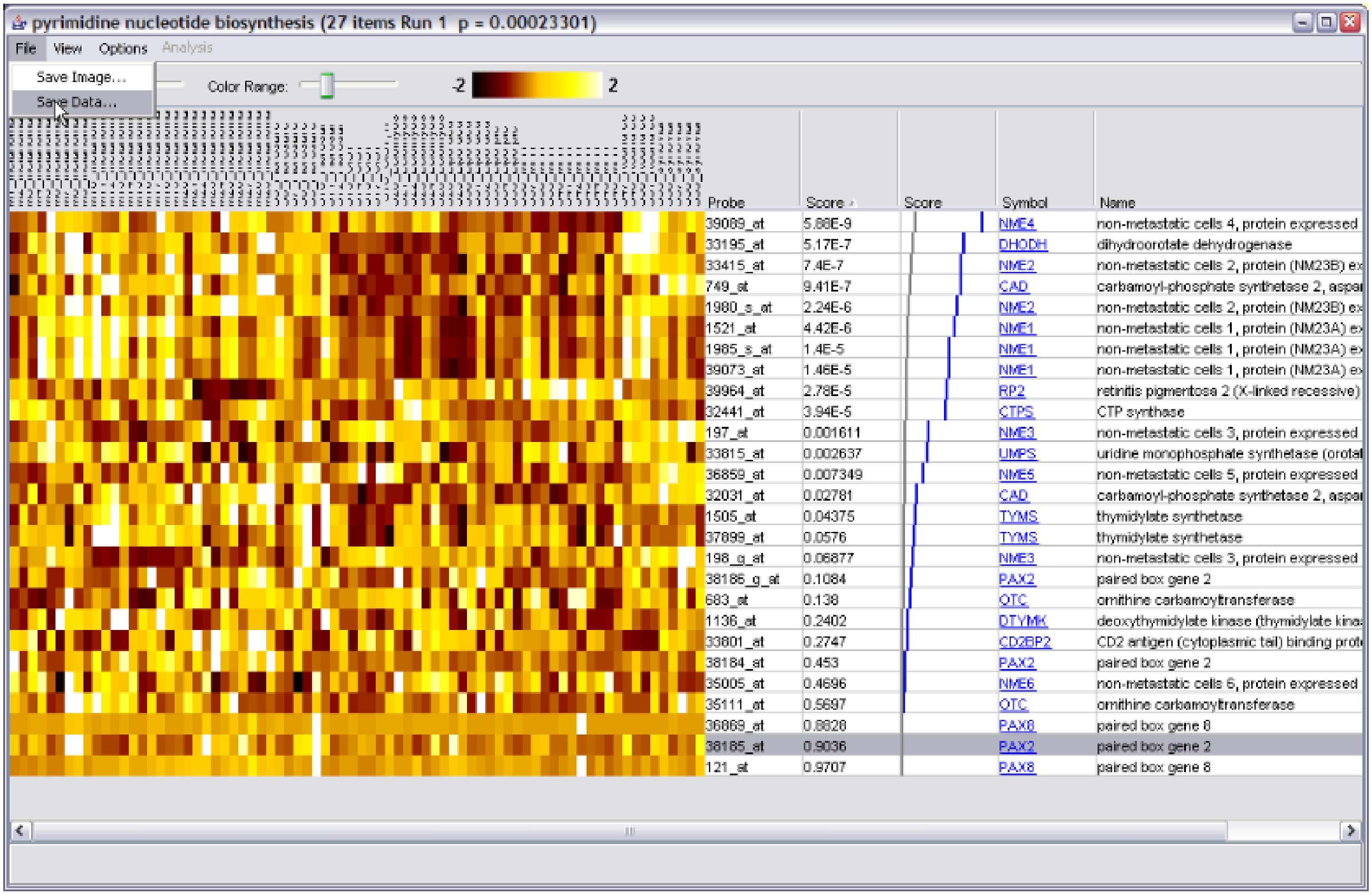
Probe	Score	Score	Symbol	Name
40077_at	1.11E-16		ACO1	aconitase 1, soluble
41314_at	1.16E-12		CS	citrate synthase
39023_at	2.1E-12		IDH1	isocitrate dehydrogenase 1 (NADP+), soluble
37395_at	1.88E-10		ATP6V1F	ATPase, H+ transporting, lysosomal 14kDa, V1 subunit F
39001_at	1.36E-9		MDH2	malate dehydrogenase 2, NAD (mitochondrial)
36104_at	2.82E-9		UQCRH	ubiquinol-cytochrome c reductase hinge protein
39160_at	7.44E-8		PDHB	pyruvate dehydrogenase (lipoamide) beta
39252_at	7.72E-8		SUCLG2	succinate-CoA ligase, GDP-forming, beta subunit
40893_at	2.05E-7		SUCLA2	succinate-CoA ligase, ADP-forming, beta subunit
38080_at	3.16E-7		COX8A	cytochrome c oxidase subunit 8A
1160_at	6.08E-7		CYC1	cytochrome c-1
35819_at	6.59E-7		CYC1	cytochrome c-1
34811_at	9.05E-7		ATP5G3	ATP synthase, H+ transporting, mitochondrial F0 complex, s...
34381_at	3.06E-6		COX7C	cytochrome c oxidase subunit VIIc

Note: Naturally, if you switch to a data set that uses a different microarray design, the results might not make much sense. Note that the gene scores displayed are always the ones you used for the last analysis (or the run you double-clicked on) to get this window)

Note: If you switch data sets, the setting is "sticky": all new visualization windows you open will use this data set, until you switch back.

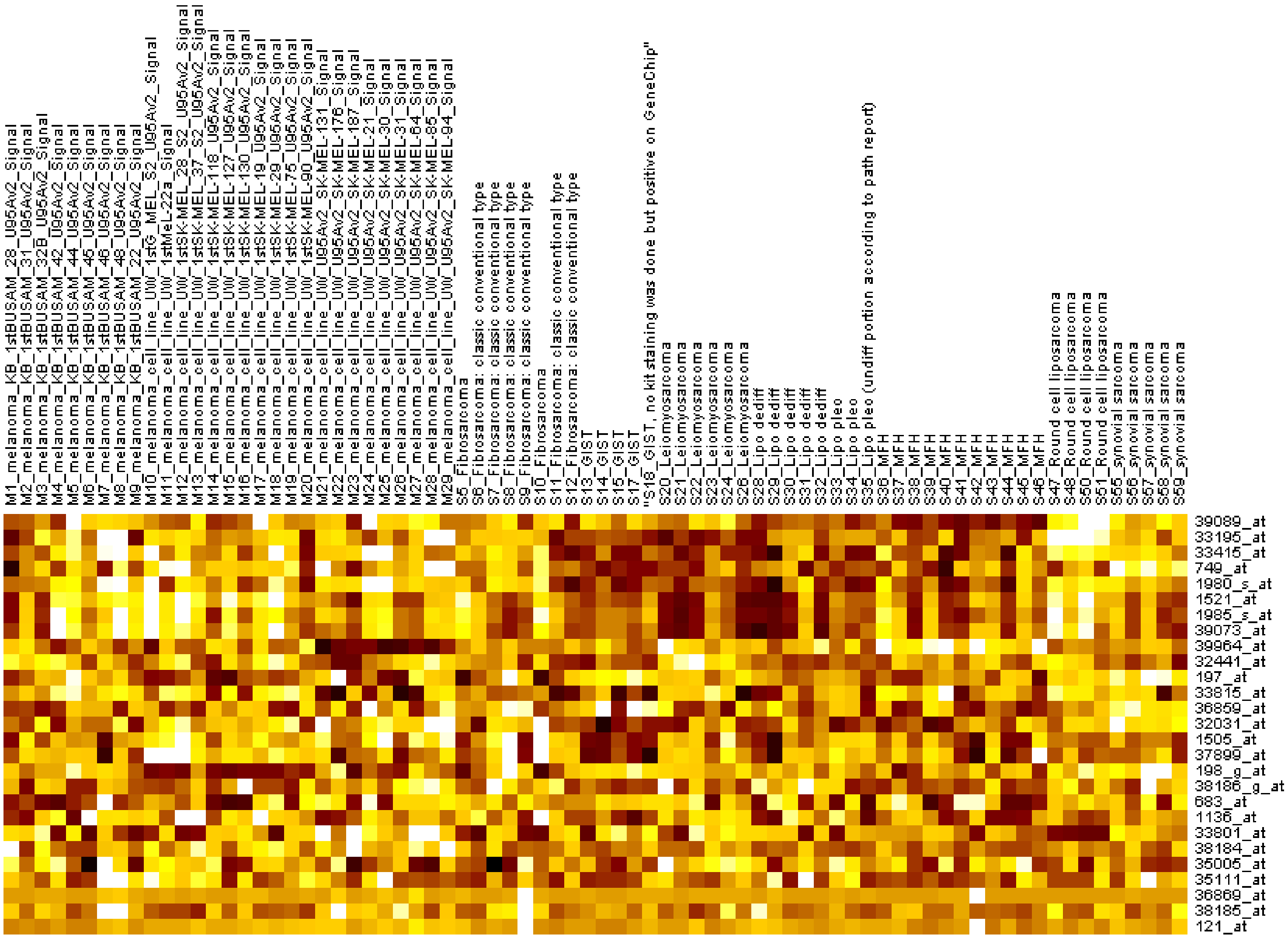
Saving the data and/or image

Using the "File" menu, you can save the data shown, in text or image formats:

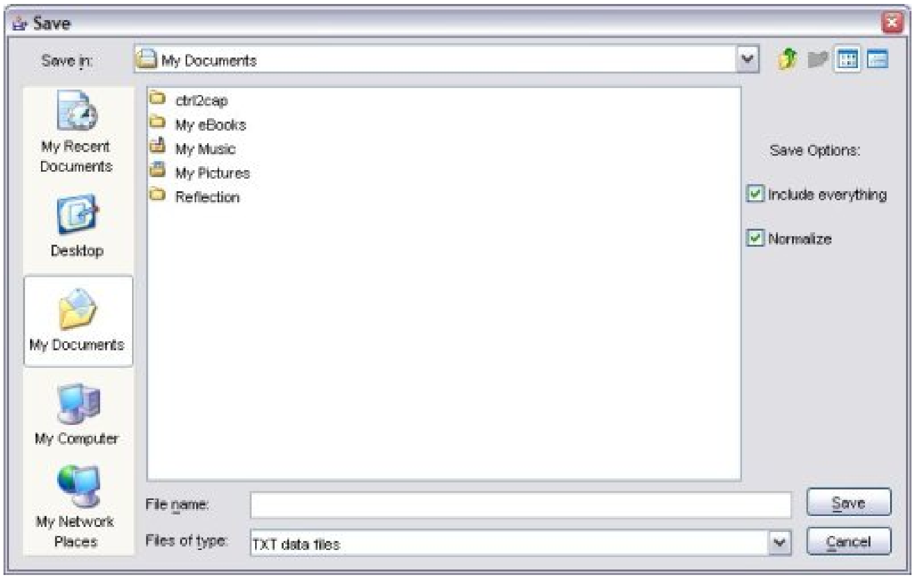

pyrimidine nucleotide biosynthesis (27 items Run 1 p = 0.00023301)
File
View
Options
Analysis
Save Image...
Save Data...
Color Range:
-2
2

Probe	Score	Score	Symbol	Name
39089_at	5.88E-9		NME4	non-metastatic cells 4, protein expressed
33195_at	5.17E-7		DHODH	dihydroorotate dehydrogenase
33415_at	7.4E-7		NME2	non-metastatic cells 2, protein (NM23B) ex
749_at	9.41E-7		CAD	carbamoyl-phosphate synthetase 2, aspar
1980_s_at	2.24E-6		NME2	non-metastatic cells 2, protein (NM23B) ex
1521_at	4.42E-6		NME1	non-metastatic cells 1, protein (NM23A) ex
1985_s_at	1.4E-5		NME1	non-metastatic cells 1, protein (NM23A) ex
39073_at	1.46E-5		NME1	non-metastatic cells 1, protein (NM23A) ex
39964_at	2.78E-5		RP2	retinitis pigmentosa 2 (X-linked recessive)
32441_at	3.94E-5		CTPS	CTP synthase
197_at	0.001611		NME3	non-metastatic cells 3, protein expressed
33815_at	0.002637		UMPS	uridine monophosphate synthetase (orotat
36859_at	0.007349		NME5	non-metastatic cells 5, protein expressed
32031_at	0.02781		CAD	carbamoyl-phosphate synthetase 2, aspar
1505_at	0.04375		TYMS	thymidylate synthetase
37899_at	0.0576		TYMS	thymidylate synthetase
198_g_at	0.06877		NME3	non-metastatic cells 3, protein expressed
38186_g_at	0.1084		PAX2	paired box gene 2
683_at	0.138		OTC	ornithine carbamoyltransferase
1136_at	0.2402		DTYMK	deoxythymidylate kinase (thymidylate kina
33801_at	0.2747		CD2BP2	CD2 antigen (cytoplasmic tail) binding prot
38184_at	0.453		PAX2	paired box gene 2
35005_at	0.4696		NME6	non-metastatic cells 6, protein expressed
35111_at	0.5697		OTC	ornithine carbamoyltransferase
36869_at	0.8828		PAX8	paired box gene 8
38185_at	0.9036		PAX2	paired box gene 2
121_at	0.9707		PAX8	paired box gene 8

Here is an example of the resulting image, which is in PNG format:



You can also save the data using the "File...Save data..." menu option:



Save in: My Documents

Save Options:

☑ Include everything

☑ Normalize

File name:

Files of type: TXT data files

A sample resulting file for the data above is here. In this case we had the "include annotations" but not "normalize" checked. This file can be loaded into Excel or a simliar program for further analysis.

ermineJ

Viewing, defining and modifying gene sets

A **gene set** is any grouping of genes defined by criteria other than the data currently being analyzed. Our baseline gene sets are defined using Gene Ontology terms as applied in the publicly available sources listed on the GO web site.

You can view, import, define, or modify gene sets as you like, using the "gene set" menu on the main panel of the software. This page explains how.

Viewing a gene set

When the software first starts, the gene sets are defined by the microarray annotation file you uploaded. Prior to doing any analysis, or at any other time, you can view the list of genes in a gene set in one of three ways.

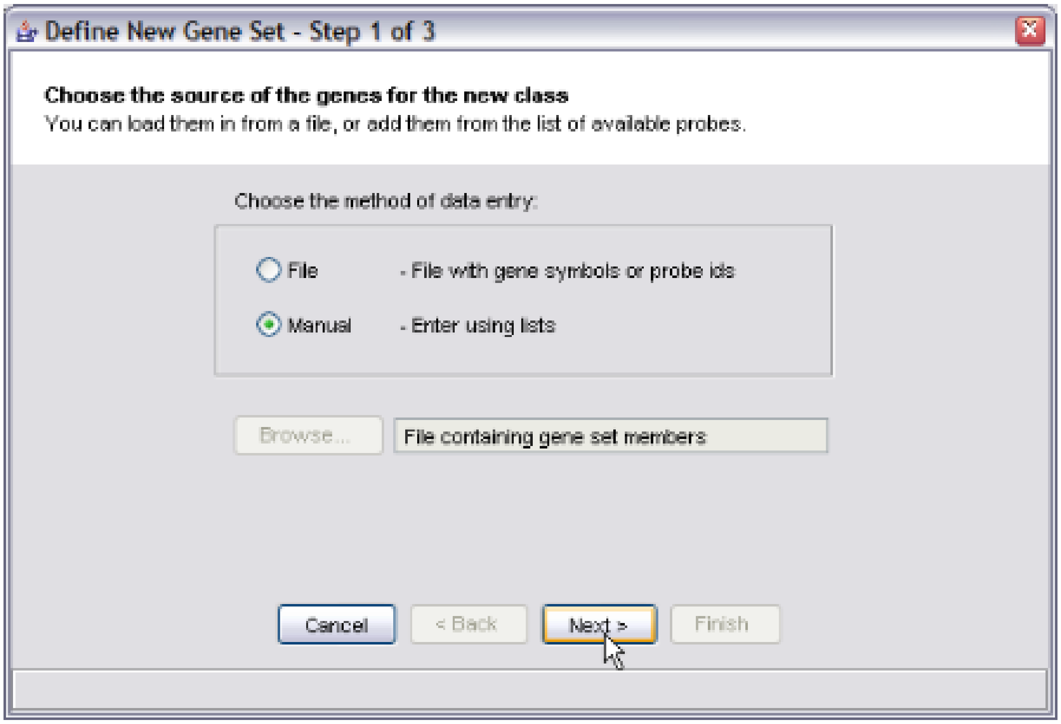
1. You can "right click" on the gene set name and select "View/Modify gene set". You will be shown the details for that gene set in a new window. If you are just looking, and don't want to modify the gene set, just hit cancel when you are done.
2. You can select the "View/Modify gene set" from the analysis menu. In this case you then select which gene set to view.
3. You can double-click on the gene set name to bring up the visualization of the gene set

The first two options are appropriate if you want to modify the gene set.

Creating a new gene set from scratch

Select "create new gene set" from the "gene set" menu. This gets you to the new gene set wizard.

You can import a file describing a gene set, or you can create one manually from the list of genes on the array you are using. You choose which to use by selecting "file" or "manual" in the first step of the new gene set wizard:



Choose the source of the genes for the new class
You can load them in from a file, or add them from the list of available probes.

Choose the method of data entry:

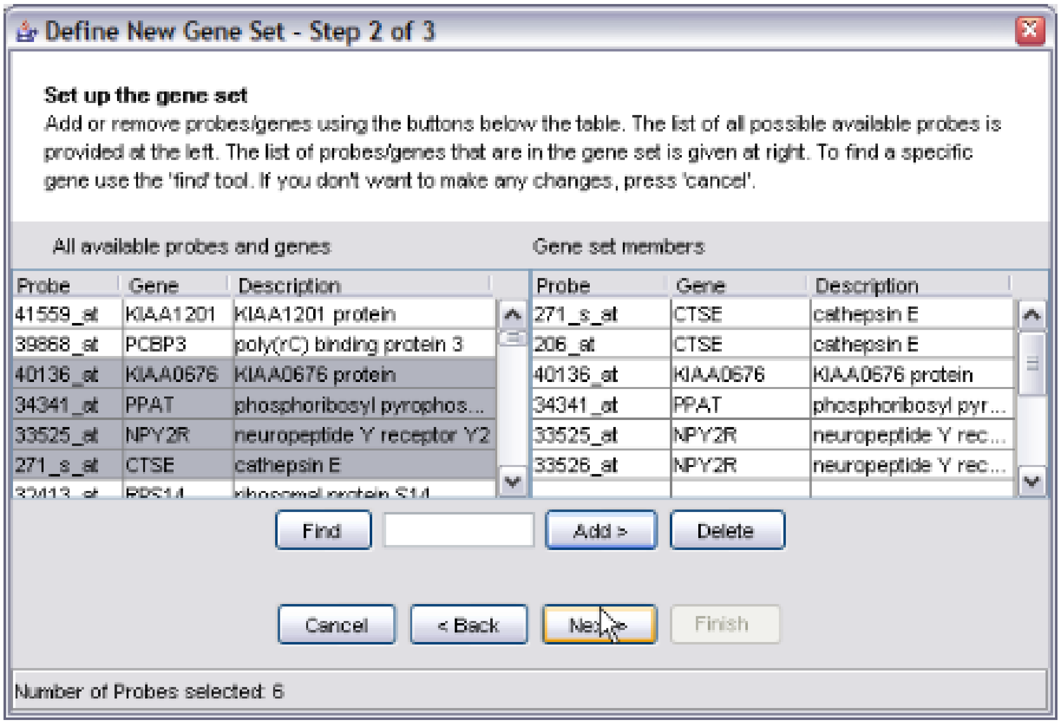
○ File - File with gene symbols or probe ids

● Manual - Enter using lists

File containing gene set members

Adding new genes manually

Select probes from the left panel and click "add" to move them to the right panel. Note that here we added one probe for the ABCA1 gene, but four probes were moved to the right. This is because there are four probes which assay ABCA1; all of them are added to the gene set automatically. Similarly you can delete probes from the right panel by using the 'delete' button.



Set up the gene set
Add or remove probes/genes using the buttons below the table. The list of all possible available probes is provided at the left. The list of probes/genes that are in the gene set is given at right. To find a specific gene use the 'find' tool. If you don't want to make any changes, press 'cancel'.

All available probes and genes

Probe	Gene	Description
41559_at	KIAA1201	KIAA1201 protein
39868_at	PCBP3	poly(rC) binding protein 3
40136_at	KIAA0676	KIAA0676 protein
34341_at	PPAT	phosphoribosyl pyrophos...
33525_at	NPY2R	neuropeptide Y receptor Y2
271_s_at	CTSE	cathepsin E
32413_at	RPS14	ribosomal protein S14

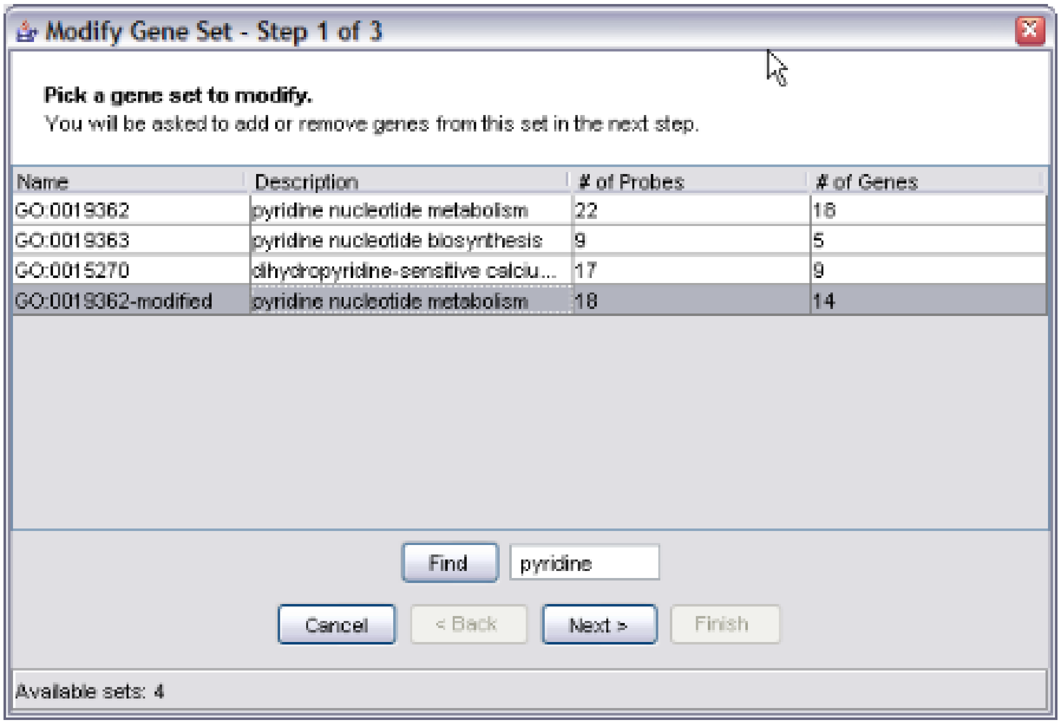
Gene set members

Probe	Gene	Description
271_s_at	CTSE	cathepsin E
206_at	CTSE	cathepsin E
40136_at	KIAA0676	KIAA0676 protein
34341_at	PPAT	phosphoribosyl pyr...
33525_at	NPY2R	neuropeptide Y rec...
33526_at	NPY2R	neuropeptide Y rec...

Number of Probes selected: 6

Using the 'find' function to help locate genes.

To make it easier to build new gene sets, a 'find' function is provided. Here we searched for "cell cycle".



Pick a gene set to modify.
You will be asked to add or remove genes from this set in the next step.

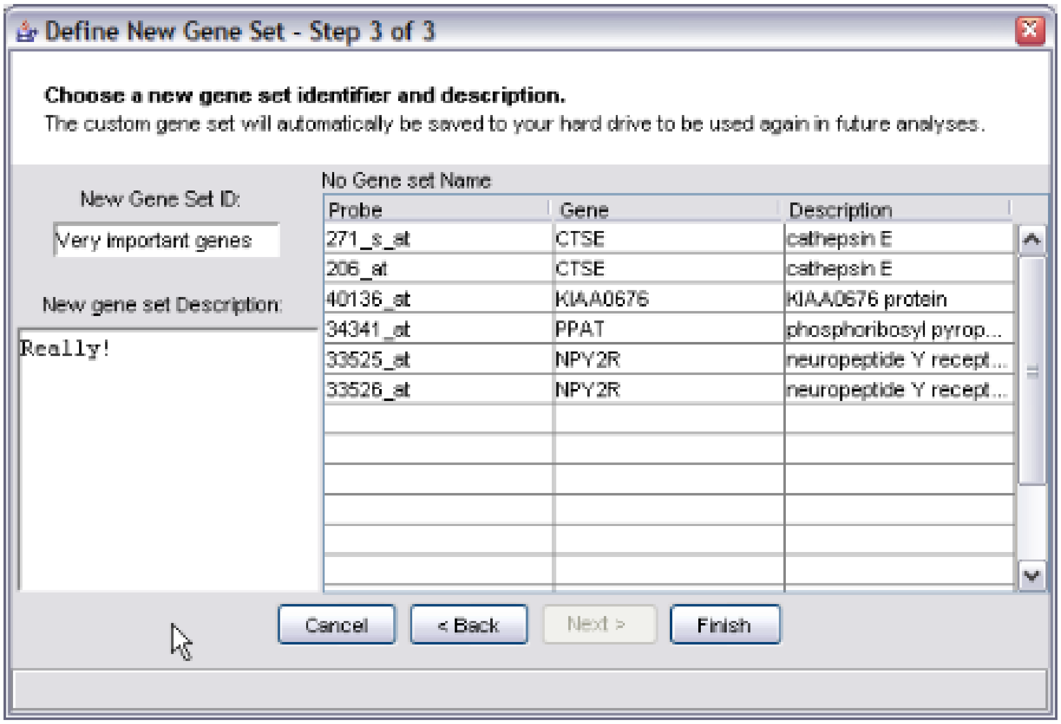
Name	Description	# of Probes	# of Genes
GO:0019362	pyridine nucleotide metabolism	22	16
GO:0019363	pyridine nucleotide biosynthesis	9	5
GO:0015270	dihydropyridine-sensitive calciu...	17	9
GO:0019362-modified	pyridine nucleotide metabolism	18	14

Find: pyridine

Available sets: 4

Giving the new gene set a name and description

After adding the probes and genes to the gene set and hitting "next", you will be asked to enter a new identifier and description for the gene set you defined. When you hit finish, the information about this gene set is saved to disk and you are returned to the main panel.



Choose a new gene set identifier and description.
The custom gene set will automatically be saved to your hard drive to be used again in future analyses.

New Gene Set ID: Very important genes

New gene set Description: Really!

No Gene set Name

Probe	Gene	Description
271_s_at	CTSE	cathepsin E
206_at	CTSE	cathepsin E
40136_at	KIAA0676	KIAA0676 protein
34341_at	PPAT	phosphoribosyl pyrop...
33525_at	NPY2R	neuropeptide Y recept...
33526_at	NPY2R	neuropeptide Y recept...

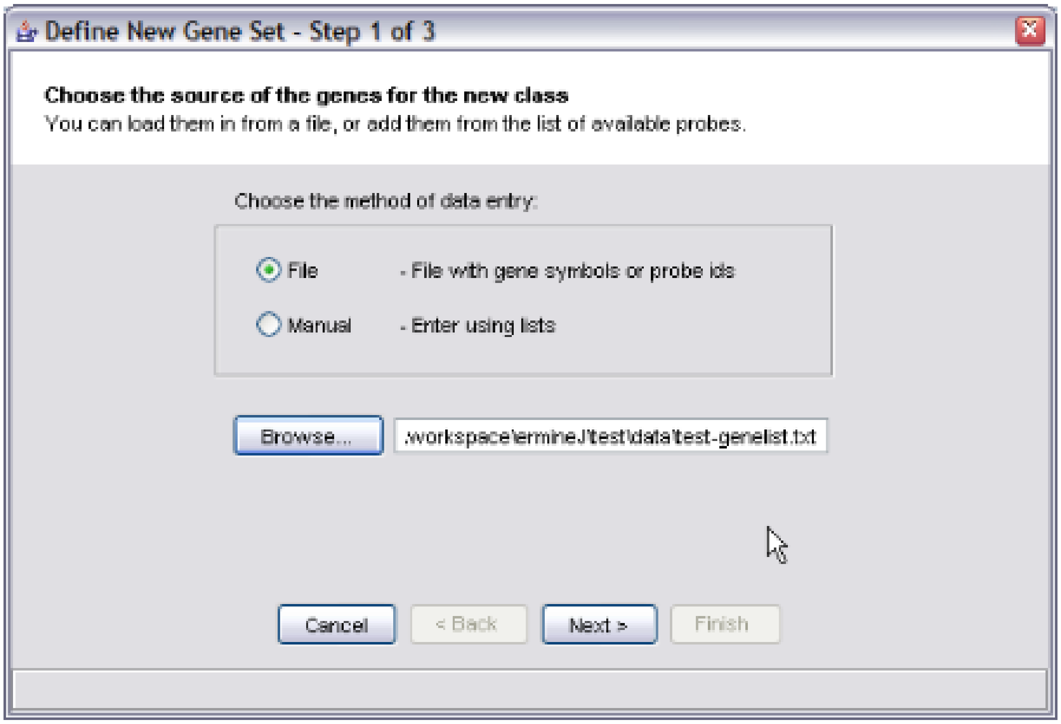
After hitting 'finish', the new gene set is shown in color in the output panel. The gene set is also saved to disk for future use. The format and file locations are described here.

If you can't see the new gene set on the output panel, press "Ctrl-U" to hide all but the user-defined gene sets. Hitting "Ctrl-U" again shows all the gene sets.

Entering a gene set from a file

Gene sets can be loaded in from files created by the user. For details on the format, see this page. But basically it is just a list of genes, one per row.

Select the "Load from file" option:



Define New Gene Set - Step 1 of 3

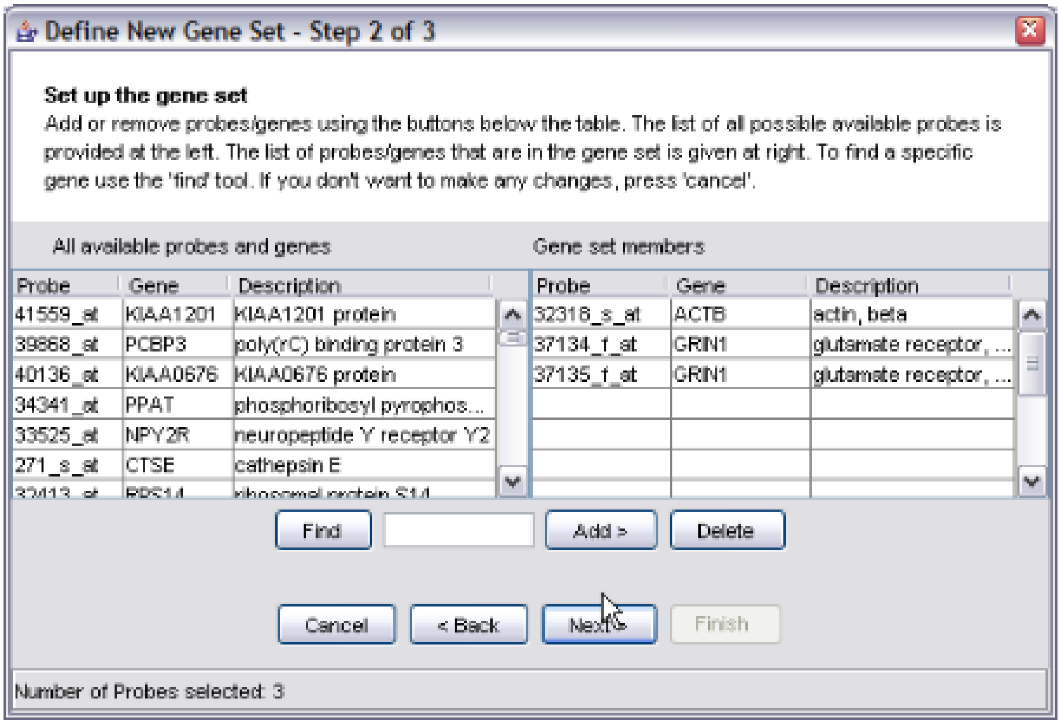
Choose the source of the genes for the new class
You can load them in from a file, or add them from the list of available probes.

Choose the method of data entry:

● File - File with gene symbols or probe ids
○ Manual - Enter using lists

Browse... .workspace\ermineJ\test\data\test-genelist.txt

After loading a gene set, you will be able to check modify it before giving it a name and saving it to disk.



Set up the gene set

Add or remove probes/genes using the buttons below the table. The list of all possible available probes is provided at the left. The list of probes/genes that are in the gene set is given at right. To find a specific gene use the 'find' tool. If you don't want to make any changes, press 'cancel'.

All available probes and genes

Probe	Gene	Description
41559_at	KIAA1201	KIAA1201 protein
39868_at	PCBP3	poly(rC) binding protein 3
40136_at	KIAA0676	KIAA0676 protein
34341_at	PPAT	phosphoribosyl pyrophos...
33525_at	NPY2R	neuropeptide Y receptor Y2
271_s_at	CTSE	cathepsin E
32413_at	RPS14	ribosomal protein S14

Gene set members

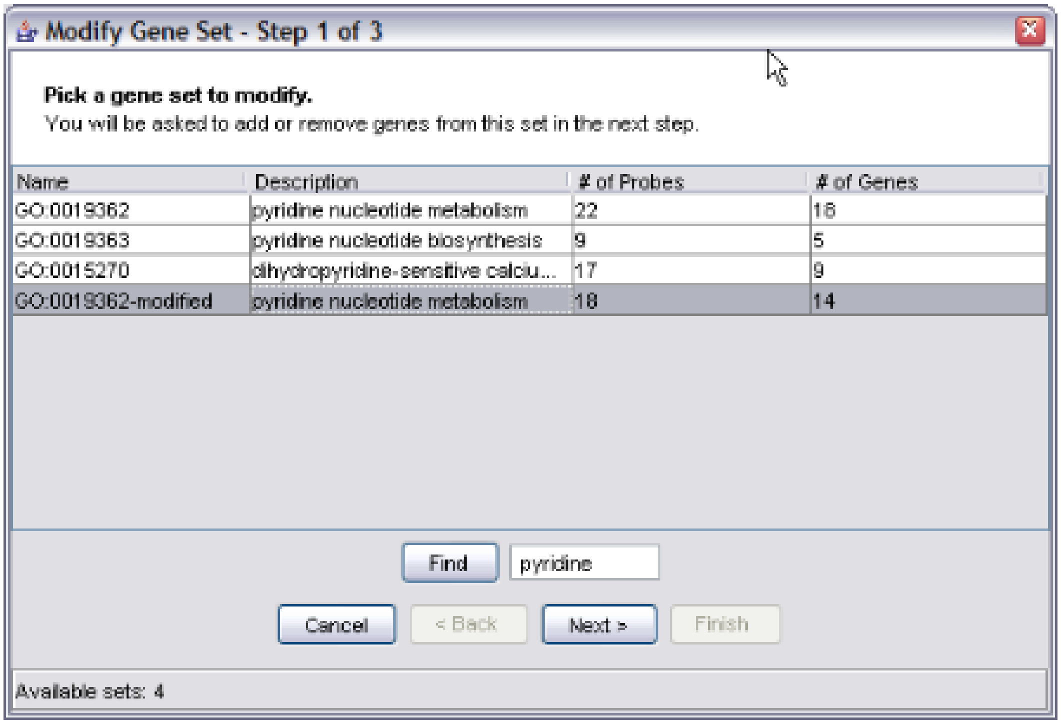
Probe	Gene	Description
32318_s_at	ACTB	actin, beta
37134_f_at	GRIN1	glutamate receptor, ...
37135_f_at	GRIN1	glutamate receptor, ...

Number of Probes selected: 3

Modifying an existing gene set

You can modify a pre-existing gene set by adding or removing genes. This is done using the "Modify gene set" wizard. This can be accessed either by right-clicking on the gene set in the Output Panel or by selecting the "modify" item from the Gene set menu.

If you use the "Gene set" menu, you will get a list of all the available gene sets. Select one by clicking on it. You can use the find function to help locate a specific set. After this step, the procedure is very similar to the "create new" gene set procedure explained above.



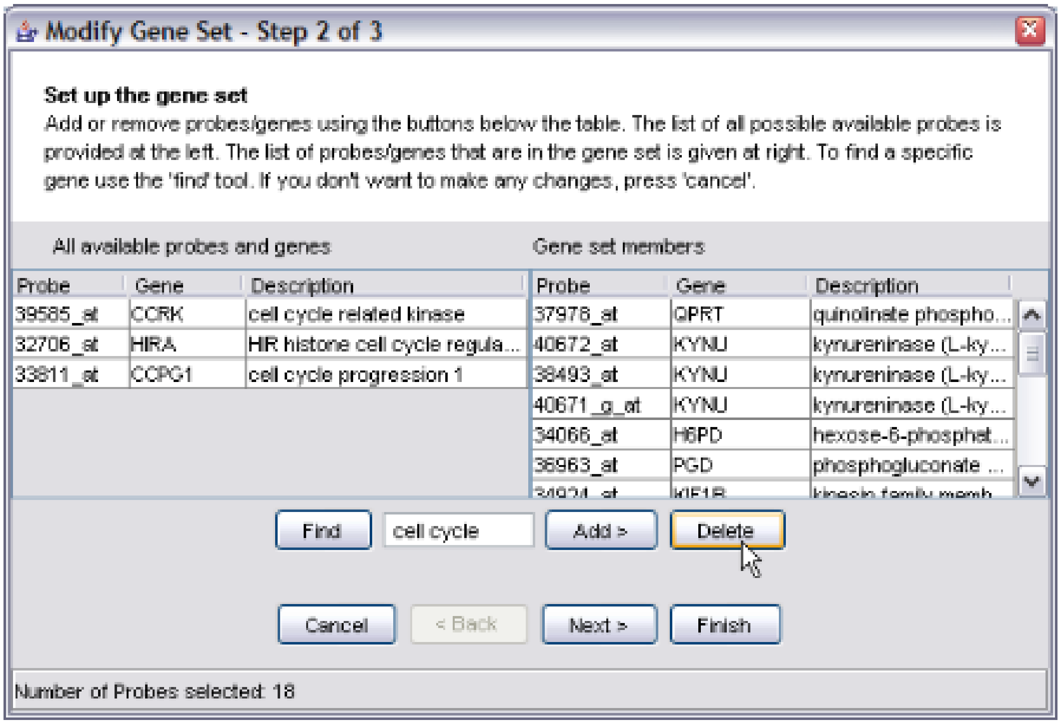
Pick a gene set to modify.
You will be asked to add or remove genes from this set in the next step.

Name	Description	# of Probes	# of Genes
GO:0019362	pyridine nucleotide metabolism	22	18
GO:0019363	pyridine nucleotide biosynthesis	9	5
GO:0015270	dihydropyridine-sensitive calciu...	17	9
GO:0019362-modified	pyridine nucleotide metabolism	18	14

Find: pyridine

Available sets: 4

The next screen lists the available probes on the left, and the probes in the set on the right. Similar to the 'create new gene set' procedure shown above, you can add or remove probes.



Set up the gene set
Add or remove probes/genes using the buttons below the table. The list of all possible available probes is provided at the left. The list of probes/genes that are in the gene set is given at right. To find a specific gene use the 'find' tool. If you don't want to make any changes, press 'cancel'.

All available probes and genes

Probe	Gene	Description
39585_at	CCRK	cell cycle related kinase
32706_at	HIRA	HIR histone cell cycle regula...
33811_at	CCPG1	cell cycle progression 1

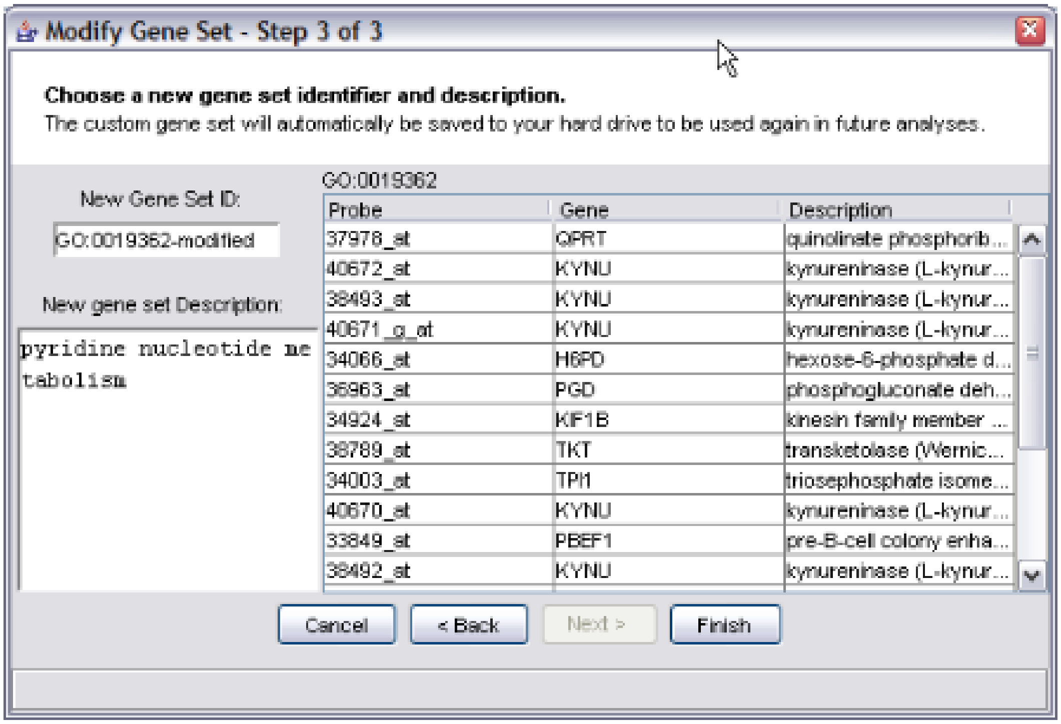
Gene set members

Probe	Gene	Description
37978_at	QPRT	quinolinate phospho...
40672_at	KYNU	kynureninase (L-ky...
38493_at	KYNU	kynureninase (L-ky...
40671_g_at	KYNU	kynureninase (L-ky...
34066_at	H6PD	hexose-6-phosphat...
36963_at	PGD	phosphogluconate ...
34924_at	KIF1B	kinesin family memb

Find: cell cycle

Number of Probes selected: 18

Finally, you can provide your gene set with a new name. This is actually optional: if you don't provide a new name, the old gene set will simply be replaced.



Choose a new gene set identifier and description.

The custom gene set will automatically be saved to your hard drive to be used again in future analyses.

New Gene Set ID: GO:0019362-modified

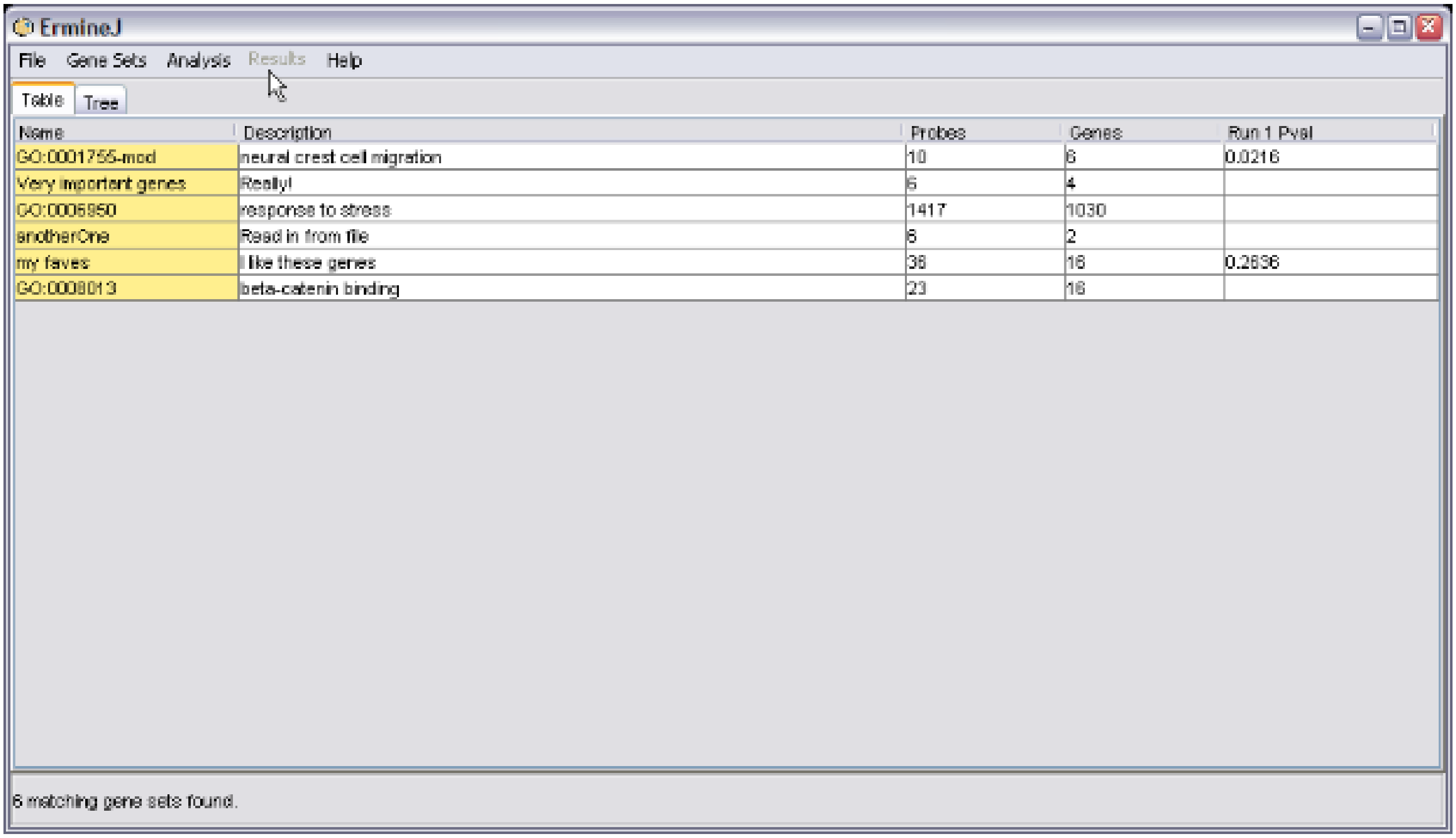
New gene set Description: pyridine nucleotide metabolism

GO:0019362

Probe	Gene	Description
37978_at	QPRT	quinolinate phosphorib...
40672_at	KYNU	kynureninase (L-kynur...
38493_at	KYNU	kynureninase (L-kynur...
40671_g_at	KYNU	kynureninase (L-kynur...
34066_at	H6PD	hexose-6-phosphate d...
36963_at	PGD	phosphogluconate deh...
34924_at	KIF1B	kinesin family member ...
38789_at	TKT	transketolase (Wernic...
34003_at	TPI1	triosephosphate isome...
40670_at	KYNU	kynureninase (L-kynur...
33849_at	PBEF1	pre-B-cell colony enha...
38492_at	KYNU	kynureninase (L-kynur...

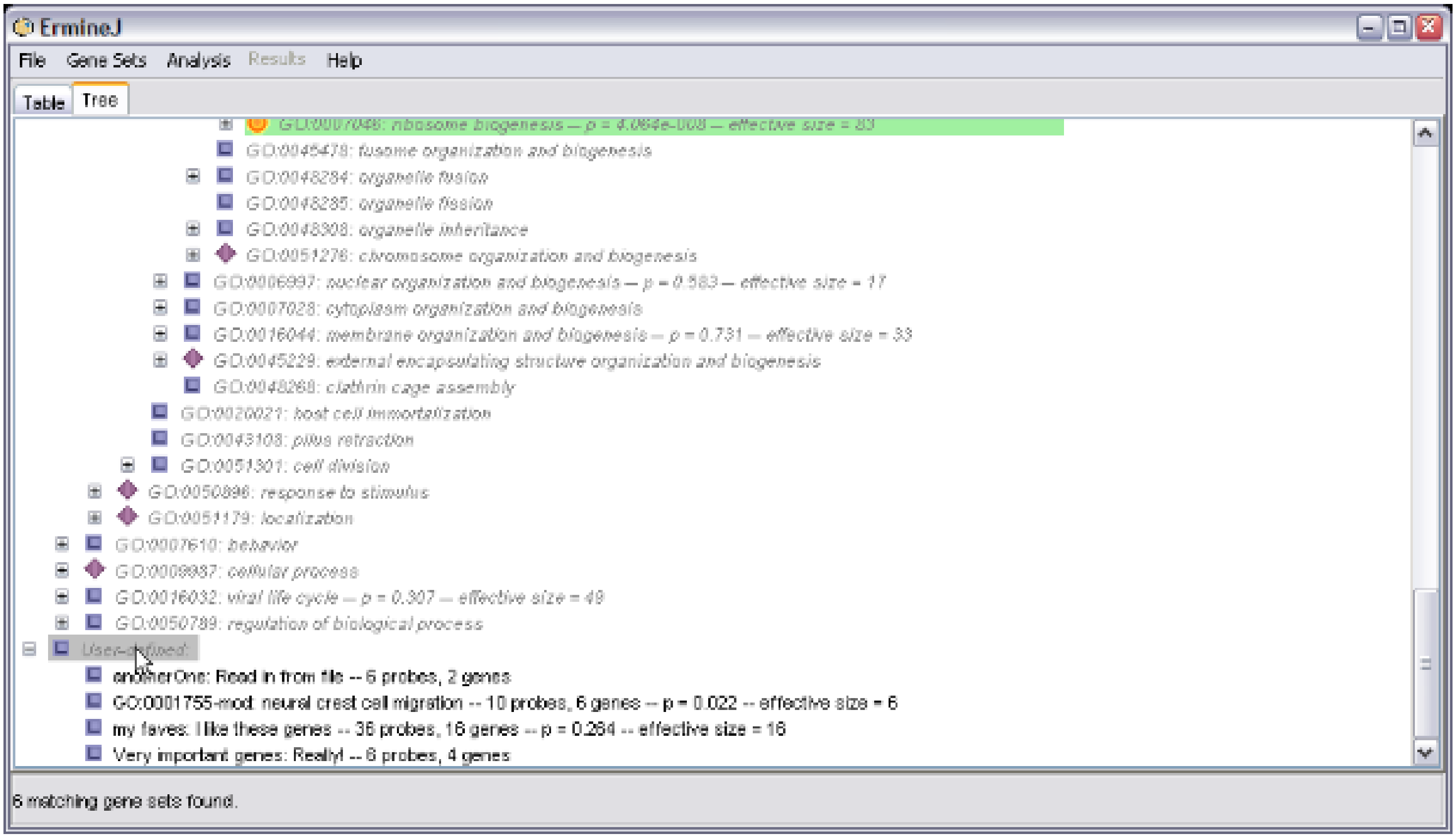
Note!: If you rename a Gene Ontology gene set, next time you start the software, your redefined gene set will always replace the one defined in the GO XML file. See below

As for new gene sets, the modified gene sets are shown on the main panel in a different color to indicate they were modified by the user. Use "ctrl-U" to make them show up on their own, or look in the "user-defined" section of the tree panel.



Name	Description	Probes	Genes	Run 1 Pval
GO:0001755-mod	neural crest cell migration	10	6	0.0216
Very important genes	Really!	6	4	
GO:0006950	response to stress	1417	1030	
anotherOne	Read in from file	6	2	
my faves	I like these genes	36	16	0.2836
GO:0008013	beta-catenin binding	23	16	

6 matching gene sets found.



```
            GO:0007046: ribosome biogenesis -- p = 4.064e-008 -- effective size = 83
            GO:0045478: fusome organization and biogenesis
            GO:0048284: organelle fusion
            GO:0048285: organelle fission
            GO:0048308: organelle inheritance
            GO:0051276: chromosome organization and biogenesis
          GO:0006997: nuclear organization and biogenesis -- p = 0.583 -- effective size = 17
          GO:0007028: cytoplasm organization and biogenesis
          GO:0016044: membrane organization and biogenesis -- p = 0.731 -- effective size = 33
          GO:0045229: external encapsulating structure organization and biogenesis
          GO:0048268: clathrin cage assembly
        GO:0020021: host cell immortalization
        GO:0043108: pilus retraction
        GO:0051301: cell division
      GO:0050896: response to stimulus
      GO:0051179: localization
    GO:0007610: behavior
    GO:0009987: cellular process
    GO:0016032: viral life cycle -- p = 0.307 -- effective size = 49
    GO:0050789: regulation of biological process
  User-defined:
    anotherOne: Read in from file -- 6 probes, 2 genes
    GO:0001755-mod: neural crest cell migration -- 10 probes, 6 genes -- p = 0.022 -- effective size = 6
    my faves: I like these genes -- 36 probes, 16 genes -- p = 0.264 -- effective size = 16
    Very important genes: Really! -- 6 probes, 4 genes
```

6 matching gene sets found.

I don't like my gene set, how do I get rid of it?

You can delete user-defined gene sets using the popup menu in the table or tree view. If you delete a class that is part of the gene ontology (only possible if you modified it), it will disappear for this session, but will be reloaded from the GO XML file next time you start up ermineJ.

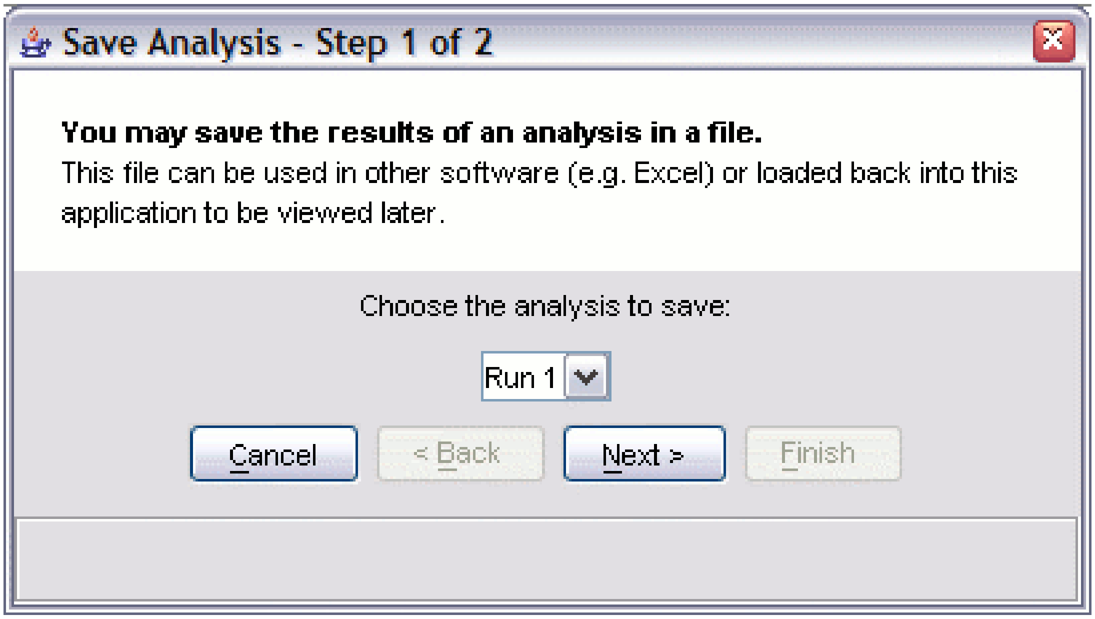
Alternatively, find the place where ermineJ stores gene sets and delete the files by hand.

Saving and loading analyses

You can save an analysis you did and load it again later.

These capabilities are accessed in the main window using the "Analysis" menu.

Saving:



Save Analysis - Step 1 of 2

You may save the results of an analysis in a file.
This file can be used in other software (e.g. Excel) or loaded back into this application to be viewed later.

Choose the analysis to save:

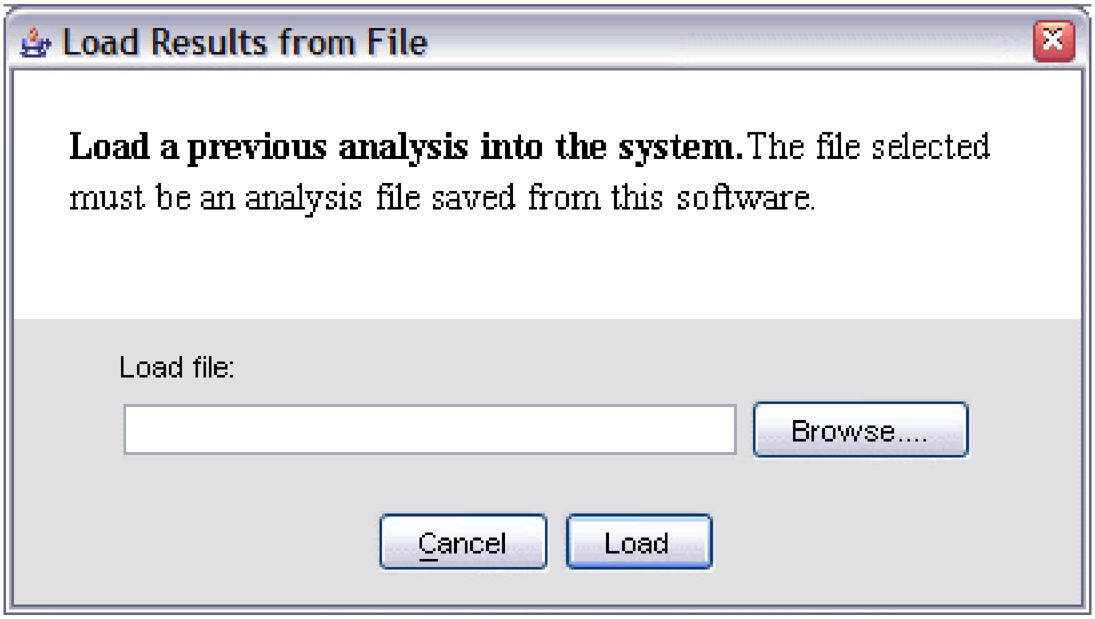
Run 1

You can also optionally check a box to "include all genes" in the output. If you select that, each line in the output file will include a list of genes which are in the gene set.

The output format is described here

Loading an analysis

Once you have saved an analysis, you can load it back in using the "Analysis -> load analysis" item:



Load a previous analysis into the system. The file selected must be an analysis file saved from this software.

Load file:

Note: The locations of files (on your hard drive) used during the analysis are stored in header of the results file. If you move those files (such as the gene score file), reading the analysis back in might not work. In some cases you will be prompted to enter the new locations of the files but to be safe, leave the files where they were when you did the analysis and you won't have this problem. Another option is to edit the header of the results file to reflect the new location, but this is not recommended.

If you delete the files referred to in the header, you will definitely have problems loading the analysis back in! (though the results file is still usable in other software such as excel).

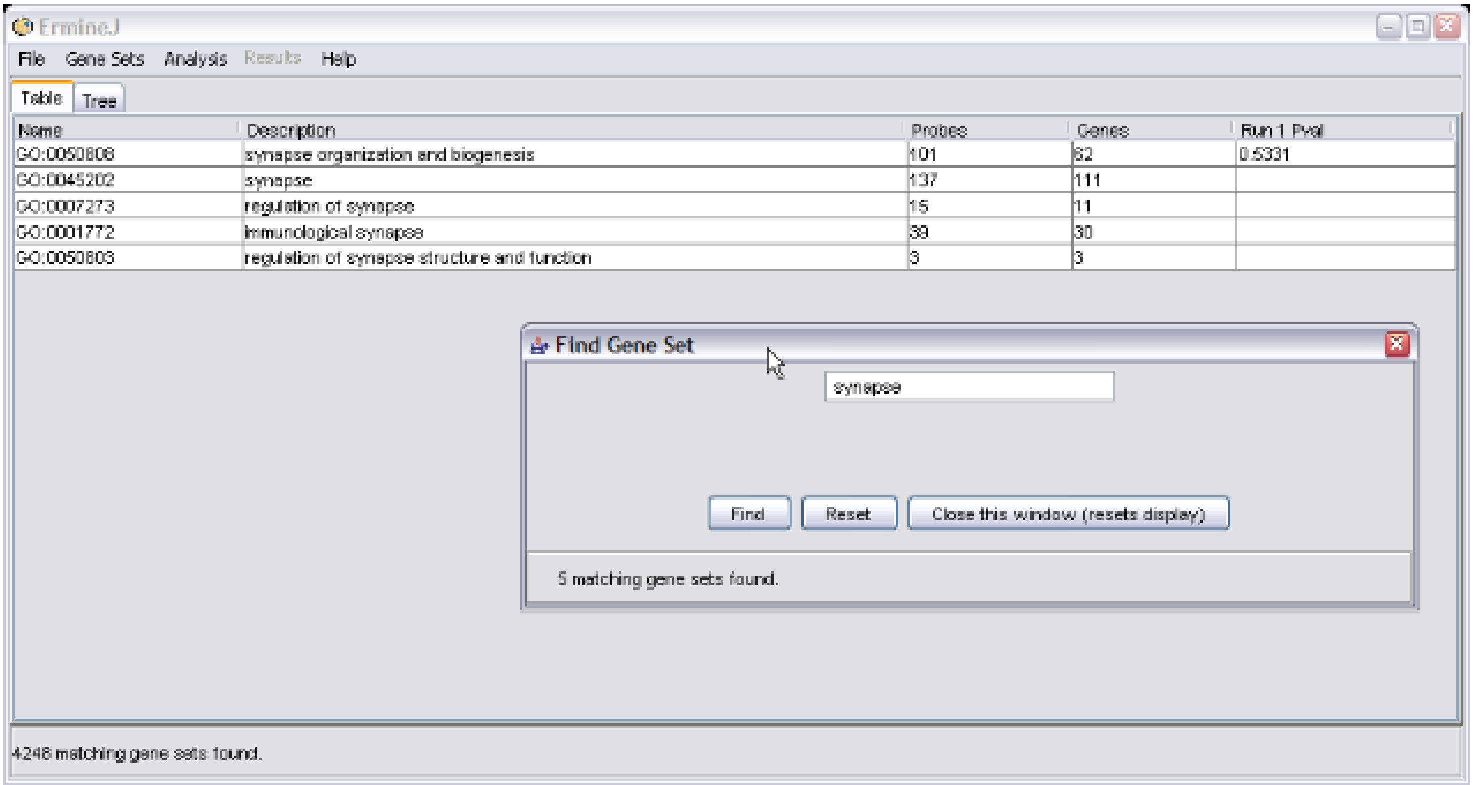
Searching for Gene sets

ErmineJ has several facilities to make it easier to locate gene sets of interest

In the main window, the menu item "GeneSets-->Find gene sets" opens a new dialog box that helps you find gene sets of interest.

Search by gene set

To use it, type text you want to search for and press "Enter" on your keyboard or click on the "Find" button. The main window is updated to show only gene sets that match. In the example we searched for "synapse"



Name	Description	Probes	Genes	Run 1 Pval
GO:0050808	synapse organization and biogenesis	101	82	0.5331
GO:0045202	synapse	137	111	
GO:0007273	regulation of synapse	15	11	
GO:0001772	immunological synapse	39	30	
GO:0050803	regulation of synapse structure and function	3	3	

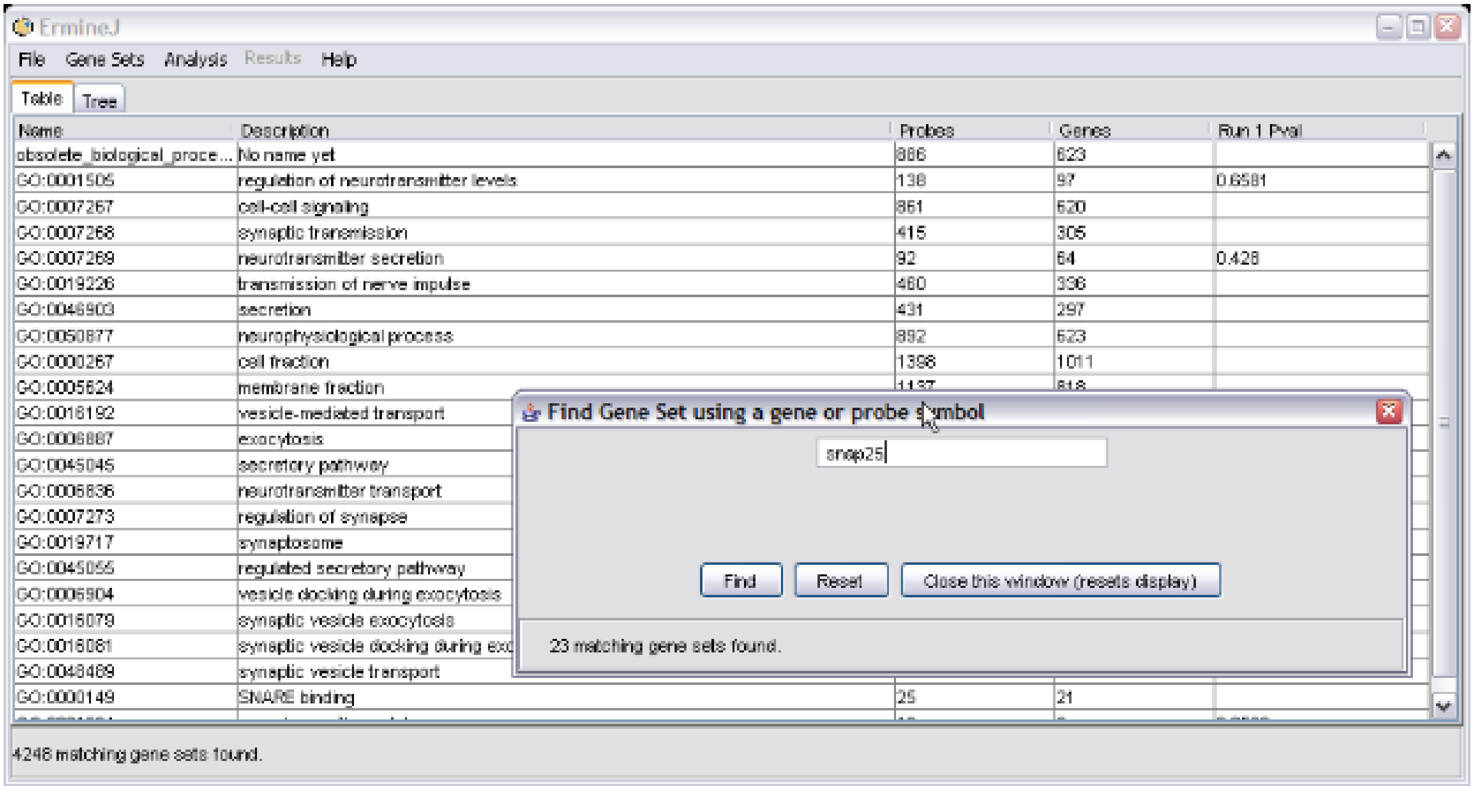
synapse

5 matching gene sets found.

4248 matching gene sets found.

Search by gene

Alternatively, you can search by gene using the "GeneSets --> Find gene sets by gene" menu item. Here is an example of searching for "snap25". All gene sets containing this gene are displayed. Note that you must enter the official gene symbol (capitalization doesn't matter).


ErmineJ
File
Gene Sets
Analysis
Results
Help
Table
Tree
Find Gene Set using a gene or probe symbol
Find
Reset
Close this window (resets display)

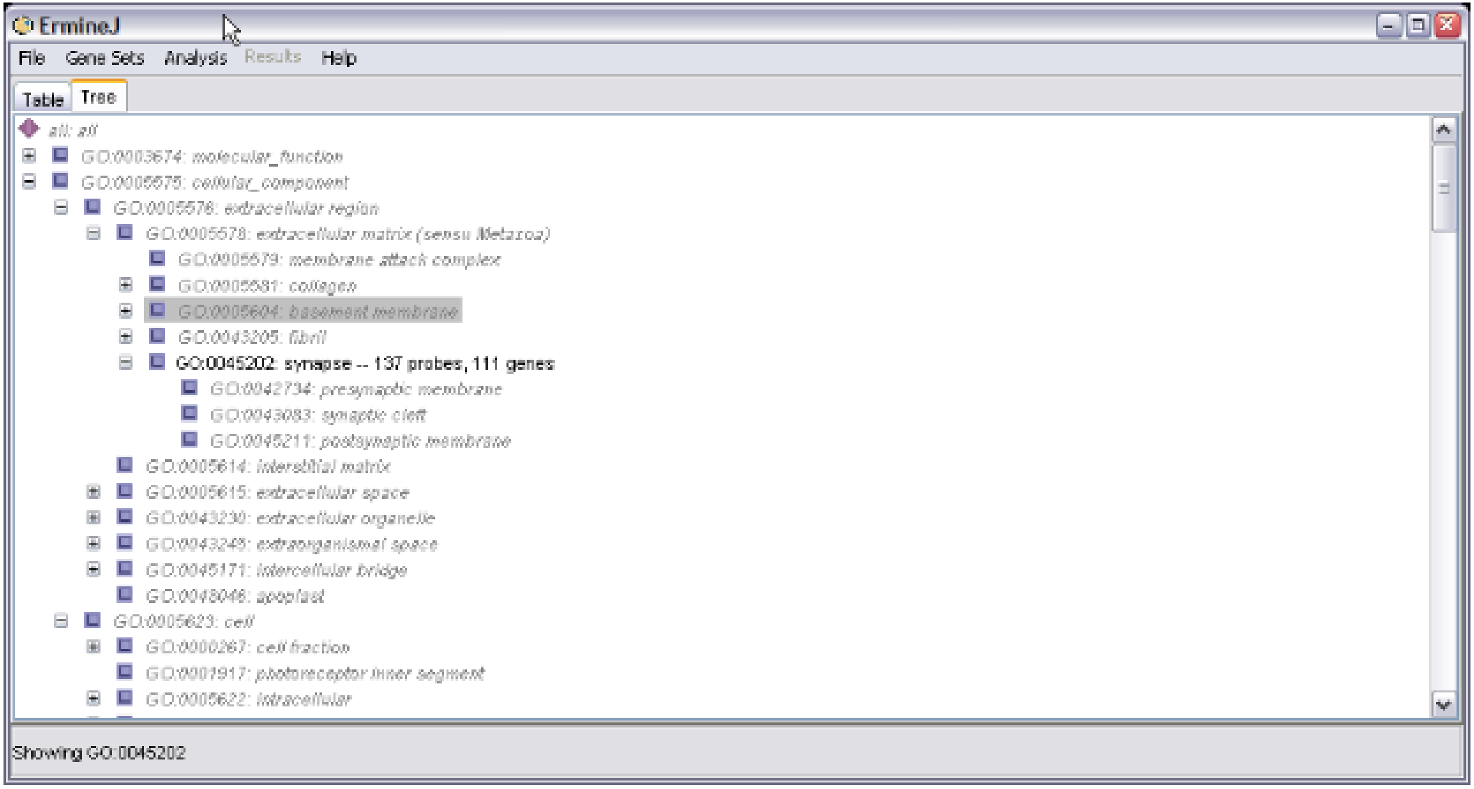
Name	Description	Probes	Genes	Run 1 Pval
obsolete_biological_proce...	No name yet	886	623	
GO:0001505	regulation of neurotransmitter levels	138	97	0.6581
GO:0007267	cell-cell signaling	861	620	
GO:0007268	synaptic transmission	415	305	
GO:0007269	neurotransmitter secretion	92	64	0.428
GO:0019226	transmission of nerve impulse	460	336	
GO:0046903	secretion	431	297	
GO:0050877	neurophysiological process	892	623	
GO:0000267	cell fraction	1398	1011	
GO:0005624	membrane fraction	1137	818	
GO:0016192	vesicle-mediated transport			
GO:0006887	exocytosis			
GO:0045045	secretory pathway			
GO:0006836	neurotransmitter transport			
GO:0007273	regulation of synapse			
GO:0019717	synaptosome			
GO:0045055	regulated secretory pathway			
GO:0006904	vesicle docking during exocytosis			
GO:0016079	synaptic vesicle exocytosis			
GO:0016081	synaptic vesicle docking during exc			
GO:0048489	synaptic vesicle transport			
GO:0000149	SNARE binding	25	21	

snap25

23 matching gene sets found.

4248 matching gene sets found.

In the treeview, the results of a search are indicated by greying out all gene sets that don't meet criteria.



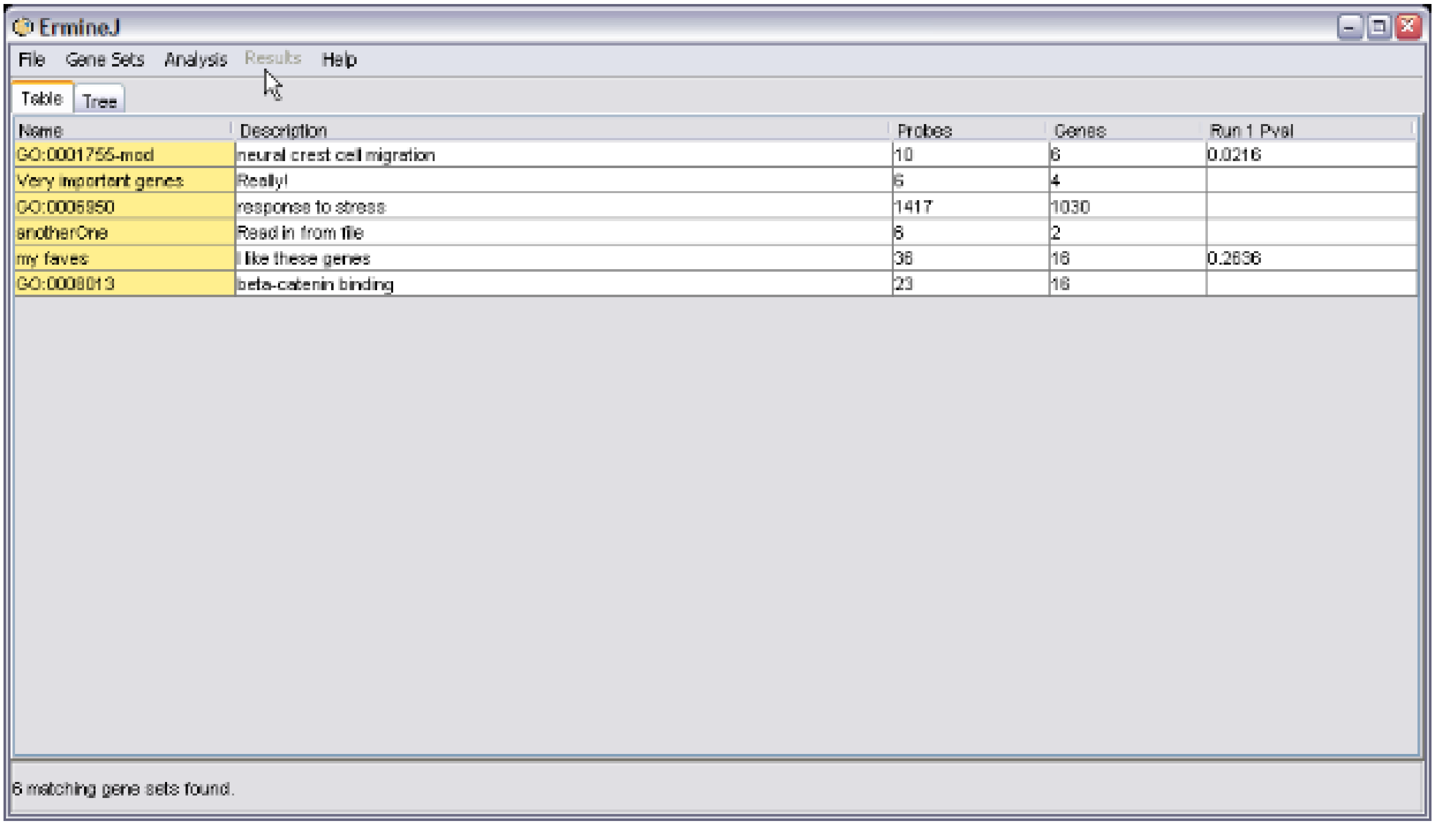
```
all: all
GO:0003674: molecular_function
GO:0005575: cellular_component
  GO:0005576: extracellular region
    GO:0005578: extracellular matrix (sensu Metazoa)
      GO:0005579: membrane attack complex
      GO:0005581: collagen
      GO:0005604: basement membrane
      GO:0043205: fibril
      GO:0045202: synapse -- 137 probes, 111 genes
        GO:0042734: presynaptic membrane
        GO:0043083: synaptic cleft
        GO:0045211: postsynaptic membrane
    GO:0005614: interstitial matrix
    GO:0005615: extracellular space
    GO:0043230: extracellular organelle
    GO:0043245: extraorganismal space
    GO:0045171: intercellular bridge
    GO:0048046: apoplast
  GO:0005623: cell
    GO:0000267: cell fraction
    GO:0001917: photoreceptor inner segment
    GO:0005622: intracellular
```

Showing GO:0045202

The 'Reset' button clears your search and shows all gene sets again. Closing the Find window also resets the display. You can leave the Find window open as you perform other tasks.

Where are the gene sets I defined?

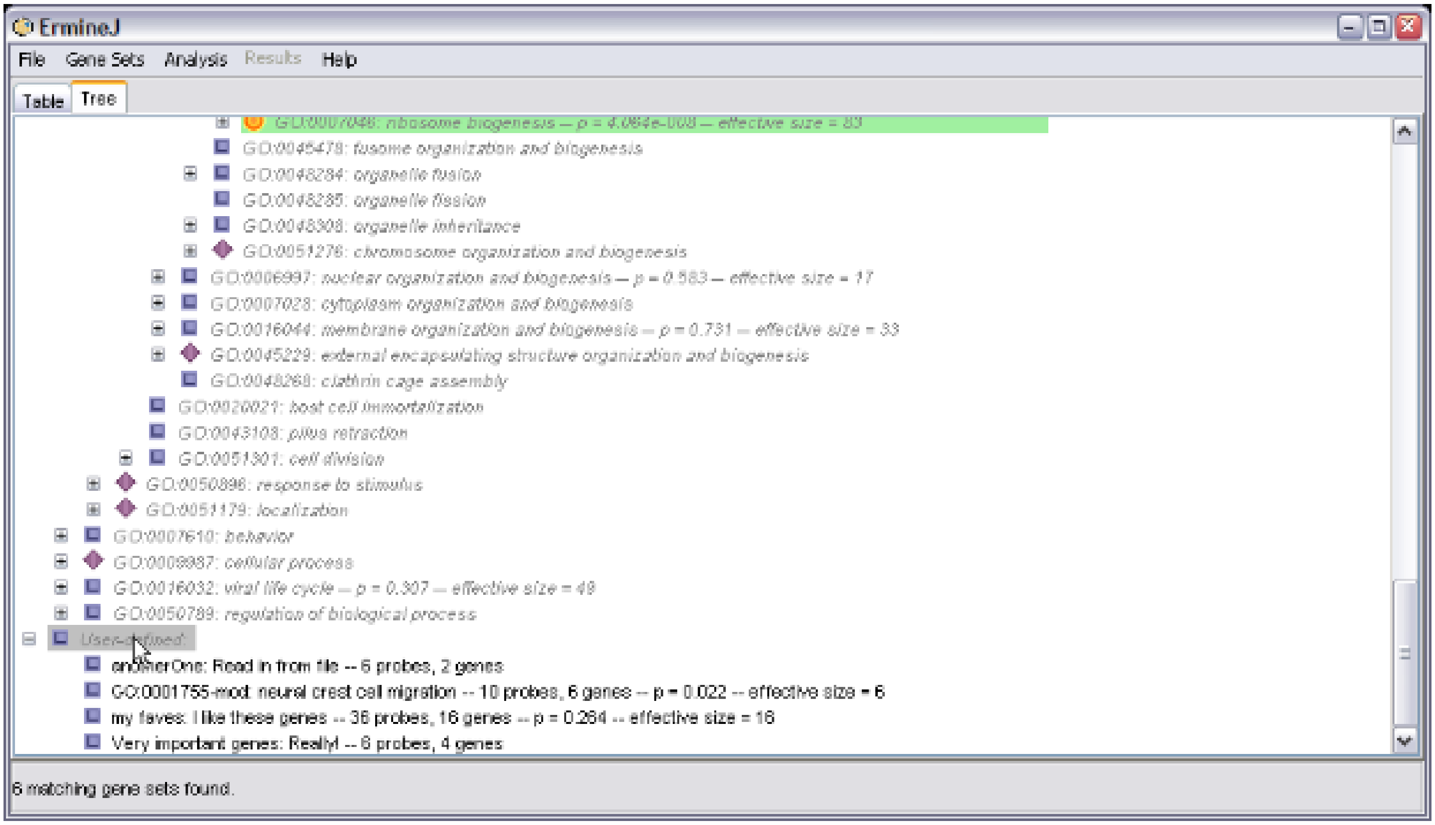
The key "ctrl-U" (or the corresponding menu item) toggles the display between showing all gene sets and just the user-defined gene sets. This is useful because the user-defined gene sets can easily get lost among all the others!



Name	Description	Probes	Genes	Run 1 Pval
GO:0001755-mod	neural crest cell migration	10	6	0.0216
Very important genes	Really!	6	4	
GO:0006950	response to stress	1417	1030	
anotherOne	Read in from file	8	2	
my faves	I like these genes	36	16	0.2636
GO:0008013	beta-catenin binding	23	16	

6 matching gene sets found.

Similarly, in the tree view, there is a separate subtree for user-defined gene sets, at the bottom of the panel.



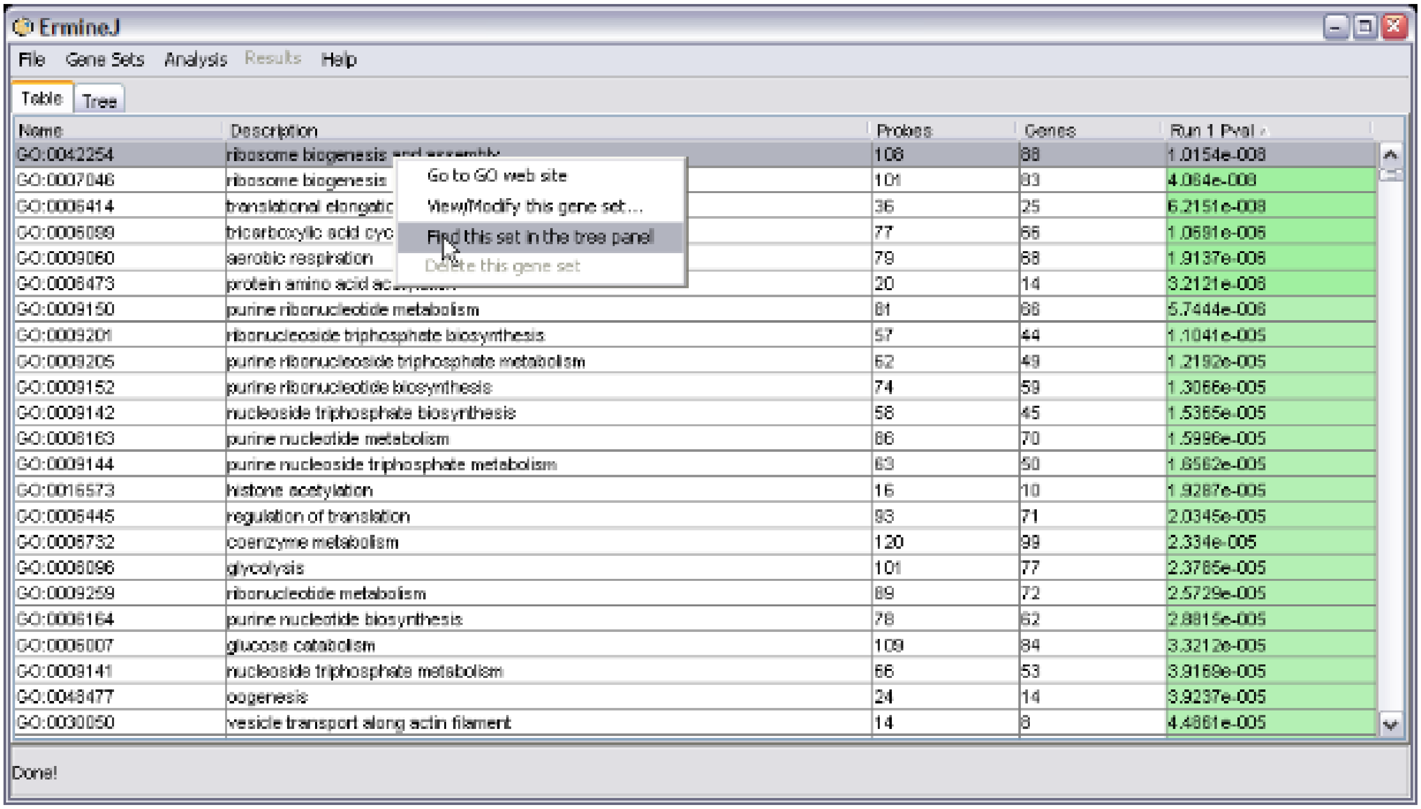
```
          GO:0007046: ribosome biogenesis — p = 4.064e-008 — effective size = 83
          GO:0045478: fusome organization and biogenesis
          GO:0048284: organelle fusion
          GO:0048285: organelle fission
          GO:0048308: organelle inheritance
          GO:0051276: chromosome organization and biogenesis
        GO:0006997: nuclear organization and biogenesis — p = 0.583 — effective size = 17
        GO:0007028: cytoplasm organization and biogenesis
        GO:0016044: membrane organization and biogenesis — p = 0.731 — effective size = 33
        GO:0045229: external encapsulating structure organization and biogenesis
        GO:0048268: clathrin cage assembly
      GO:0020021: host cell immortalization
      GO:0043108: pilus retraction
      GO:0051301: cell division
    GO:0050896: response to stimulus
    GO:0051179: localization
  GO:0007610: behavior
  GO:0009987: cellular process
  GO:0016032: viral life cycle — p = 0.307 — effective size = 49
  GO:0050789: regulation of biological process
User-defined:
  anotherOne: Read in from file -- 6 probes, 2 genes
  GO:0001755-mod: neural crest cell migration -- 10 probes, 6 genes -- p = 0.022 -- effective size = 6
  my faves: I like these genes -- 36 probes, 16 genes -- p = 0.264 -- effective size = 16
  Very important genes: Really! -- 6 probes, 4 genes
```

6 matching gene sets found.

How do I find gene sets in the tree?

It can be hard to find a gene set in the tree because they can be anywhere in the hierarchy. Because searches generally can yield multiple gene sets, currently the search function doesn't work the way you might think in the tree view. Instead, locate the gene set you want in the table, and then use the context-specific menu (right-click or option-click) to choose the "Find this set in the tree panel" option.

This switches you to the tree view and highlights the gene set you requested.



Name	Description	Probes	Genes	Run 1 Pval
GO:0042254	ribosome biogenesis and assembly	108	88	1.0154e-008
GO:0007046	ribosome biogenesis	101	83	4.064e-008
GO:0006414	translational elongatic	36	25	6.2151e-008
GO:0006099	tricarboxylic acid cyc	77	66	1.0691e-006
GO:0009060	aerobic respiration	79	68	1.9137e-006
GO:0006473	protein amino acid ac	20	14	3.2121e-006
GO:0009150	purine ribonucleotide metabolism	81	66	5.7444e-006
GO:0009201	ribonucleoside triphosphate biosynthesis	57	44	1.1041e-005
GO:0009205	purine ribonucleoside triphosphate metabolism	62	49	1.2192e-005
GO:0009152	purine ribonucleotide biosynthesis	74	59	1.3066e-005
GO:0009142	nucleoside triphosphate biosynthesis	58	45	1.5365e-005
GO:0006163	purine nucleotide metabolism	86	70	1.5996e-005
GO:0009144	purine nucleoside triphosphate metabolism	63	50	1.6562e-005
GO:0016573	histone acetylation	16	10	1.9287e-005
GO:0006445	regulation of translation	93	71	2.0345e-005
GO:0006732	coenzyme metabolism	120	99	2.334e-005
GO:0006096	glycolysis	101	77	2.3765e-005
GO:0009259	ribonucleotide metabolism	89	72	2.5729e-005
GO:0006164	purine nucleotide biosynthesis	78	62	2.8815e-005
GO:0006007	glucose catabolism	109	84	3.3212e-005
GO:0009141	nucleoside triphosphate metabolism	66	53	3.9169e-005
GO:0048477	oogenesis	24	14	3.9237e-005
GO:0030050	vesicle transport along actin filament	14	8	4.4661e-005

Done!

Input files

ErmineJ uses files provided by the user together with files we provide to analyze your data. ErmineJ accepts four different types of input files, which you should read about by following these links:

- Gene scores - supplied by you (optional for correlation scores)
- Gene expression profiles - supplied by you (optional except for correlation scores)
- Gene Set descriptions (GO) - required, supplied by the GO consortium.
- Gene annotations - required, supplied by us (though in principle you can provide your own).

In addition, individual **gene sets** can be defined or modified by the user. These are saved in files described here.

Finally, there is an explanation of the output format.

Please note that all files are text files, not Excel spreadsheets or other binary formatted files. To convert your files to text in Excel, see these instructions .

Input file formats: gene scores

A "gene score" is any value that is applied to genes in a microarray experiment and which represents some measure of "quality" or "interest". Examples might be a t-test p value or fold change. These scores must be computed separately and then supplied to the ermineJ software for analysis.

! Make sure your gene scores are on a sensible scale

- **If your gene score are raw p-values (most common case)**, then you should be sure to either -log transform your values, or use the -log options in the ermineJ software.
- **If your gene scores are "fold-change",** you might want to use the absolute value of log(fold change).
- **If your gene scores are NOT raw p-values,** then make sure you select the right combination of settings to get your data interpreted correctly.

! You need a complete set of gene scores, not just the "selected gene list"

Unlike some software packages, ermineJ requires a complete set of gene scores, rather than just those for the "selected genes". Thus, if your microarray has 12,000 probe sets on it, you will provide 12,000 gene scores. A caveat to this is that if you have filtered your data to exclude "unexpressed" genes (for example), you might only have gene scores for some probes. This is perfectly fine and the analysis will be based only on the probes for which you provide data. However, the analysis is really only valid if you include all the probes that you performed your gene selection analysis on.

The resampling-based analysis requires the gene scores, while for other analyses it is not required.

! Make sure your gene scores are on the right format

The gene score file is a **simple tab-delimited text file** , minimally having just two columns (like this example , which is from the HG-U95A microarray design). For a simple case please use the following rules:

- The first column contains the probe identifiers
- The second column contains the scores. **Non-numeric values such as "NaN" or "#NUM!" are interpreted to equal zero.** If you want to avoid this you should remove those rows from your file entirely.
- A one-line header is expected (if you don't include the header, the first line of data will be skipped).
- Files with more than two columns are also fine.
- The file **CANNOT be an Excel spreadsheet**. Use "Save as... text" in Excel. For instructions on doing, see this page .

The table below shows the basic idea.

Probes	ProbeScore
117_at	0.593537874
1007_s_at	0.0643101
...	... (etc.)

Input file formats: microarray data

The "raw" gene expression profile data can optionally be loaded into the software for visualization and/or analysis. The correlation-based analysis requires the expression profile data, while for other analyses it is not needed.

The data file is a simple tab-delimited text file, with each row representing the dependent variable measurements (e.g., expression levels or ratios) for one set of observations. Each column represents an observation or sample (e.g., a microarray).

Quick tips to avoid common file format problems:

- **The input files are *tab* delimited**. Comma or space-delimited files will not work. Exporting files as text from Excel is a good way to produce appropriate files.
- **Missing values are ok**, but they should be indicated by blanks, not by "NA" or other non-numerical characters. In Excel, they just look like blank cells.
- **Notice the 'corner' string in the example below** - all columns including the example names have a heading. It does not matter what you put in the corner, but it must not be blank. The parser uses the header to figure out how many features you have, so if you skip the corner string it will appear that you have extra data, resulting in an error message.
- **You can only have one column of descriptors, all other data must be your numeric feature data**. In other words, don't include extra columns in your file that are not part of the data or the example labels. Extra columns will either result in an error (most likely) or invalid results (if your extra columns look like data).
- This means that each row represents the expression measurements for one gene. The columns then represent different arrays which were run. It helps in later analysis if the data columns are arranged by condition: for example, put the "wild type" columns all together and the "mutant" columns all together after that. So the top of your data file might look like this (when nicely formatted)

gene	mutant	mutant	mutant	wildtype	wildtype	wildtype
100001_at	-36.3	77.8	64.4	89.4	126.6	86.2
100002_at	1504.2	1512	944.5	1157.9	1652	1358.9
100003_at	845.9	966.5	1057.4	987.4	764.1	878.5
100004_at	2304.4	1991.1	2783.7	1929.8	2236.8	2664.1
100005_at	3826.5	2876.9	4514.1	3187.8	2454.3	3730.6
100006_at	3635	2584.6	3554.9	2810.9	1629	2248.6

100007_at	6328.4	6197.8	7236.4	6224.9	6950	6206.8
100009_r_at	6580.6	8715.9	5280.3	6569.4	8513.4	7236
100010_at	368.2	344.5	-62.4	200	282.7	583.4
100011_at	1949.7	2511.3	1937.8	2684.1	1722.5	2101.3
100012_at	3145.6	2936.7	3358.4	4250.8	2706.4	2776
100013_at	-1098.4	-720.8	-1418.8	-886.9	-764.4	-1247.6
100014_at	1108	1197	985.4	1216.7	1328.1	1161.5
100015_at	6005	1040.6	4434.1	1069.4	864.8	2617.4
100016_at	4485.3	3236.2	4910.2	3474.6	3447.1	3493
100017_at	497.5	399.3	964.2	347.7	524.5	561.3
100018_at	540	1209.7	811.1	1880.8	317.9	587.8
100019_at	-303.5	46.4	0.9	53.4	-252.6	-346.9
100020_at	1606.3	1570.4	1996.6	3319.7	1803.4	1811.7
100021_at	1349.8	1193.5	764.7	331.5	1175	783.9

(etc, possibly for many more lines)

There are a number of things to be careful about in setting up your file

- It must be plain text. In excel, use the "file->save as..." menu selection and then choose "text (tab delimited)" option. Excel will warn you about keeping the file in this format; this is just an annoyance but can be ignored. See "Saving excel files as text" for ridiculously detailed instructions.
- All columns and rows must have something written in them, including the "upper left corner".
- The values in the file, other than the row and column labels, must be numbers (no letter or other characters). This means that there can be no other columns of text.

ermineJ

Input files: Gene Ontology XML

ErmineJ currently uses the Gene Ontology RDF XML* file to obtain a complete set of data about GO terms and their definitions. We distribute a recent version of this file with the software. If you want to upgrade to a newer version, go to http://www.godatabase.org/dev/database/archive/latest/ and select the latest "go_XXXXXX-termdb.rdf-xml.gz" file (not the very large assocdb file). This file should be placed in your ermineJ.data/genesetdata directory.

As of version 2.1, the files can be gzipped or zipped. There is no need to unpack them.

If the link above is broken, go to http://www.godatabase.org/ and look for the download link -- and let us know

Older versions of the GO files should be available as well. Problems with getting the GO XML files should be reported to the GO web site maintainers or the go-database mailing list (not us!).

Note that the annotation file you use is not assured to be 'in synch' with the GO XML file you download. This matters because the GO terms change, and it is possible for an annotation file to have terms which are newer than the GO XML file you use.

If you are using the annotation files we provide, you are advised use the GO XML file that was used during the latest annotation build. A link to that file is maintained at http://microarray.cpmc.columbia.edu/annots/ .

If you are using an annotation file provided by a third party, or which you made yourself, we advise that you try to use the same version of GO that was used to created the annotations.

*If you don't know what RDF or XML mean, don't worry, you don't need to know.

File information: gene annotations

You can use two types of annotation files, ones we provide on our web site, or ones provided by Affymetrix for their microarray platforms. Requests for support of other formats will gladly be entertained. You can also use files you have created, so long as they follow one of the accepted formats.

You inform ermineJ of which format you are using with the pull-down menu used at startup.

As of version 2.1, the files can be gzipped or zipped. There is no need to unpack them.

Using files we provide

In ermineJ, we refer to these as "ermineJ format", but the files are very simple and useful in other contexts

You can download annotation files here (http://microarray.cpmc.columbia.edu/annots/)

We provide annotation files for some popular (and some not-so-popular) array formats. These files contain the probe (or probe set) identifiers, the gene symbols and names, and gene set membership information. For our current annotations, this means that a list of the Gene Ontology terms associated with a gene are listed. For each term, the 'parent' terms are also listed, so that genes associated with very specific terms are also included in the less specific categories.

For species or platforms we don't support, ask us for assistance or set it up yourself. The files are not hard to prepare if you have Gene Ontology (or other gene set descriptor) annotations available.

For species we support, but for new platforms, often you will be able to create a new annotation file by pulling information out of our existing files using a simple Perl script.

Using files from the Affymetrix web site

Obviously this only helps you if you are using an Affymetrix GeneChip. (The writers of ermineJ are not affiliated with Affymetrix in any way).

The files we tested our software on were obtained from the Affymetrix site and are in CSV (comma-separated value) format. As of May 2005 ermineJ worked with these files, but if Affymetrix changes the format (they have in the past) it is possible that they will no longer work with ermineJ.

To use this, select the "Affy CSV" option on the startup dialog.

Using files you create

Annotation files that you created can be used so long as they adhere to one of the accepted formats. There are a few things to consider:

- The probe IDs must exactly match the ones you provide in your Gene score file. Any probes not having an entry will be ignored.
- The gene symbols are used internally by the software to decide which genes are present on the array more than once. Therefore, if two probes refer to the same gene, make sure the symbol you use is the same for both probes. (It doesn't matter what the symbol is).
- The gene names or descriptions are optional, and blank values will just show up as "No description" or something similar.
- In the ermineJ format, the GO ids must be in "long" format (with the GO: prefix). The GO terms themselves should be omitted. The parents of all terms listed are automatically included in the analysis (subject to other constraints such as the maximum Gene Set size you set in the analysis), so there is no need to list these explicitly.

Gene set files

User-defined gene sets can be created from within ermineJ, or imported using two different format. This page has instructions on how to import your own files. You can either use the ermineJ-native format, or import a simple list of genes .

Option 1: Use ermineJ-native format and put the files in the genesets directory

This method might be useful if you have a lot of gene sets to load, have lists in terms of probes (not genes), or are moving an ermineJ installation from one machine to another and are copying the custom gene sets over.

Gene sets created in ermineJ are saved in, and loaded from, a simple text file format. Knowledge of the format gives you the ability to define gene sets outside of ermineJ. When ermineJ starts up, it looks for these files in a predefined location (see below) and loads them.

Here is a sample:

```
probe
MyGeneSet
Genes I Like
36495_at
271_s_at
37983_at
34071_at
128_at
129_g_at
206_at
38466_at
32017_at
346_s_at
32018_at
```

(or download the sample as a file)

How ermineJ stores your results

When you save your results to a file from ermineJ, what you get is a plain text file that can be opened in Excel or a similar program. Here is a sample.

If you intend to load the results back into ermineJ at some future point, we recommend not editing this file. Instead, make a copy after you load it into excel.

The file is tab-delimited and has three parts:

1. A header, which contains information about the configuration used during analysis. This information is used by ermineJ to recreate the conditions that existed after the analysis was run. Most of it is human-readable. Currently some settings are encoded as numbers (e.g.,"rawScoreMethod = 0", which means that the mean of gene scores is taken during analysis). Future versions of ermineJ may changes this. Note that the number of lines in the header is not fixed and may change in future versions of ermineJ.
2. Column headers, which are preceded by a "#". For explanations of these values, see below.
3. The values, one row for each gene set. Note that due to numerical roundoff, when you reload an analysis into ermineJ you may lose some precision in the pvalues, for example. The

The columns in the output file are:

1. ! - a column of !'s. These are here for boring reasons explained below.
2. Name - the name of the gene set
3. ID - the id of the gene set
4. Probes - the number of probes in the gene set ("size" in earlier versions)
5. Genes - the number of genes in the gene set ("effective_size" in earlier versions)
6. RawScore - the raw statistic for the gene set
7. Pval - the p value for the gene set
8. CorrectedPvalue - the corrected p pvalue
9. Same as - a list of gene sets which have the exact same members as this one. Such gene sets are not listed anywhere else.
10. Similar to - Currently not used
11. Genes - If you selected the "Include genes" option when saving, this will contain a list of the genes that are in the gene set, separated by "|".

Note to programmers: ErmineJ uses Apache Commons configuration to manage properties files. The results file is treated as a properties file when it is reloaded, but we're only interested in the header. Therefore lines that are not properties must be commented out. This is why there are lines that start with # and !. We use ! instead of # to help parsers distinguish between data lines and header lines. One of these days we'll fix this oddity.

Gene set files created by ermineJ are saved in the directory `ermineJ.data/genesets`. This directory is installed in the directory where ermineJ is installed (e.g., C:/ `Program Files/ermineJ/ermineJ.data/genesets`). You should place your own "handmade" gene set files in this location so they are automatically visible to the software.

Note! If you create a gene set when using one microarray design, and then switch to another next time you run ermineJ, ermineJ will try to load your old gene sets. If any probes on the previous design match the identifiers on the current one, the gene set will be loaded to the extent it can. We may change this in a future version of ermineJ.

The full description of the format is:

- Each file describes just one gene set.
- The file is plain text (ASCII)
- The first line describes the type of identifier in the file and is either "probe" or "gene".
- The second line is the name of the gene set.
- The second line is a description of the gene set. There is no limit to the length of this description but in practice it should be just a few words.
- The third and subsequent lines are the identifiers (probe ids or official gene names).

Option 2: Import files containing lists of genes using the "Define new gene set" menu item

This method has the benefit of requiring a very simple format, but you must load the files one at a time using ermineJ. (If this is a pain, a simple Perl or Python script can convert the lists into the other format.)

The file in this case is just a list of genes, with one on each line. The names must be the gene symbols that are used in your gene annotation file. Other symbols will be ignored. Here's an example with just three genes:

```
alox12b
ALOX15
alox12
```

A full description of the file format is:

- Each file describes just one gene set.
- The file is plain text (ASCII)
- Each line contains the official symbol of one gene
- Capitalization is ignored
- Blank lines and symbols not found in the current array design are ignored

On loading in, the list of genes is converted to a list of probes. You will be given the chance to edit the gene list and give it a name before finalizing it

Handling repeated genes

Any given gene can occur more than once in a microarray data set. This is due to the occurrence of two or more probes (or probe sets) that target the same gene. On most microarray designs we have looked at, somewhere in the neighborhood of 30 percent of the probes are "repeats", though this depends on the design. We refer to these as "gene replicates", because they provide replicate measures of the same gene. However, these "replicates" may not be equivalent: they may target different splice forms, or have different sensitivity or specificity. In some cases a probe set may not work at all and give very poor signals, while another probe for the same gene gives a robust signal.

In a gene set score analysis, it does not make sense to count each of these "replicates" independently when determining the size of a gene set: a gene set that consists of five replicates of the same gene should not be considered five genes in the statistical analysis. Thus in ermineJ a gene is only counted once. For the correlation analysis, each pair of genes is only counted once. This means that even if there are five occurrences of a gene on the array, it will only be counted once.

What remains to determine is how to summarize the results for the replicates: the five replicates in our example have to be distilled down to a single value.

ErmineJ offers two ways of dealing with this situation, one which is conservative and one which, while less conservative, might be sensible when there is uncertainty as to the reliability of individual replicates.

The conservative choice is referred to as "**Mean**": the different occurrences of the gene are each given equal weight. For correlation analysis, the same rule applies but at the level of pairs of genes. Thus if there are two replicates of gene A and two replicates of gene B, a total of 4 comparisons is possible. The final contribution of the A-B comparison is 1/4 of the correlations measured between all the replicates.

The less conservative choice is referred to as "**Best**": the only score counted for a gene is the best one. Thus if a gene has two replicates with scores 3 and 4 (- log p values), then only the 4 is counted; with the mean method the final value would be 3.5. For correlation analysis, the best pairwise correlation is stored. In the A-B example give above, this means that the best of the four comparisons is kept.

The choice of which method to use depends on how conservative you want to be, combined with knowledge you have about the microarray design. If you consider gene "replicates" not to really be replicates, then using the "best" option might make sense. If your microarray design just has exact replicate spots of the same sequence, then using "Mean" might be sensible.

Speeding up resampling

To make resampling faster, we have implemented two "tweaks" to the process. In the current implementation of the software, you can turn these both on or both off at the same time only. The control is the "*Always use full resampling"* checkbox in the analysis setup wizard (step 5).

1. Sparse sampling of larger gene set sizes

The background distribution of scores tends to get narrower for larger gene set sizes. Intuitively, if you are only choosing two items from the data, the possibility of getting an extreme value for their average is much higher than if you chose 100 items.

This implies that we can't just use one background distribution for all gene sets.

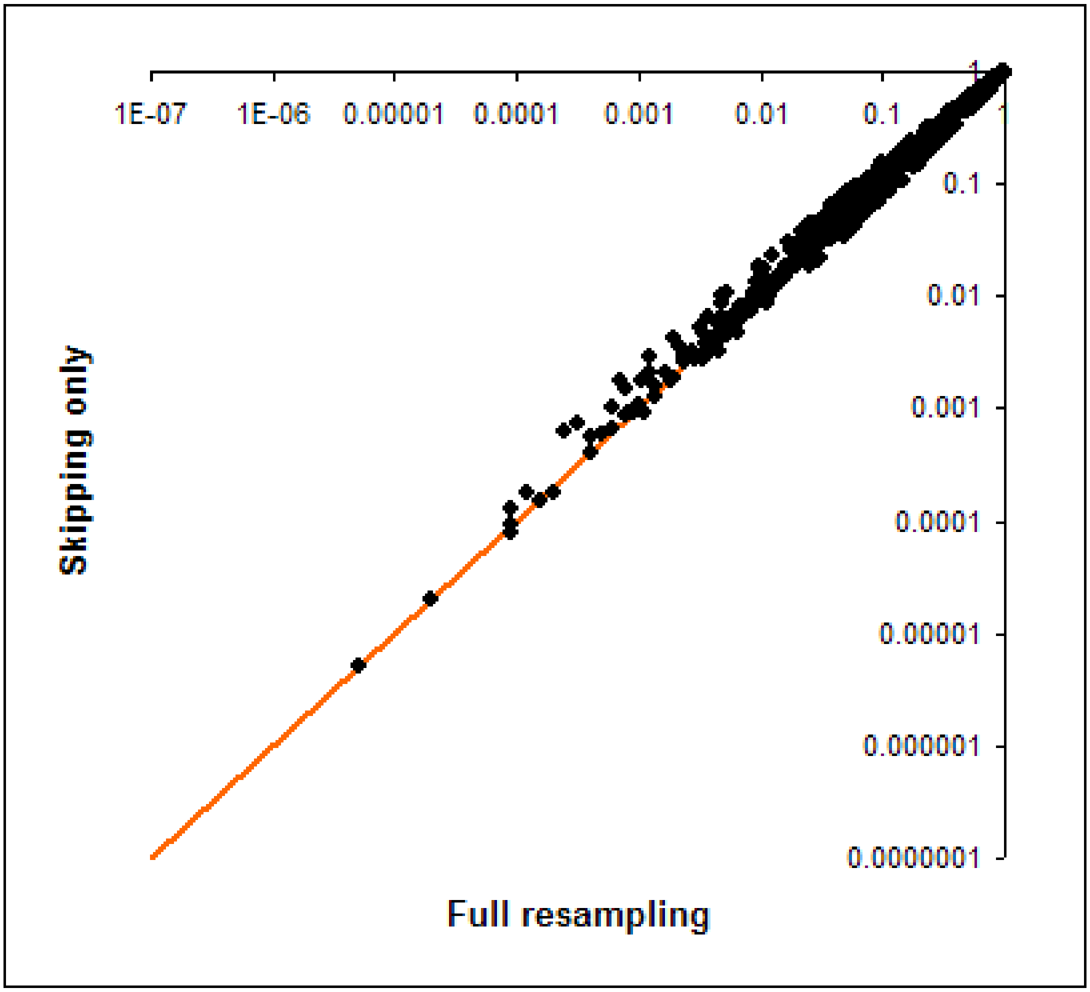
In our original implementation of the algorithm, a background distribution was generated for each gene set size. Thus if you analyze sizes from 4 to 100, 97 different resampling runs have to be performed.

Because each run requires computation of a summary statistic for *n* items, where *n* is the gene set size, the time this takes increases as the gene set sizes increase. (increasing on the order of n^2 for the correlation scores) .

To reduce this problem, we make use of the fact that the difference in the null distribution for a gene set of size 80 is not much different than the one for gene set size 83. The effect of small differences gene set sizes diminishes as gene set sizes increase.

Thus, one optimization is to skip some gene sets sizes when examining large gene set sizes. Currently the exact way we do this is not settable by the user (other than to turn it off). At this writing the implementation is to start skipping some sizes when gene set sizes hit 21, and to increase the step size by 0.1 times the current gene set size. Thus the progression is 20, 23, 26... and later ...69, 76, 84, 93...

The following graph shows the difference this makes for a gene-score based analysis. Note that the axes are on a log scale. This shows that for the most part, the difference in the p-value you get is minor.



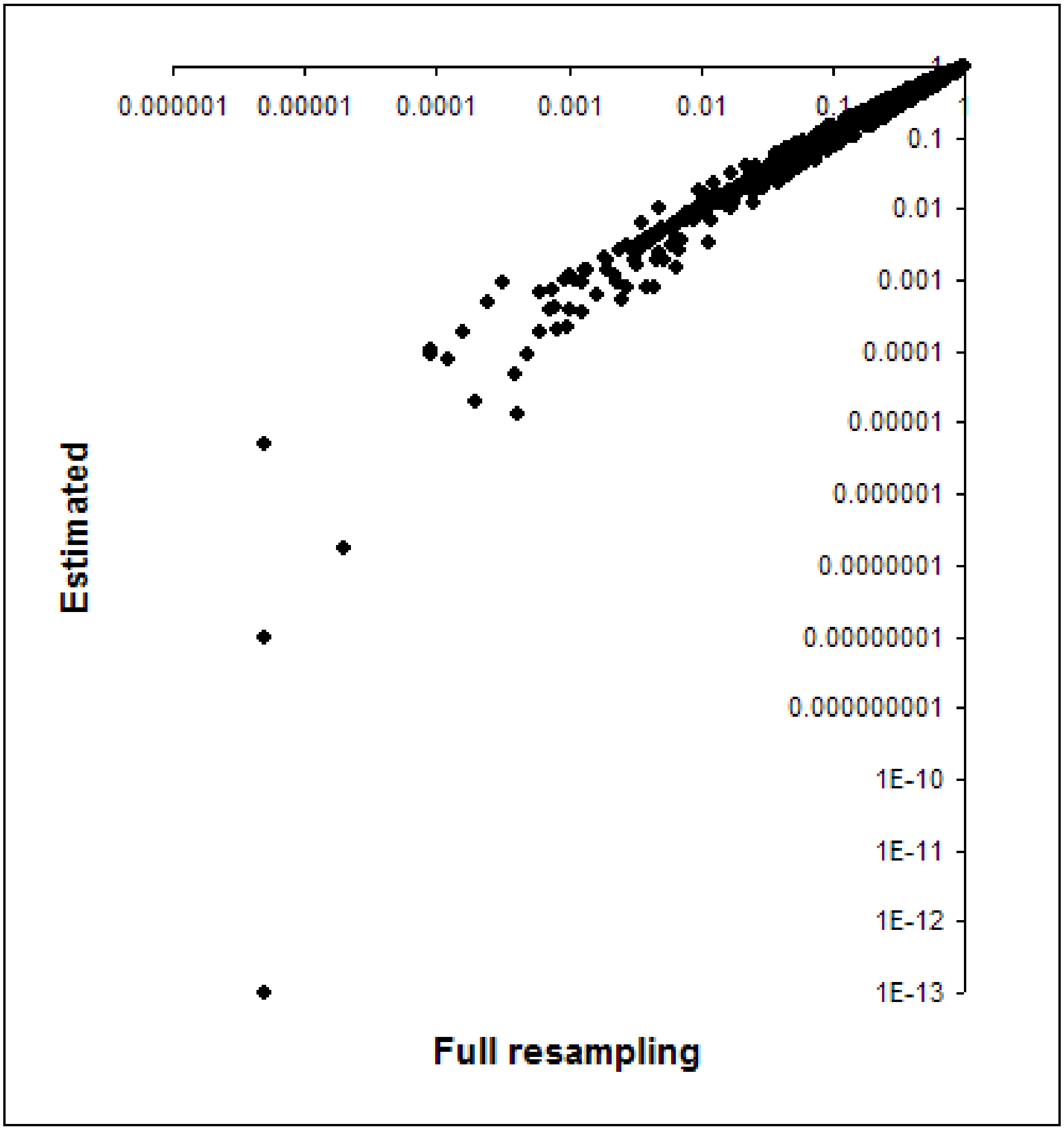
2. Approximating with normal distributions.

The next approximation makes use of the central limit theorem, which states (roughly) that the distribution of means sampled from a population will tend to be normal. If you don't know what that means, don't worry.

To implement this, we start by using random samples, but stop when the mean and variance of the "fitted" normal distribution converge (stop changing).

As gene set sizes increase, the agreement with the normal is bound to improve. In the current implementation, the approximation kicks in for gene sets of size 10 or greater for correlation-based scoring, and 30 or greater for gene score-based analysis.

The next graph shows a comparison of the gene set p values obtained using pure resampling (100,000 samples). This shows that using this approximation tends to yield better p values (though not always). This is partly because the fit isn't perfect, but also because the analytic distribution can yield better p values than the empirical one (which is limited by the number of samples taken; in fact here we set pvalues that came out as zero to a fixed low value so they would plot on a log scale). However, overall the results are reasonable, the ranking of gene sets will be similar, and the speed increase (especially for the correlation-based scoring) is dramatic.



Using the "Median" method for raw scores

Strictly speaking, the central limit theorem doesn't hold for samples of the median of a population. However, for this type of data it should work reasonably well. We have not tested this extensively.

Why are the number of gene and probes shown in the output panel incorrect when my data have been filtered?

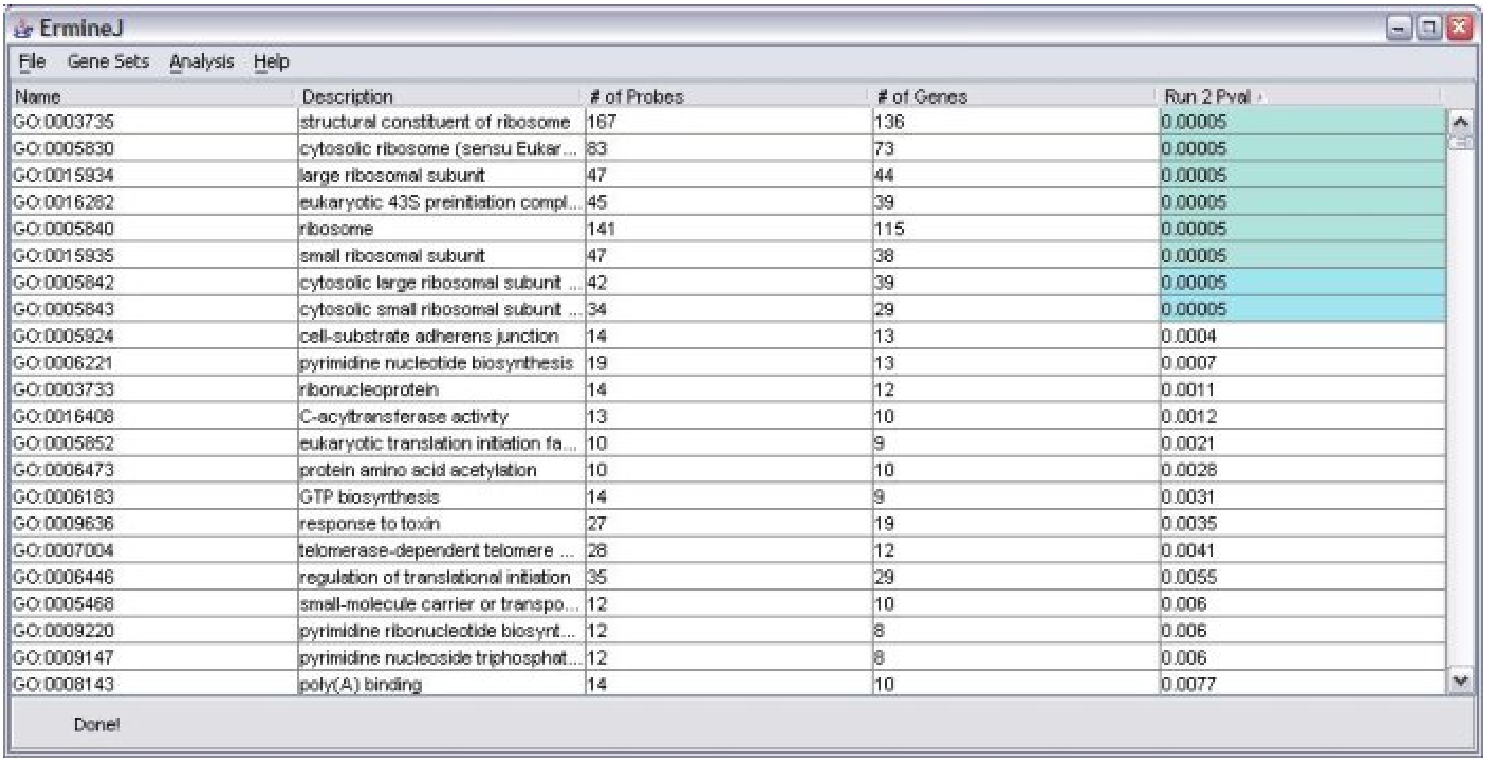
The values are not incorrect, but could be confusing. This is due to a decision we made in the design of the user interface.

The "Probes" and "Genes" columns show how many probe and genes are on the *array design* , not necessarily the number represented in your data.

That is, if you have pre-filtered your input gene scores or your data (for correlation analysis), not all the genes on the array design are in your data set. For example, if you use the Affymetrix HG-U95A array, which has about 12,500 probe (sets), but filtered out ones you deemed "not expressed", you might only have 7000. This is going to affect the sizes of the gene categories represented.

To display this, we would have to adjust the values in those columns to reflect the actual situation in your data. This would be fine if we only had one analysis to show. However, because we allow you to display multiple analyses, potentially from multiple data sets which have been filtered differently, there might not be a single "correct" value to show. Therefore we always just show the value for the full array design.

You can always get the "correct" number by holding your mouse over the p value value; a tool-tip will appear which shows the actual values used for that gene set, for that particular run. When you save a run to a file, the values for the number of probes and genes are the ones used for your particular data. (Other types of work-arounds could be envisioned, which we are considering for a future version of the software)



Name	Description	# of Probes	# of Genes	Run 2 Pval
GO:0003735	structural constituent of ribosome	167	136	0.00005
GO:0005830	cytosolic ribosome (sensu Eukar...	83	73	0.00005
GO:0015934	large ribosomal subunit	47	44	0.00005
GO:0016282	eukaryotic 43S preinitiation compl...	45	39	0.00005
GO:0005840	ribosome	141	115	0.00005
GO:0015935	small ribosomal subunit	47	38	0.00005
GO:0005842	cytosolic large ribosomal subunit ...	42	39	0.00005
GO:0005843	cytosolic small ribosomal subunit ...	34	29	0.00005
GO:0005924	cell-substrate adherens junction	14	13	0.0004
GO:0006221	pyrimidine nucleotide biosynthesis	19	13	0.0007
GO:0003733	ribonucleoprotein	14	12	0.0011
GO:0016408	C-acyltransferase activity	13	10	0.0012
GO:0005852	eukaryotic translation initiation fa...	10	9	0.0021
GO:0006473	protein amino acid acetylation	10	10	0.0028
GO:0006183	GTP biosynthesis	14	9	0.0031
GO:0009636	response to toxin	27	19	0.0035
GO:0007004	telomerase-dependent telomere ...	28	12	0.0041
GO:0006446	regulation of translational initiation	35	29	0.0055
GO:0005468	small-molecule carrier or transpo...	12	10	0.006
GO:0009220	pyrimidine ribonucleotide biosynt...	12	8	0.006
GO:0009147	pyrimidine nucleoside triphosphat...	12	8	0.006
GO:0008143	poly(A) binding	14	10	0.0077

Done!

ermineJ

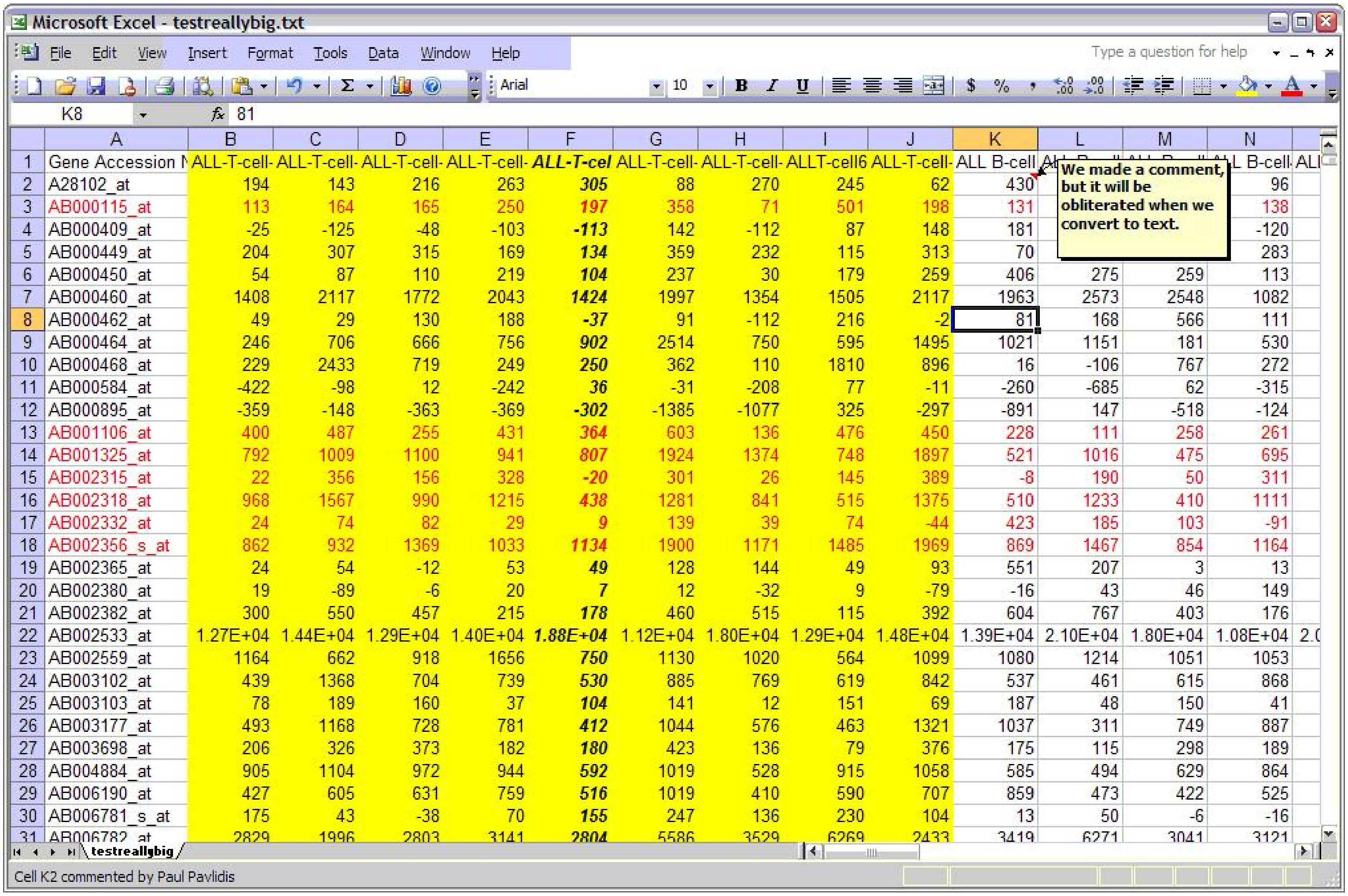
Saving Excel spreadsheet tables as tab-delimited text files

All of the software we provide uses plain text files as input; they cannot use Excel spreadsheets or any other format. This page describes in detail how to convert an excel spreadsheet into a suitable text file. These instructions were set up under Windows but for MacOS X it should be similar.

First make sure you know about the following important points.

- Your original Excel file will not be affected by this procedure, because you are saving a copy of the file.
- The worksheet you are viewing now is the one which will saved. Other worksheets are not saved.
- All row and column names must not contain spaces. This is because our parsers may get confused by names containing spaces.
- Missing values should just look blank; avoid putting spaces or other characters in any fields.
- The values in the file, other than the row and column labels, must be numbers (no letters or other characters)

Here is what a reasonable spreadsheet might look like in Excel before you save it as text. Note that all the pretty formatting will be lost when you save it as text. If you open the saved text file in excel again, the colors, formulas, etc. will be gone. So make sure you have already saved the file as an "xls" document.


Microsoft Excel - testreallybig.txt
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial
10
B I U
$ %
K8
testreallybig

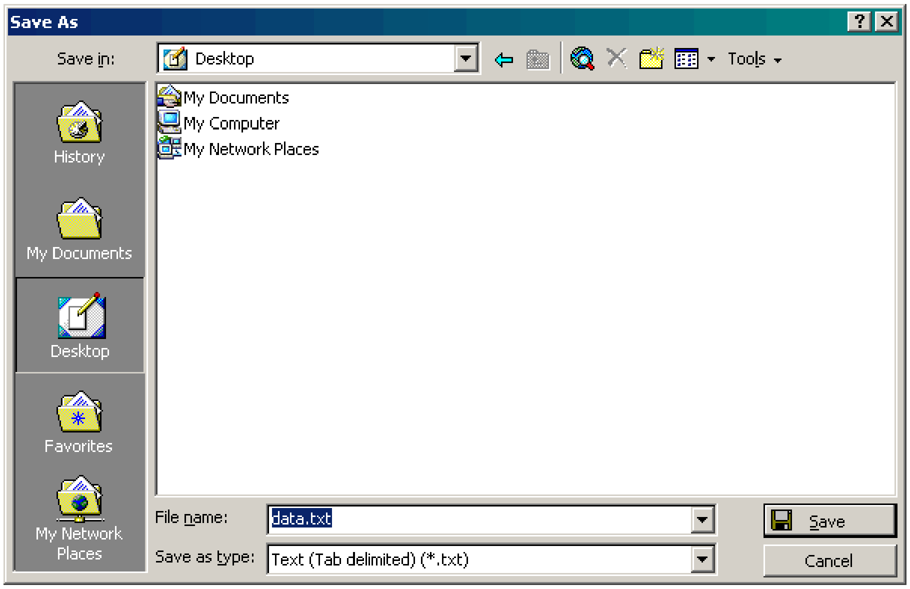
fx 81

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Gene Accession N	ALL-T-cell-	ALL-T-cell-	ALL-T-cell-	ALL-T-cell-	***ALL-T-cel***	ALL-T-cell-	ALL-T-cell-	ALLT-cell6	ALL-T-cell-	ALL B-cell	Al		L B-cell	ALI
2	A28102_at	194	143	216	263	***305***	88	270	245	62	430			96	
3	AB000115_at	113	164	165	250	***197***	358	71	501	198	131			138	
4	AB000409_at	-25	-125	-48	-103	***-113***	142	-112	87	148	181			-120	
5	AB000449_at	204	307	315	169	***134***	359	232	115	313	70			283	
6	AB000450_at	54	87	110	219	***104***	237	30	179	259	406	275	259	113	
7	AB000460_at	1408	2117	1772	2043	***1424***	1997	1354	1505	2117	1963	2573	2548	1082	
8	AB000462_at	49	29	130	188	***-37***	91	-112	216	-2	81	168	566	111	
9	AB000464_at	246	706	666	756	***902***	2514	750	595	1495	1021	1151	181	530	
10	AB000468_at	229	2433	719	249	***250***	362	110	1810	896	16	-106	767	272	
11	AB000584_at	-422	-98	12	-242	***36***	-31	-208	77	-11	-260	-685	62	-315	
12	AB000895_at	-359	-148	-363	-369	***-302***	-1385	-1077	325	-297	-891	147	-518	-124	
13	AB001106_at	400	487	255	431	***364***	603	136	476	450	228	111	258	261	
14	AB001325_at	792	1009	1100	941	***807***	1924	1374	748	1897	521	1016	475	695	
15	AB002315_at	22	356	156	328	***-20***	301	26	145	389	-8	190	50	311	
16	AB002318_at	968	1567	990	1215	***438***	1281	841	515	1375	510	1233	410	1111	
17	AB002332_at	24	74	82	29	***9***	139	39	74	-44	423	185	103	-91	
18	AB002356_s_at	862	932	1369	1033	***1134***	1900	1171	1485	1969	869	1467	854	1164	
19	AB002365_at	24	54	-12	53	***49***	128	144	49	93	551	207	3	13	
20	AB002380_at	19	-89	-6	20	***7***	12	-32	9	-79	-16	43	46	149	
21	AB002382_at	300	550	457	215	***178***	460	515	115	392	604	767	403	176	
22	AB002533_at	1.27E+04	1.44E+04	1.29E+04	1.40E+04	***1.88E+04***	1.12E+04	1.80E+04	1.29E+04	1.48E+04	1.39E+04	2.10E+04	1.80E+04	1.08E+04	2.(
23	AB002559_at	1164	662	918	1656	***750***	1130	1020	564	1099	1080	1214	1051	1053	
24	AB003102_at	439	1368	704	739	***530***	885	769	619	842	537	461	615	868	
25	AB003103_at	78	189	160	37	***104***	141	12	151	69	187	48	150	41	
26	AB003177_at	493	1168	728	781	***412***	1044	576	463	1321	1037	311	749	887	
27	AB003698_at	206	326	373	182	***180***	423	136	79	376	175	115	298	189	
28	AB004884_at	905	1104	972	944	***592***	1019	528	915	1058	585	494	629	864	
29	AB006190_at	427	605	631	759	***516***	1019	410	590	707	859	473	422	525	
30	AB006781_s_at	175	43	-38	70	***155***	247	136	230	104	13	50	-6	-16	
31	AB006782_at	2829	1996	2803	3141	***2804***	5586	3529	6269	2433	3419	6271	3041	3121	

We made a comment, but it will be obliterated when we convert to text.

Cell K2 commented by Paul Pavlidis

From the file menu, select "save as...". You should see the following dialog box. Select "text (tab delimited)" from the pull down menu. Fill in the file name you want to use. Remember that Unix file names cannot contain spaces or other cetain 'special' characters. (Details about unix are beyond the scope of this tutorial).

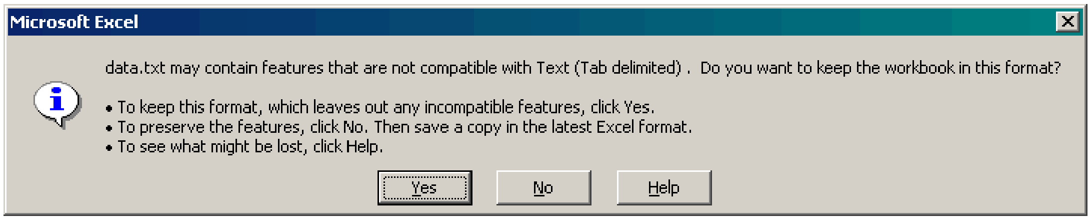


Save in: Desktop

File name: data.txt

Save as type: Text (Tab delimited) (*.txt)

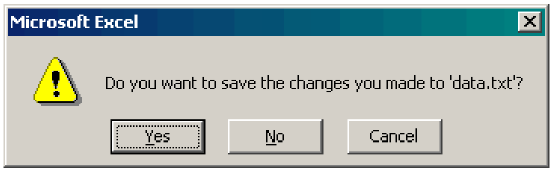
Then, for some silly reason, Excel shows you the following dialog box. Click "yes".



data.txt may contain features that are not compatible with Text (Tab delimited) . Do you want to keep the workbook in this format?

- To keep this format, which leaves out any incompatible features, click Yes.
- To preserve the features, click No. Then save a copy in the latest Excel format.
- To see what might be lost, click Help.

If you now close the document, Excel will usually give you the following prompt. Select "no" (otherwise you go back to the beginning again).



Do you want to save the changes you made to 'data.txt'?

That's it, you're done.

Using the ermineJ Application Programming Interface

ErmineJ consists of two jar files: ermineJ.jar and baseCode.jar. The latter contains code that is common to other projects, while ermineJ contains classes that are specific to the gene set scoring task. ErmineJ has dependencies on a number of other third party libraries including Commons Configuration. Use of the ermineJ API is covered by the Lesser Gnu Public License.

While the internal use of the ermineJ API is fairly complex, most of that complexity is not needed for use by third parties. The minimal requirements for an analysis are:

- A java.util.List of probe ids. This should contain all the probes on the microarray design (or at least, not just the ones that met some selection criterion)
- A List of gene symbols (in the same order as the probe ids)
- A List of Collections of GO terms for the gene symbols (keyed by the probe ids, in the same order as the probes)
- A List of scores for the probe ids. Typically these are p-values, but they can be any value you like, inluding a dummy variable indicating cluster membership etc.

The use of java.util.Lists was intended to make it very easy for third parties to create data structures that ermineJ can handle. It is the programmer's responsibility to make sure the Lists are in the correct order. While ermineJ will detect some types of problems with the input data structures, it cannot tell that you put the probe IDs in a different order than the gene symbols.

Once the above are assembled, the analysis proceeds in three phases:

1. Create a ClassScoreSimple object with the above lists as arguments to the constructor
2. Configure settings.
3. Run the analysis by calling the 'run' method.

The results can then be obtained with a simple method call..

The following code snippets demonstrate how to implement these steps.

```
List probes = null; // List of identifiers to be analyzed
```

```
        List genes = null; // List of genes corresponding to the probes.
Indicates the Many-to-one mapping of probes to genes.
        List goAssociations = null; // List of Collections of go terms for the
probes.
        List geneScores = null; // List of Doubles

        /* code to initialize data structures omitted */

         ClassScoreSimple css = new ClassScoreSimple( probes, genes,
goAssociations );

        // in our raw data, smaller values are better (like pvalues, unlike fold
        // change)
        css.setBigGeneScoreIsBetter( false );

        // set range of sizes of gene sets to consider.
        css.setMaxGeneSetSize( 100 );
        css.setMinGeneSetSize( 5 );

        // use this pvalue threshold for selecting genes. (before taking logs)
        css.setGeneScoreThreshold( 0.001 );

        // use over-representation analysis.
        css.setClassScoreMethod( Settings.ORA );
        /* ... etc. Reasonable defaults (?) are set for all parameters if you
don't set them. */

        css.run( geneScores ); // might want to run in a separate thread.

        // You should iterate over your tested gene sets.
        double fooPvalue = css.getGeneSetPvalue( "foo" );
        double barPvalue = css.getGeneSetPvalue( "bar" );
```

ermineJ

ErmineJ has a simple command line interface (CLI). In addition to providing a scriptable interface to the software, it provides access to some features that are not accessible through the graphical user interface (GUI). The CLI can also be used to start the GUI.

To access the CLI, you need to have installed the generic bundle. See the instructions.

Once you have set up the package, you should be able to access ermineJ by running ermineJ.bat or ermineJ.sh.

```
OPTIONS
        The following options are supported:

        -a file ...
                Sets the annotation file to be used.

        -F format (--format)           Set format (default is our own; 'affy'
is other valid value)   -c file ...
                Sets the class file to be used.

        -d dir ...
                Sets the data folder to be used.

        -e int ...
                Sets the column in the score file to be used for scores.

        -f die ...
                Sets the class folder to be used.

        -g int ...
                Sets the gene replicant treatment:  1 (best gene score used) or
2 (mean gene score used).

        -b Sets 'big is better' option for gene scores (default is false).

        -h or --help
                Shows help.

        -i int ...
                Sets the number of iterations.
        -l {0/1} ...
                Sets whether or not to take logs (default is true).

        -m int ...
                Sets the raw score method:  0 (mean),  1 (quantile), or  2 (mean
above quantile).

        -n int ...
                Sets the analysis method:  0 (ORA),  1 (resampling of gene
scores),  2 (profile correlation),  3 (ROC), 4 (T-test), 5 (Kolmogorov-Smirnov
test)[not all methods may be implemented]

        -o file ...
                Sets the output file.

        -q int ...
                Sets the quantile.

        -r file ...
                Sets the raw file to be used.

        -s file ...
                Sets the score file to be used.

        -t double ...
                Sets the pvalue threshold.

        -x maximum class size ...
                Sets the maximum class size.

        -y minimum class size ...
                Sets the minimum class size.

        -M method or --mtc method
                Sets the multiple test correction method: 0 = Bonferonni,  1 =
Westfall-Young (slow),  2 =  Benjamini-Hochberg (default)

        -C file ... or --config file ...
                Sets the configuration file to be used.

        -G or --gui
                Launch the GUI.

        -S file ... or --save file ...
                Save preferences in the specified file.
```

Frequently asked questions

How do I know if a gene set p value is 'significant'

One has to consider the problem of multiple testing, but because the gene sets (Gene Onotology group) overlap, simple methods will be too conservative. For this reason we don't recommend using a Bonferroni correction, though if a Bonferonni correction yields 'significant' results, you can certainly consider them reasonably trustworthy.

The software implements a false discovery rate (FDR) algorithm that leads to p values shown in a color at different levels of false discovery rate (see this page). However, this method still assumes the gene sets are independent and will lead to a convervative estimate of the FDR.

If you want to use Bonferroni, you can either use the command-line interface, or compute them yourself. To compute the Bonferroni-corrected p values, multiply your gene set p values by the number of gene sets analyzed (maximum of 1). Then you can use a threshold of 0.05 (which would not otherwise be reasonable).

My data set has no "significant" genes, but when I run ermineJ with resampling I get significant gene sets. How do I interpret these results?

Carefully. ErmineJ's resampling methods are based on analyzing your gene expression data in isolation, without reference to any theoretical background model (other methods may be added in future versions of ermineJ that allow this). It is capable of identifying the most interesting gene sets in your data, but this does not necessarily mean those gene sets are interesting in general. We recommend using such findings as an exploratory method for helping identify potential genes of interest, but because no gene is statistically significant on its own, the results for any given gene would have to be dealt with on a case-by-case basis using considerations other that statistical significance.

For a gene set to be considered statistically significant outside the context of your data set, a reasonable null hypothesis would yield a distribution of p-values in the gene set that is uniform. A uniform distribution of p-values is displayed in the details view of each class. When the blue line (your p-values) is far to the right of the expected distribution under the null (grey line), you can have higher confidence in the importance of that gene set. The reason we do not use this method exclusively is that in data sets where many genes have changed, every gene set will appear significant. Reasonable interpretation of the results requires consideration of both types of analyses.

Do I input the list of gene scores for the significantly changed genes, or all the genes?

All of them. The resampling method uses the gene scores for all the genes. The built-in overrepresentation analysis will use the threshold you set in the gene score threshold parameter to select genes. Note that we find that removing unexpressed genes can help the analysis, but it isn't required. Note that if your data have been filtered, the output panel does not list the actual number of probes or genes in your gene sets. See this page for a clarification.

How do I find out which genes are in a gene set without doing an analysis?

You can use the "View/Modify Gene set" tools, available from the main 'analysis' menu, or the context-specific pop-up menu on the gene set list. See this page

When I open the output file in Notepad it looks like a mess of letters and numbers.

The output will not look right when viewed in a text editor or a web browser. It will look correct in Excel if you open it as a tab delimited file.

My excel spreadsheet with my gene scores isn't being accepted as input.

You cannot input Excel spreadsheets; you have to save them as text first. See this page for details.

I don't see an annotation file for my microarray. What should I do?

Unfortunately we cannot generally create annotations for microarrays that are not commonly used. If we identify a commonly used platform, we will add it. This means that 'home made' spotted microarrays are not supported in most cases. Feel free to let us know if you have a microarray design that needs annotations for class scoring. In the meantime, you could try to use one of the "Generic" annotation files we provide, possibly with the help of your local friendly bioinformatician.

When doing an analysis, why did I see a warning "Attempt to take the log of a non-positive value"?

If your gene score file contains negative or zero values, and you check the "negative log-transform gene scores" box, you will see this error. If this concerns you, you should clean up your gene score file. If not, ermineJ sets these values to a small number (10^{-15}).

When doing an analysis, why did I see a warning ""Some probes in your gene score file don't match the ones in the annotation file" "?

This happens if your gene annotation file isn't the right one for the microarray you used (in which case your analysis will not work well), but can also happen if your file format isn't quite right. The analysis still proceeds when this happens.

When doing an analysis, why did I see a warning""Non-numeric gene scores(s) ('#NUM!') found for input file. These are set to an initial value of zero." ", or something similar?

This warning appears if the column you selected for your gene score file contains non-numeric data. This can happen if you have missing or invalid values (sometimes appearing as '#NUM!' in Microsoft Excel), but can also happen if you have chosen the wrong column in your data. Because such values are interpeted as zeros, you should be careful that your file contains the data you want. If you want to avoid problems you should remove those rows from your file entirely.

The resampling is taking too long, any suggestions?

This problem primarily affects the correlation score analysis in particular, which is very computationally intensive.

There are few ways to speed things up. One is to **uncheck** the "*Always use full resampling*" checkbox in the analysis setup wizard. This enables approximations which, while potentially yielding less accurate p-values, greatly increase processing speed. The results you get with this box unchecked will be reasonably similar to what you would get with full resampling.

Another tip is to not look at very large values for the "maximum class size" value. We usually use a value on the order of 200.

Finally, you can reduce the number of iterations. You might try temporarily setting this to 10000 while you experiment with the other settings, and then, once you find a setting you like, you can crank up the iterations for the "final" run (for example to 200000, which still should not take very long for the gene-score based resampling).

Why do my resampling results vary slightly from run to run?

This happens either when too few iterations are run, or sometimes when using the approximation methods. To ensure the highest accuracy, set the number of iterations to a larger value (say, 200,000 for gene score resampling) and **check** the *Always use full resampling*" checkbox in the analysis setup wizard.

What happens to probes that don't have any GO annotations?

They are ignored.

For resampling analysis of gene scores, how is the gene score threshold determined?

This is a trick question. There is no threshold used. Instead, all genes in a set contribute to the score for the gene set. For more details on how this works, see this page .

What happens if there are two probes for one gene?

See this page .

Why are the number of gene and probes shown in the output panel incorrect when my data set has been filtered?

See this page.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement

including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- **a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.
Copyright (C)
yyyy
name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA  02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C)
year
name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `` `show w' `` and `` `show c' `` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `` `show w' `` and `` `show c' ``; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.


signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```