Gene function analysis in complex data sets using ErmineJ

Jesse Gillis[1,2], Meeta Mistry[1–3] & Paul Pavlidis[1,2]

[1]Department of Psychiatry, University of British Columbia, Vancouver, British Columbia, Canada. [2]Centre for High-Throughput Biology, University of British Columbia, Vancouver, British Columbia, Canada. [3]CIHR/MSFHR Graduate Program in Bioinformatics, University of British Columbia, Vancouver, British Columbia, Canada. Correspondence should be addressed to P.P. (paul@chibi.ubc.ca).

Published online 3 June 2010; doi:10.1038/nprot.2010.78

ErmineJ is software for the analysis of functionally interesting patterns in large gene lists drawn from gene expression profiling data or other high-throughput genomics studies. It can be used by biologists with no bioinformatics background to conduct sophisticated analyses of gene sets with multiple methods. It allows users to assess whether microarray data or other gene lists are enriched for a particular pathway or gene class. This protocol provides steps on how to format data files, determine analysis type, create custom gene sets and perform specific analyses—including overrepresentation analysis, genes score resampling and correlation resampling. ErmineJ differs from other methods in providing a rapid, simple and customizable analysis, including high-level visualization through its graphical user interface and scripting tools through its command-line interface, as well as custom gene sets and a variety of statistical methods. The protocol should take approximately 1 h, including (one-time) installation and setup.

INTRODUCTION

Microarray gene expression experiments provide relative mRNA levels for thousands of genes simultaneously. For a given gene, expression level changes can represent the differences between experimental conditions, indicating a connection between the condition and the gene. For example, if the study was performed under two experimental conditions, changes in expression level would represent a differential expression. A subsequent point of interest would then be to determine whether genes that show higher levels of differential expression are enriched for a common function as characterized by, for example, terms from the Gene Ontology (GO)[1]. The emergence of high-throughput genomics methods has enabled the analysis of large gene lists a common bioinformatics activity, and many programs exist to perform such analyses[2–5] (with an extensive list present in Huang *et al.*[6]). The most common algorithms used generally rely on Fisher's exact test or the hypergeometric distribution to evaluate the probability that the degree of enrichment observed in a list of selected genes would be observed by chance.

The tool covered in this protocol, ErmineJ[7], is designed for ease of use but offers a high degree of control and customization in analysis. Many tools have been developed to perform gene set enrichment analysis (including as mentioned in refs. 8–12). The specific statistical implementation will often differ subtly between methods, but typically, the most important factor is the user interface and the tasks it easily allows the user to perform. ErmineJ offers a few analysis techniques that are less common, particularly permutation and rank-based statistics, but there are few features in any gene set enrichment tool that are not available in at least one other. ErmineJ is most useful for users who ultimately wish to assess their results in a detailed manner. Many users may only be interested in obtaining a best first pass of significant enrichment categories without being too concerned about the role of subtle choices (such as exact thresholds) in their results. Although ErmineJ can be used simply with default options, it is primarily geared toward users who may be interested in exploring the biology of the statistical results in more detail, without requiring any particular statistical background knowledge. In addition to providing a large number of statistical options, ErmineJ also offers visualization of expression data and scripting for high-throughput analysis. An example of software offering a contrasting approach is DAVID[6]. DAVID is a web-based tool that allows users to upload a gene list for overrepresentation analysis. This means that, although DAVID is simple to use, interaction is limited to downloading results. In contrast, ErmineJ[7] is designed to be installed on the user's computer, and analyses are run locally. Thus, it offers greater capacity to compare algorithms, visualize the data and customize the analysis. For example, ErmineJ allows users to generate custom gene sets. Finally, it offers three distinct methods of analysis, in addition to the most common hypergeometric distribution–based overrepresentation analysis (ORA): gene score resampling (GSR), rank-based gene scores in receiver-operator curves (ROC) or correlation resampling (CR). We describe these methods briefly in the following section.

Analysis methods

Gene score resampling differs from ORA primarily in that a threshold score does not need to be set; the method uses all the gene scores for the genes within a gene set to produce a score. Because of this, even genes that are below the threshold for selection will have an effect on the output scores. Unlike ORA, GSR uses the underlying gene scores rather than their ranks. This, too, can have an effect, as more information from the original data is preserved. Typically, ORA and GSR will produce almost the same result. One advantage of GSR over ORA is that because it does not require a threshold to be set, it can be more robust than ORA. Further, if no threshold can be determined or no genes meet selection criteria for the threshold (problematic for ORA), GSR can still generate significant results. GSR is described in more detail in reference 13.

The ROC method is an alternative to using the ORA method or resampling methods. The ROC is a standard, fast, nonparametric way of evaluating scores of items—in this case, gene scores—by their ranks. In essence, ROC measures the degree to which high-scoring genes belong to a given set. Performance of the ROC is usually characterized by the area under the ROC curve, which can

be intuitively understood as the probability that, given two genes with one of them belonging to a given class, the correct one can be identified. Thus, an ROC value of 1 denotes perfect classification performance, whereas 0.5 represents random performance. Both exceptionally high and low values are statistically significant, but ErmineJ assumes that prediction relates specifically to the fact that genes scoring unusually near the top of the list are of interest, and genes scoring unusually near the bottom of the list are not. Thus, *P*-values are for high values only and are computed as in Breslin *et al.*[14]. The ROC method is based only on the order of the scores, and the underlying values have no effect beyond determining this ranking.

Correlation resampling uses the underlying expression profiles and not the gene scores to determine significance (and thus differs from ORA and GSR). The score for each gene comes from the correlation of the expression profiles. In essence, the degree to which genes in a set cluster with one another is being computed, although they may not be in a single cluster. For example, a gene set consisting of genes in two very tight clusters may get a good score (although a single good cluster would perform better). This method should be used when gene clustering is of more interest than differential expression. Correlation scoring is also useful as a control for the other methods based on gene score. Correlated gene sets can disrupt score-based analysis in which the differential expression effect is not strong. To test this, one needs to perform CR as a reanalysis, having already performed a score-based method (such as ROC) on the data. If a gene set is common to both analysis types, it is important to check that the correlation is not linked to the differential expression.

Selecting analysis type (see Step 2)

Overrepresentation analysis is based on hypergeometric distribution (or an approximation of it) and thus is similar or equivalent to the methods used by most software packages for computing gene set enrichment. The drawback of ORA is that it requires the determination of a threshold that defines the distinction between 'hit' and 'non-hit' genes. Because changing the threshold can change the results, sometimes dramatically, ORA is most appropriate when the threshold for 'hits' is not arbitrarily chosen. If the genes being examined can naturally have a binary classification applied to them (for example, 'on chromosome 2' and 'not on chromosome 2'), then ORA is a good choice. Failing that, it is better to use the ROC methods or GSR.

Gene score resampling is appropriate for analyzing differential expression and assumes that values have been confidently estimated rather than assuming confidence in the ranking generated by the values. GSR accounts for the relative differences between gene scores; that is, a gene with a much better *P*-value than the next gene on the list will have more weight than if the next gene's *P*-value was just slightly better.

The ROC method more closely resembles a generalization of ORA, in which no threshold is set but only ranks are used. As is often true of nonparametric techniques, using ranks reduces statistical power but provides the benefit of making fewer assumptions. Specifically, the ROC is appropriate when one is confident of the ranks rather than of the *P*-values themselves.

Correlation resampling is quite different from any of the other three methods. It does not use gene scores or the concept of 'hit' genes, and thus is not well suited for situations in which the user's analysis yields some type of ranking. Instead, CR analyzes the gene expression profiles directly. The score for a given gene group is determined by how well correlated the expression profiles are and can be thought of as measuring the degree to which genes within a set cluster with one another.

Choosing custom gene sets (see Step 4)

Custom gene sets are based on previous knowledge the user has regarding genes of interest. Generally, this will involve comparing the current data to gene sets based on a review of appropriate literature. For example, a user of ErmineJ could be conducting a study that will produce a list of genes believed to be enriched for synaptic function genes. The enriched GO categories will reflect this information. However, there may be a specific previous study that produced a similar list to which it would be interesting to compare the current work. Use of the custom gene set allows this in a convenient way. Thus, custom gene sets are useful for testing concordance with specific studies conducted earlier. Included in this use would be the generation of a novel gene set based on isolated reports of these genes being separately involved in some specific function (or disease). Preexisting databases of disease genes could be used to generate such custom lists (e.g., refs. 15,16)

Choosing GO aspects (see Step 5)

The three major branches of GO are biological process, molecular function and cellular component. It is often desirable to exclude one or more of these GO aspects to improve biological meaning and reduce the penalty imposed by multiple test correction. The three GO aspects show some redundancy because a given biological process will frequently have an associated molecular function and cellular location. This can yield three similar gene groups in which one would capture the grouping of interest, and testing all three aspects can be redundant. Biological process is usually of greatest interest to retain, because it groups genes in a way that is most congruent with the biologist's concepts of 'gene function' and 'pathways'.

Choosing the gene set size and gene replicate (see Step 6)

There are several reasons to avoid using extreme gene set sizes. The more gene sets you examine, the worse the multiple-testing issues; in addition, very small or very large gene sets tend to not be as informative.

'Gene replicate treatment' refers to what is carried out when a gene occurs more than once in the data set. Multiple probes targeting the same gene are typically used in expression profiling studies. We use the term 'gene replicates' to refer to these probes because they (theoretically) provide replicate scores for the same gene. Often these 'replicates' are not truly equivalent, and will, for example, have quite different sensitivity and specificity (or may target alternate splice forms). Because they vary in this way, one probe may yield a poor signal and another may result in a robust signal in the same data set.

In assessing gene scores, these replicates cannot be treated independently; for example, a gene with five replicates cannot be treated in the same manner as five different genes. In ErmineJ, each gene is only counted once (or in correlation analysis, each pair is only counted once), so that even if the gene had five associated probes, only one value will be used.

ErmineJ has two methods for determining what score should be used in the case of replicates ('Use Best scoring replicate' and 'Use Mean of replicates'; see Step 5), one of which is conservative and one of which, although less conservative, is sensible if individual replicates are specifically suspected not to be confident estimates.

Using the mean score is the conservative choice, and in this case, all values for the gene will be weighed equally. This also applies in the case of correlation analysis, in which it will operate between pairs of genes. Specifically, if gene A has three replicates and gene B has two replicates, there will be six comparisons between those genes in total. This method is sensible if the replicates are true replicates in cases in which the same sequence has been used more than once on the array.

The 'Best' choice is less conservative and only uses the score that will provide the most significant result in the final output. For example, if a gene has scores of both 5 and 6 ($-\log P$-value), then the 6 would be the score that is used, in contrast with the 'Mean' method in which 5.5 would be the score used. Similarly, for correlation analysis, only the best correlation is stored; in the case of the example above, between genes A and B, only the best correlation of the 6 combinations would be used. This method is typically reasonable if the 'replicates' do, in fact, target different transcripts (or sets of transcripts).

Applications of ErmineJ

To date, a number of transcriptomic studies have used ErmineJ as a means of extracting a biologically interesting interpretation of the results. These studies differ in their implementation of ErmineJ, each carefully selecting the method that best complements the work undertaken. The popularity of ORA is a consequence of expression profiling studies, which often impose a threshold to generate a list of differentially expressed genes. Rather than assessing the biological relevance of each individual gene, an ORA can identify the specific functional pathways that may be altered because of the affected genes. A recent study comparing the function of transcription factor PPARα in mouse and human hepatocytes used the ORA method in ErmineJ to assess their resulting gene lists, identifying various target processes[17]. These findings were in high concordance with results obtained from a pathway analysis and GSEA (both hypergeometric distribution based) on the same gene lists. A number of other studies have been conducted in much the same manner[18–20].

For studies in which expression changes are subtle (such as in psychiatric diseases), it is not uncommon to find few or no genes to be below the selected threshold. In such cases, the order in which genes appear in the list is worth investigating to identify representative biological processes. This can be carried out in ErmineJ using the ROC method. Alternatively, using the GSR method in ErmineJ can also be quite useful[21,22]. Sequiera *et al.*[21] used expression profiling to examine gene expression changes associated with major depression in the brains of suicide victims. They applied the GSR method and used top-scoring gene sets to identify a final list of affected differentially expressed genes.

Limitations

The current most-requested feature for ErmineJ is a more fully automated way of generating custom gene sets, and in general, of enabling the use of gene organization schemes other than GO. Currently, each custom gene set must be loaded individually, so using large numbers of gene sets from databases other than GO is problematic. This limitation is planned to be removed in a future release. Another limitation is that, whereas branches of the GO can be chosen, other specific restrictions on the analysis (for example, only terminal or leaf GO groups) cannot be carried out except by choosing the GO size. A limitation of ErmineJ compared with some web-based tools is that the user is unable to store an incomplete analysis in a saved session and return to it from elsewhere at a later date. Rather, the results from an analysis in ErmineJ must be downloaded after running the analysis and before closing down the application.

We note that this protocol does not cover the use of ErmineJ in a command-line or scripting environment. Users requiring automated or high-throughput analysis of their gene lists (for example hundreds of gene rankings to be analyzed) will want to use the command-line interface of ErmineJ, whereas this protocol focuses on the GUI analysis of individual gene lists.

MATERIALS

EQUIPMENT

- **Installing ErmineJ:** ErmineJ requires a recent version of Java (1.5 or higher). Once Java is installed, one must obtain ErmineJ either by Web Start or by downloading the package and installing it as described below.
- **Annotation files (see Step 1):** The gene annotations used by ErmineJ have two parts, each in its own file. The first is the GO term file, provided by the GO Consortium. This generic file (in XML format) describes the names of the gene categories and their relationships to each other but does not list which genes are in which category. This necessary file is provided with the software, but you may want to get an updated version from the GO server (http://archive.geneontology.org/latest-termdb/). This file is included as **Supplementary File 1**. The second file describes the probes (and/or genes) in your experiment and which genes are in which category; we refer to this as the 'probe annotation file'. ErmineJ-compatible files for many microarray designs as well as more generic gene-based files can be downloaded from our website (http://www.chibi.ubc.ca/microannots/). The file used in this protocol is included as **Supplementary File 2**. Links to these resources are also given on the ErmineJ website. ErmineJ also accepts annotation files in formats provided by some popular microarray manufacturers.
- **Input files (see Step 3):** Data should be assembled into a gene score file (format as described below) and/or supplied as a microarray raw data file for correlation analysis or visualization. The input files used in this protocol are available as **Supplementary File 3** (gene score file) and **Supplementary File 4** (raw data file).

EQUIPMENT SETUP

Installing ErmineJ

Using Web Start: The simplest way to use ErmineJ is to use Java Web Start. This allows you to run ErmineJ on any computer that has Java installed, without running a separate installer. Once ErmineJ is downloaded through Web Start, you should be able to run it even when not connected to the Internet, as long as it is maintained in the Web Start cache. Note that the Web Start version does not come with a GO file, this must be downloaded separately. The Web Start link is http://www.chibi.ubc.ca/ermineJ/webstart/ermineJ.jnlp. To make it easier to run offline, you should save the small JNLP file that links to your hard disk. To run ErmineJ, you have to double-click the file and, when prompted, allow Java to execute it.

Microsoft Windows: Windows installer is available from http://www.chibi.ubc.ca/ermineJ/downloadInstall.html. Download the installer (named ErmineJ-XXXXX.exe), and double-click on the file. An installation wizard will walk you through the installation process to set up the software. Once installed, ErmineJ will appear as a desktop icon, which can be double-clicked to

start the program. In addition, it is helpful to know that the data directory is located in the installation directory (e.g., C:\Program Files\ErmineJ\ErmineJ.data). This directory is used to store information by ErmineJ that will be used in the course of an analysis. The directory will be created during the installation. In addition to loading custom gene sets through ErmineJ, they can simply be placed in the ErmineJ/ermineJ.data/genesets directory for later use.

Macintosh: We do not provide a separate installer for Macintosh OSX. We recommend using the Web Start version, or installing the 'generic bundle' described next.

Other platforms: We provide a 'generic bundle' that can be used to run ErmineJ without an installer. This works well for Unix-like systems (such as MacOSX or Linux) and can be used on any platform, but requires a little bit of extra configuration. To set it up, you should unpack the distribution. You will end up with a directory called 'ErmineJ-2.1' or similar. Set the environment variable ERMINEJ_HOME to this directory. Add $ERMINEJ_HOME/bin to your path if you want to make it easier to execute. You can now execute ErmineJ by running the script ErmineJ.sh (Linux or MacOSX) or ErmineJ.bat (Windows) in the bin directory. By default, executing the script will simply print usage instructions. To use the graphical user interface (GUI), you should use `ermineJ.sh -G`. By using the other options, you can cause ErmineJ to run an analysis noninteractively, which is useful for scripting and high-throughput applications.

Input file formats

Gene score file: The input file for running an analysis is the gene score file. A 'gene score' is any value that is applied to genes in a microarray experiment and that represents some measure of 'quality' or 'interest'. Typical examples are a *P*-value, *t*-test or fold change indicating differential expression. We refer to this generically as 'gene selection'. ErmineJ does not compute such scores, hence they must be computed in another fashion (for example, in R) and supplied to the software for analysis.

The gene score file is a tab-delimited text file, at minimum having only two columns (as in the example in **Table 1**, taken from the Affymetrix HG-U95A microarray design). For simple use, use the following rules for the gene score file:

- A one-line header is required (if the header is missing, then the first line of data is omitted). The content of the header is unimportant (a blank line will suffice).
- The first column lists the gene names or probe identifiers. These identifiers must match the identifiers in the probe annotation file.
- The second column lists the scores associated with the first column. Nonnumeric values such as 'NaN' or '#NUM!' are interpreted as being equivalent to zero. To avoid the inclusion of the data, these rows should be removed when the file is generated for use.
- The file cannot be an Excel spreadsheet. Use 'Save as. Text' in Excel.

See **Table 1** for an example of an input file.

Gene score file cautions: If gene scores are raw *P*-values (a common case), it is important to either take the negative of the log transform of your values, or use the '–log' option in ErmineJ. 'Fold change' values are also often modified to be the absolute value of the log of the fold change.

TABLE 1 | Gene score file.

Probes	Probe score
117_at	0.592537874
1007_s_at	0.0643101
… (etc)	… (etc)

Unlike some software packages, ErmineJ needs the entire set of gene scores, rather than just the scores for the 'selected genes': that is, given a microarray with 15,000 probes on it, you will need to provide 15,000 gene scores. If you input only the 'significantly changed genes', ErmineJ will yield invalid results. One caveat to this general principle is if you have filtered your data so that 'unexpressed' genes (for example) have been excluded. In that situation, only the gene scores for some probes might be available. This is fine for the analysis, but the results will be based only on the probes for which you were able to provide data. However, ideally, analysis is performed so as to include all the probes from which the original gene selection was carried out.

If you do not have gene scores, only a list of 'hits' and 'non-hits', you can still use ErmineJ for ORA. Make a gene score file that contains scores of 1 for the 'hits' and 0 for the 'non-hits', and provide a threshold of 0.5.

Microarray raw data file: ErmineJ also allows the use of the underlying gene expression profile for visualization and/or analysis. These raw data are necessary for correlation-based analysis, whereas for the other analyses, they are only necessary for visualization and can be omitted. The data file to be loaded into the software should be in the format of a simple tab-delimited text file. Each row in the file is assigned to the dependent variable value (for example, expression levels or ratios) for one set of observations. Each column is assigned to one sample or observation (such as a microarray).

For the microarray raw data, use the following rules:

- Input files must be tab delimited. Using space-delimited (etc.) files will generate an error. One easy way to produce files in the appropriate form is to 'export as text' from within Excel.
- Missing values are not a problem but should be indicated by blanks, not by 'NA' or any similar nonnumerical characters.
- The first column must contain probe names or gene names to match those used in your gene score file and in the annotation file.
- Only one column can be used by descriptors and all other columns must represent numerical data; that is, do not include additional columns in the file that are neither data nor the necessary label.

Each row will represent the expression values measured for a specific gene, with the columns representing different arrays (samples). Later analysis can be conducted in a more convenient manner if data columns are grouped according to the conditions under which they were run. For example, all 'wild type' columns could be placed together, and all 'mutant' columns could also be placed together following wild-type columns.

See **Table 2** for an example microarray raw data file.

PROCEDURE

Getting started

1| On initially starting the software, you are presented with a dialog box requesting a GO XML file and a probe annotation file (**Fig. 1**). Enter the GO XML file you are using and the probe annotation file (see EQUIPMENT for further details about these annotation files), using the 'Browse' buttons to locate them on your computer. Select the format of the annotation file from the pull-down menu. If the file from the ErmineJ web site is obtained, the format is 'ErmineJ'. Then click on 'Start'. The program will take 15–30 s to load the data files. Because these files are only loaded once on start-up, there will not be a similar wait for each analysis. At the top of the window, the menu bar offers various options enabling users to search and access the results of an analysis. The status bar, found at the bottom of the window, is used to show relevant information pertaining to the choice of action taken from the menu bar.

? TROUBLESHOOTING

TABLE 2 | Raw data file.

Gene	Mutant	Mutant	Mutant	Wild type	Wild type	Wild type
100001_at	36.3	77.8	64.4	89.4	126.6	86.2
100002_at	1,504.2	1512	944.5	1,157.9	1,652	1,358.9
100003_at	845.9	966.5	1,057.4	987.4	764.1	878.5
100004_at	2,304.4	1,991.1	2,783.7	1,929.8	2,236.8	2,664.1
100005_at	3,826.5	2,876.9	4,514.1	3,187.8	2,454.3	3,730.6
100006_at	3,635	2,584.6	3,554.9	2,810.9	1,629	2,248.6
100007_at	6,328.4	6,197.8	7,236.4	6,224.9	6,950	6,206.8
100009_r_at	6,580.6	8,715.9	5,280.3	6,569.4	8,513.4	7,236
100010_at	368.2	344.5	62.4	200	282.7	583.4
100011_at	1,949.7	2,511.3	1,937.8	2,684.1	1,722.5	2,101.3
100012_at	3,145.6	2,936.7	3,358.4	4,250.8	2,706.4	2,776
100013_at	1,098.4	720.8	1,418.8	886.9	764.4	1,247.6
100014_at	1,108	1,197	985.4	1,216.7	1,328.1	1,161.5
100015_at	6,005	1,040.6	4,434.1	1,069.4	864.8	2,617.4
100016_at	4,485.3	3,236.2	4,910.2	3,474.6	3,447.1	3,493
100017_at	497.5	399.3	964.2	347.7	524.5	561.3
100018_at	540	1,209.7	811.1	1,880.8	317.9	587.8
100019_at	303.5	46.4	0.9	53.4	−252.6	−346.9
… (etc)	…	…	…	…	…	…

Selecting analysis type

2| Next, you can perform an ORA, a GSR, an ROC analysis or a CR by selecting 'Run analysis' from the Analysis Menu (**Fig. 2**); see INTRODUCTION for further information on selecting analysis type.

▲ **CRITICAL STEP** ORA, GSR, ROC and CR follow the same method for Steps 2–6 in this protocol but they differ in Step 7.

File list request

3| Two data files are requested, a gene score file and a raw data file (**Fig. 3**); see EQUIPMENT and EQUIPMENT SETUP for further details on the format of these input files. For ORA or ROC, only the gene score file is necessary. Entering the raw data is necessary only to use some of the visualization tools. In this panel, you must also select which column contains the scores. The first column in the score file contains probe identifiers and the gene score numeric values are in any of the subsequent columns. If your gene score file has only two columns, just use the default value of 2.

? TROUBLESHOOTING

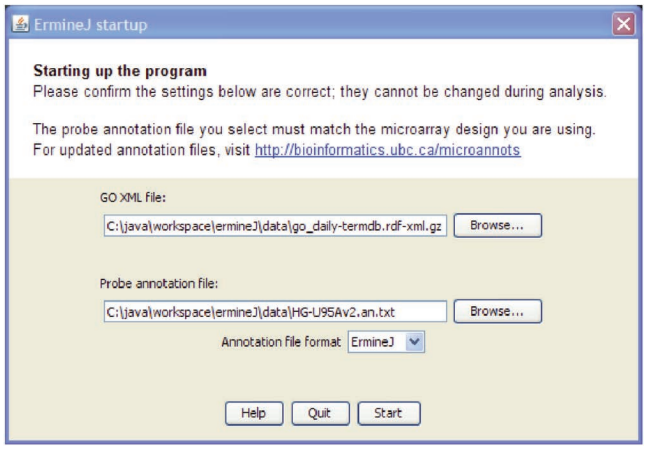

Figure 1 | Enter GO XML and Probe annotation file window. GO XML and probe annotation files need to be loaded on startup of ErmineJ to provide gene set information and microarray-specific gene information (Step 1). An example of the GO XML file can be found in **Supplementary File 1** and an example of the probe annotation file in **Supplementary File 2**.

Choose custom gene sets

4| This step asks you to add any custom gene sets you may have defined (**Fig. 4**); see INTRODUCTION for further information on choosing custom gene sets. In a default GO enrichment analysis, this is not necessary.

Choose GO aspects

5| Select the GO aspects to be included in the analysis from the choices provided (**Fig. 5**); see INTRODUCTION for further information on choosing GO aspects.

Choose gene set size and gene replicate treatment

6| The maximum and minimum gene set sizes determine the range of gene set sizes that will be considered (**Fig. 6**). We recommend avoiding the use of very small or very large gene sets and suggest a range of 5–200. There are two choices about gene replicates: using the 'Mean' score for the probes or the 'Best' score for the probes (see INTRODUCTION for further information regarding how to choose gene set size and gene replicate treatment).

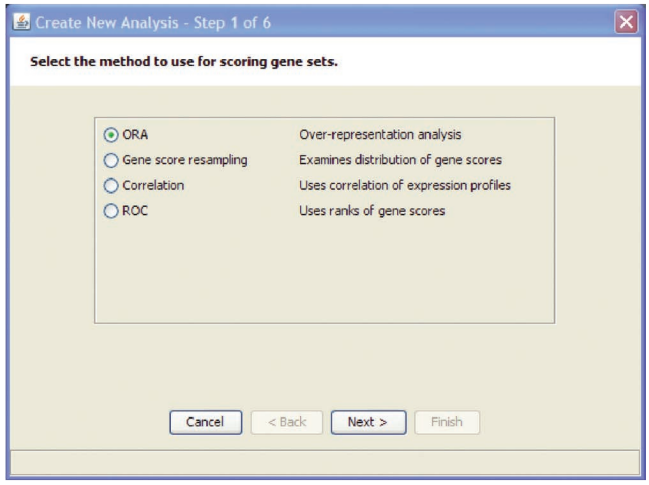

Figure 2 | Select analyses window. ErmineJ offers four analysis options for gene list significance (Step 2): overrepresentation analysis (ORA), gene score resampling (GSR), a receiver operator characteristic curve analysis (ROC) or correlation resampling (CR).

Further method options for specific analyses

7| At this stage, there are several options in ErmineJ, including 'Gene score threshold', log transformation, 'larger gene scores are better' and number of iterations (**Fig. 7**). Depending on whether you selected ORA, GSR, ROC or CR in Step 1, your choices at this point will differ. The log transformation and 'larger gene scores are better' options are common to all methods and are described in Step 8. The remaining two options should be used in a method-specific manner, as described in this step; to set gene thresholds or specify the number of iterations as appropriate, you should use option A for ORA, option B for GSR or option C for CR. Note that ROC uses only ranks (and not scores) and thus requires no thresholds to be set—in this case, proceed directly to Step 8.

(A) Setting gene score threshold for ORA

(i) The gene score threshold provides selection criteria to ErmineJ so that it can select which genes are 'good'. This is different from some other tools in which the 'good' genes are identified directly.

▲ **CRITICAL STEP** If you are using raw *P*-values as your gene scores, make sure your threshold is a value between 0 and 1 (for example, 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 to the log-transformed threshold. However, if your *P*-values are already log-transformed, you should use the exact opposite settings.

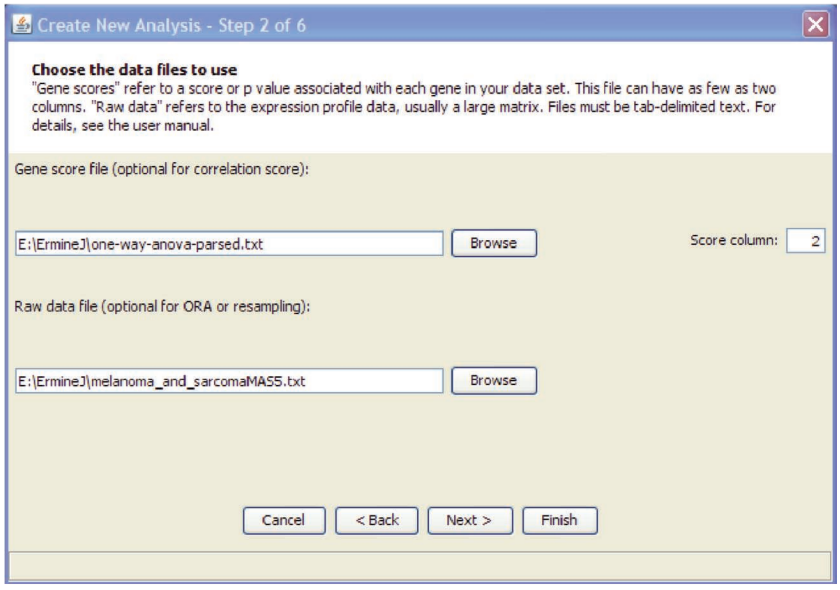

Figure 3 | Request for gene score file and raw data window in ErmineJ. See Step 3. Only correlation resampling requires raw microarray data. Gene score files take the form of a header line, then two columns (minimum), with the probe on the left and a score on the right (see example files; **Supplementary File 3** for gene score file and **Supplementary File 4** for raw data file).

(B) Determining the number of iterations to run for GSR

(i) In GSR, you must determine the number of iterations to be run. We suggest 10,000 iterations as a reasonable starting value.

▲ **CRITICAL STEP** Alternatively, it is possible to speed up the analysis by unchecking the 'Always use full resampling' checkbox, which will enable some approximations. When parameters have been finalized, it is typically desirable to choose a large number of iterations (200,000 or more) to obtain sufficient precision in the *P*-values and allow multiple test correction to work correctly.

(C) Determining the number of iterations to run for CR

(i) CR requires only a specified number of iterations, typically 10,000.

? TROUBLESHOOTING

8| The following options, log transformation (option A) and larger gene scores (option B), allow for correct interpretation of gene scores and are common to ORA, GSR and ROC methods. Note that this step can be skipped for CR.

(A) Log transformation

(i) If you choose this option, your input gene scores are transformed according to the function $f(x) = -\log_{10}(x)$. This option is convenient to modify gene score values that are *P*-values, a common occurrence. Conversion places *P*-values on a scale more useful for analysis, where larger values are better, but this does not mean that the 'larger scores are better' box (following) should be checked. That option refers to your original data, before transformation.

(B) Are larger gene scores better?

(i) If your gene scores are fold-change values, you will typically want to check the box indicating that 'larger scores are better'. That is only true if you have taken the absolute values of the fold-change values before loading them in ErmineJ. If you have taken the absolute value, both changes up and down will be considered. Alternatively, you could perform a one-sided fold-change test by retaining the sign of the values and then choosing 'larger scores or better' (or not) depending on whether you wish to test for positive or negative fold changes. This would allow you to see the effects of increases and decreases separately, perhaps depending on your specific hypothesis.

? TROUBLESHOOTING

● **TIMING ~1 h**

With all data prepared in advance, gene function analysis will take approximately 1 h, including the variation of analysis options to examine data in detail. The one-time-per-program initial loading of annotation files should take 15–30 s. Individual data analyses take a few seconds.

In gene set resampling (Step 7B(i): GSR options), a potential rate-limiting step is the sampling procedure. To speed up this process, we have implemented two optimizations, with a small sacrifice in accuracy. In the current implementation of the software, you can turn these both on or both off at the same time only. The control is 'Always use full resampling'. We suggest that for preliminary analyses you turn this option off, whereas you might want to turn it on for your final runs. In this case, we recommend setting the number of samples to be high (100,000 or more); the analysis may take a couple of minutes, depending on how many gene sets need to be analyzed and how big they are.

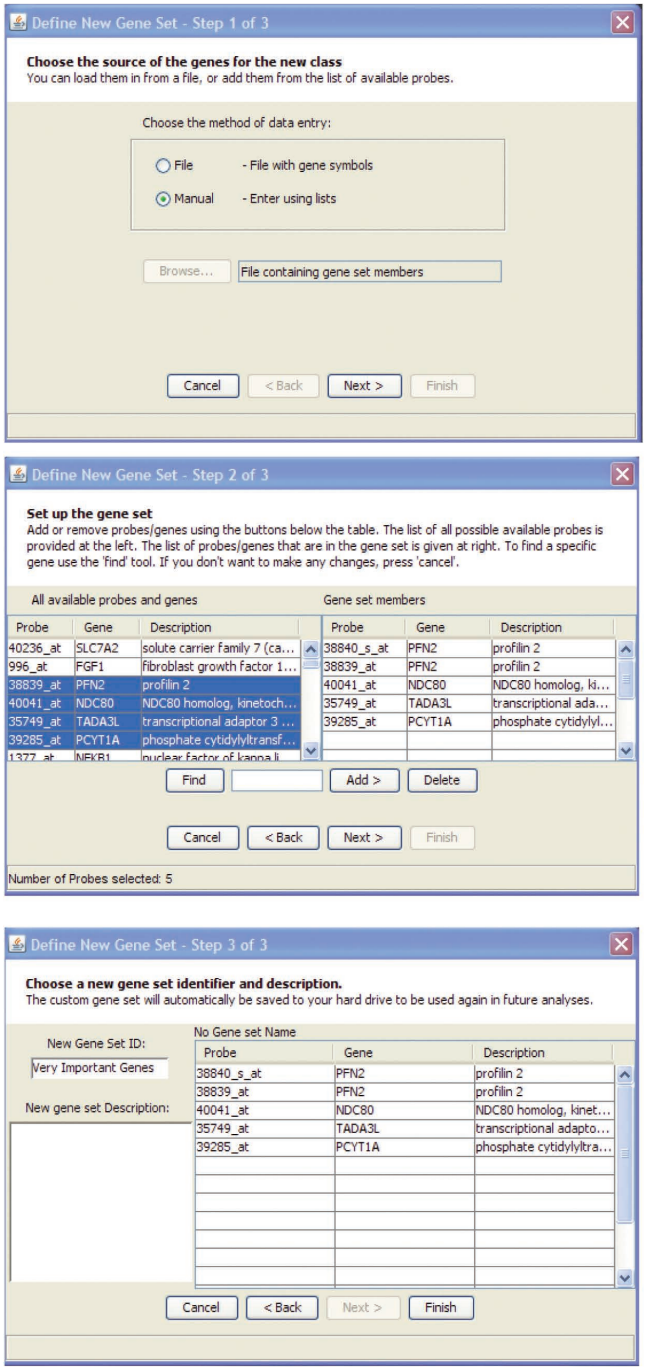

Figure 4 | Custom gene sets window in ErmineJ. Three steps allow the optional creation of custom gene sets (Step 4). Top: choosing whether to import the set from a file or input manually. Middle: manually adding or removing genes from a set. Bottom: giving the set a description.

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 3**.

TABLE 3 | Troubleshooting table.

Step	Problem	Possible reason	Solution
1	No annotation file for your microarray	Not a commonly used array	Contact the authors or use one of the 'generic' annotation files
3	Warning: 'Some probes in your gene score file don't match the ones in the annotation file'	Gene annotation file is not correct for the microarray or file format is incorrect	Analysis will proceed, but annotation file may need to be replaced
	Warning: 'Non-numeric gene scores(s) ('#NUM!') found for input file. These are set to an initial value of zero'	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	Check data files and replace or remove any non-numeric rows
7	Resampling is taking too long	Correlation score analysis is computationally intensive	Uncheck 'Always use full resampling' or reduce the number of iterations
8	Warning: 'Attempted to take the log of a non-positive value'	The gene score file contains zero or negative values and the 'negative log-transform gene scores' box was checked	Gene scores may need to be changed, as otherwise ErmineJ will set these values to a small number (10^{-15})

ANTICIPATED RESULTS

Following an analysis, a results window will appear. The three key parts of the window are the menu and status bars (as described earlier in Step 1) and the output panel.

The output panel comprises the bulk of the window, showing the results of the analysis. The user may choose to view the results in a table format or use a tree view, or switch between them using the labeled tabs in the output panel.

Table view

The table view displays four standard columns containing information on each gene set and thereafter can display any number of columns representing the results for each analysis that is run (**Fig. 8**).

Column headers are as follows:

- Name—the name or ID number of the gene set.
- Description—the description of the gene set.
- Probes—the number of probes in the gene set. Note that this value represents the number of probes on the array design, not necessarily the number of probes in your data set (in which some of the probes may have been filtered out).

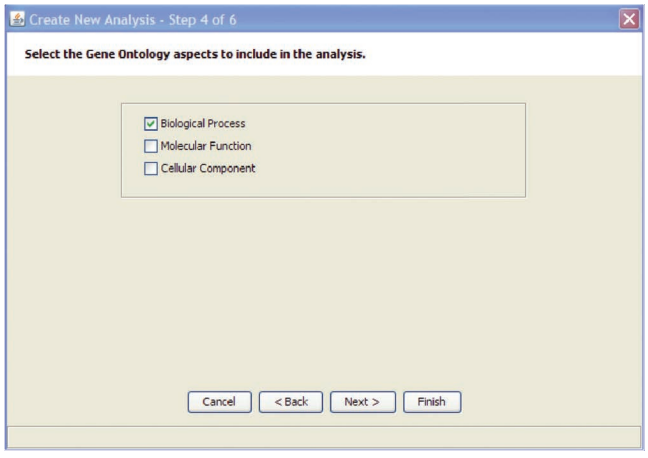

Figure 5 | Choose GO aspects window in ErmineJ. The choices are biological process, molecular function and cellular component (Step 5).

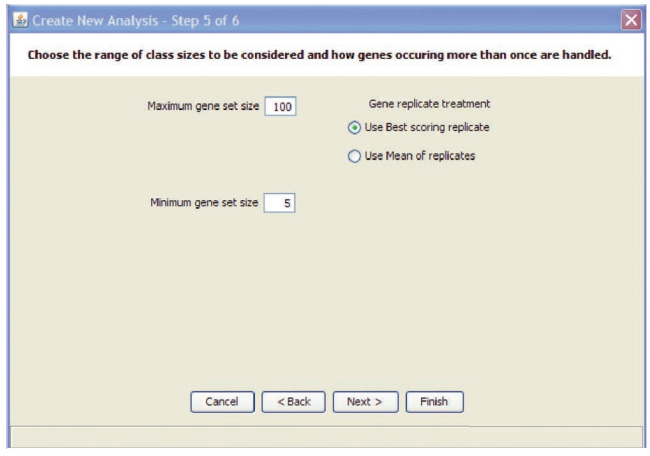

Figure 6 | Maximum and minimum gene set sizes window in ErmineJ. See Step 6. The maximum and minimum gene set sizes determine the range of gene set sizes that will be considered. Gene replicates can be chosen to be averaged ('Mean') or replaced by their most significant value ('Best').

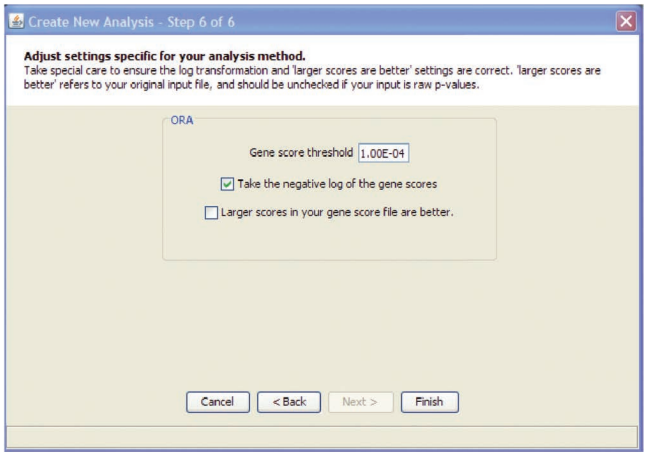

Figure 7 | Method options window for specific analyses. Options at this stage (Step 7) in ErmineJ include 'Gene score threshold', log transformation, 'Larger gene scores are better' and number of iterations. Depending on earlier choices of method, some of these options will not be necessary (for example, ROC uses rank so no threshold is necessary). Figure shows choices for the ORA method.

To determine the number of probes specific to your data set, you should hover the mouse pointer over the results column and read the sizes from the tooltip. This will also be the case for the 'Genes' column described below.

- Genes—number of genes in the gene set. Note that this value will always be less than or equal to the value in the 'Probes' column. This is because of the fact that a single gene can be represented by more than one probe on the array.
- Run Pval—the *P*-value representing the significance of the gene set for a particular run. This column will only appear once an analysis has been run. For each analysis that is run, a new column is appended to the right-hand side of the table in the output panel and is numbered sequentially. The user can rename the column by right-clicking on the column header or by selecting the 'Results' option in the menu bar.

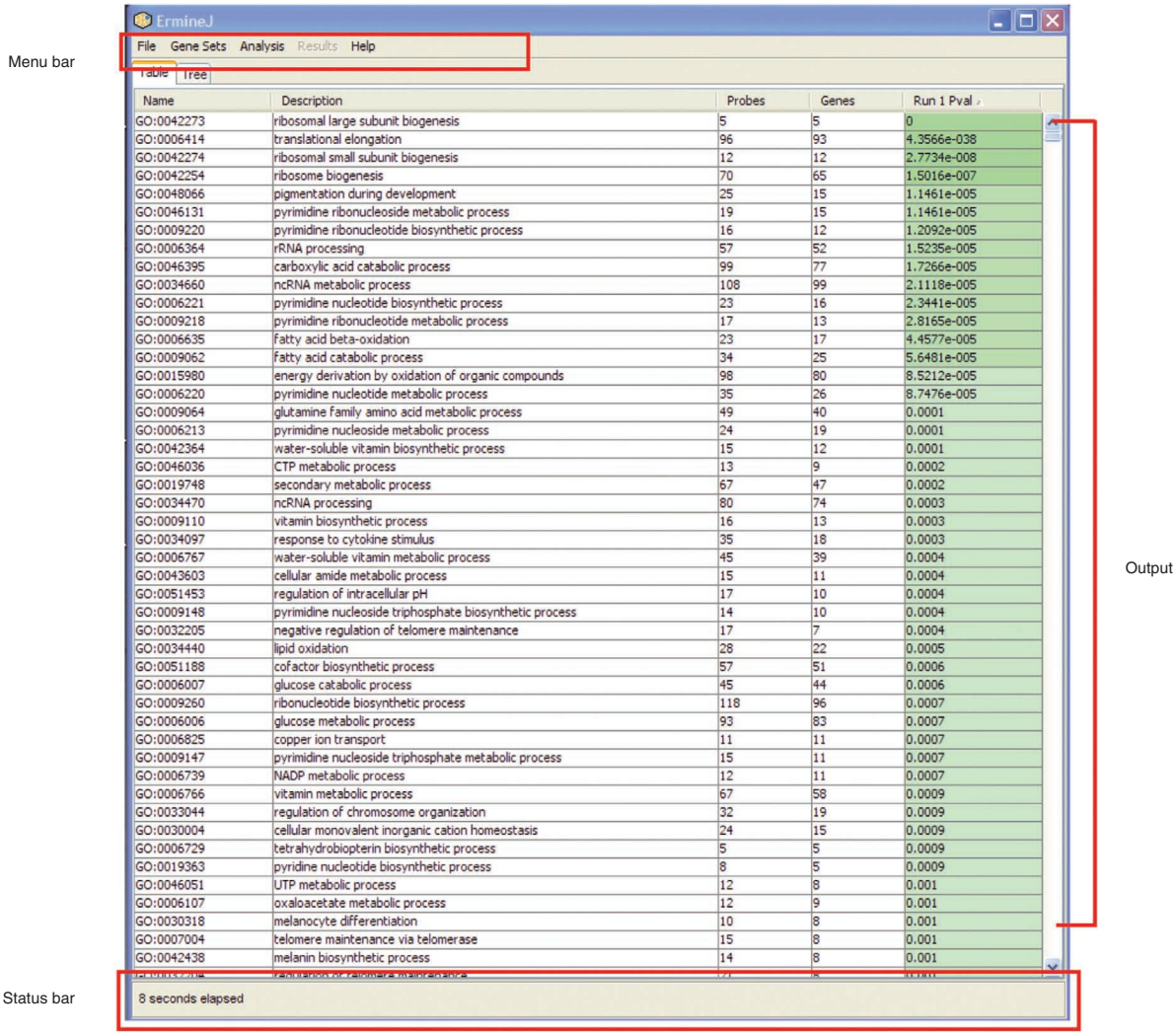

Figure 8 | Table view for results. One of the two standard views in ErmineJ, in this case in the postanalysis view. Significant gene sets by false discovery rate are shown in darker shades of color. The three key parts of the window have also been labeled: menu bar, status bar and output panel.

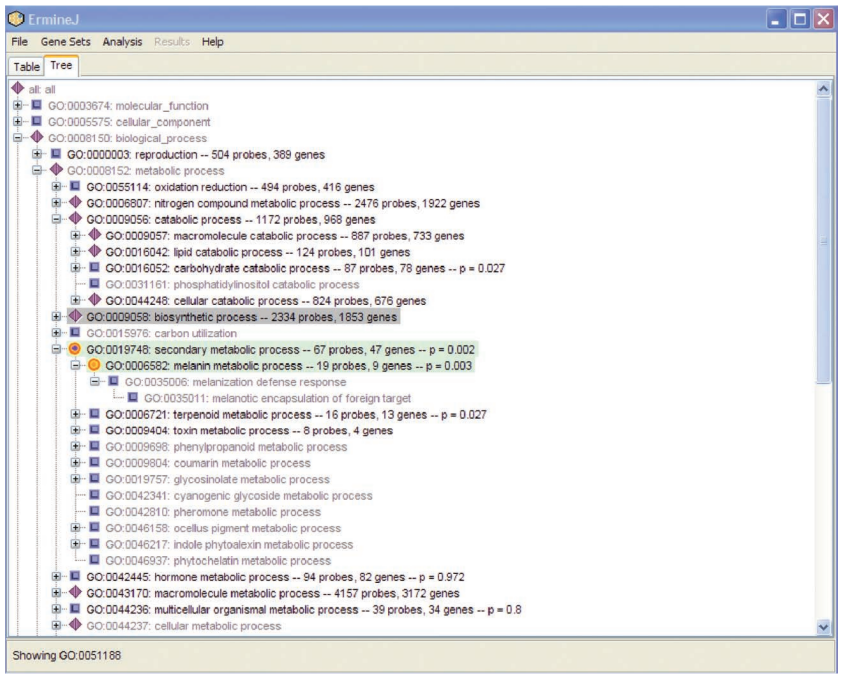

Figure 9 | Tree view for results. The second of the two standards views in ErmineJ shows GO terms in their hierarchical relationships. GO terms are shown with colored icons to indicate nonsignificant terms that have a statistically significant child (purple diamonds); significant terms that also have a statistically significant child term (yellow circle with purple middle) and significant terms (yellow circle).

You will find that the rows of the results column(s) are color coded by their *P*-value. ErmineJ uses the Benjamini-Hochberg[23] correction of *P*-values to determine which gene sets are selected with a particular false discovery rate (FDR). Gene sets that meet an FDR of 0.1 are considered 'good' *P*-values and are highlighted in various shades of green (colors are subject to change). Currently, colors indicate FDRs of 0.001, 0.01, 0.05 and 0.1, with brighter shades of green used to denote lower FDRs. Gene sets that do not meet the criterion are not colored.

If it is difficult to determine the exact color, let the mouse pointer hover over the result to read corrected *P*-values from the tooltip. Alternatively, you can select the 'Analysis' option from the menu bar to save the results to a tab-delimited text file, with the corrected *P*-values for each gene set saved in a separate column.

Note that 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, whereas at less-stringent thresholds it 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 gene set that has that FDR, and all gene sets listed above it would be selected.

If using the FDR is not suitable for analysis, there are other methods of multiple test correction (for example, Bonferroni) that can be accessed from the command-line interface. You can also save your results to a file and implement other methods of correction on your data.

Tree view

The 'tree' tab will switch you to a view of the gene set setup in a hierarchal structure (**Fig. 9**). In the tree panel, the entire GO is displayed (along with gene sets you have defined under 'user-defined'), which can make it difficult to find a specific gene set located anywhere in the hierarchy.

As this view is linked to the table view, the best way to locate a specific gene set would be to first select it in the table view, then right-click and select the option to 'Find set in the tree panel'. The color-coding scheme used in the table view also applies to the tree view; that is, gene sets highlighted in brighter shades of green indicate 'good' *P*-values. Gene sets that do not meet the criteria are listed but grayed out. In addition to color-coding of gene sets, we have also included various icons to facilitate a quicker identification of interesting results. The icons are as follows (note: the icons are subject to change with future versions of the software):

- Yellow spots indicate that the gene set has a 'good' *P*-value (which would also be highlighted in green).
- Purple diamonds indicate that the gene set has a child that has a 'good' *P*-value.
- Yellow spots with a purple center indicate that the gene set has a good *P*-value and it has a child that has a good *P*-value.
- Blue squares indicate a gene set that has no children with good *P*-values.

Exploring gene set details

From the menu bar you can select the 'Gene Sets' option to modify gene sets, create new gene sets and search for gene sets. Finding relevant gene sets in the results table is made easy with keyword searches, so that the user need not scroll through thousands of possible gene sets. You can identify all gene sets containing a particular phrase, or search a gene symbol to identify all gene sets containing that gene. Moreover, user-defined gene sets can easily be extracted using this menu option, or more readily by hitting 'Ctrl-U'.

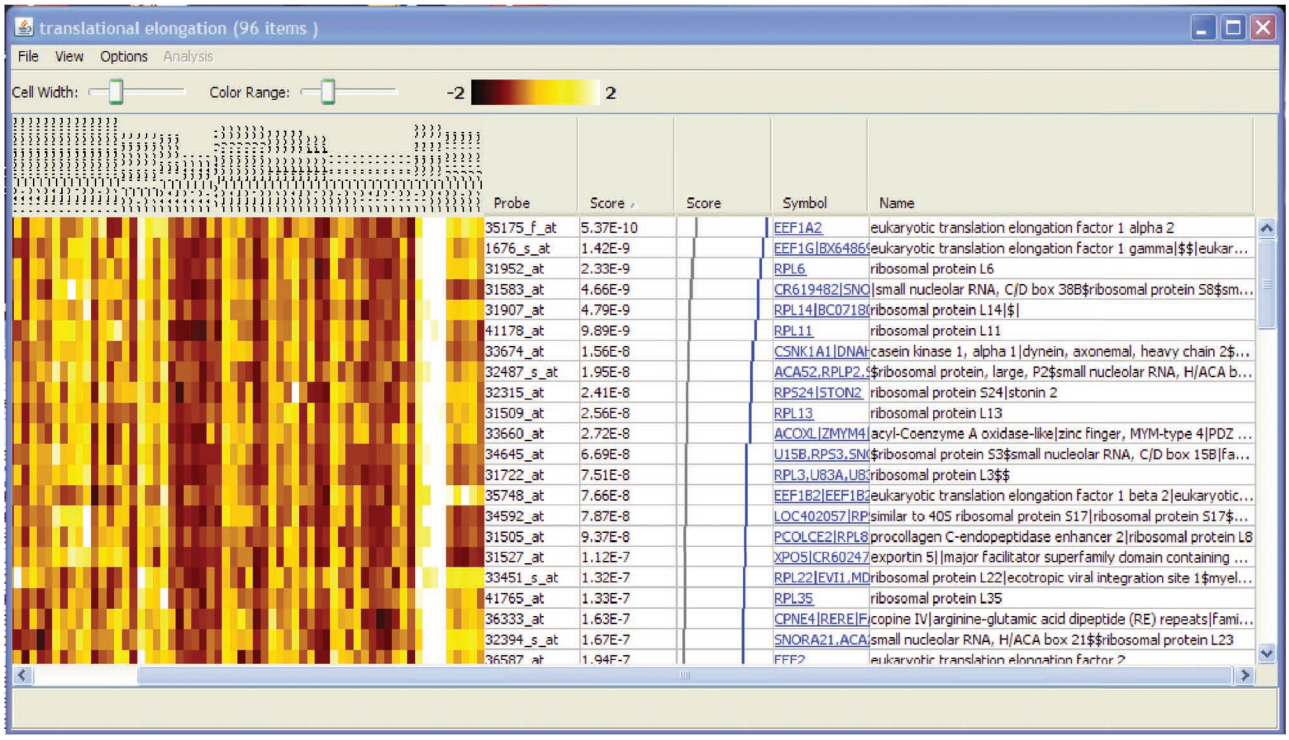

Figure 10 | Exploring gene details in ErmineJ. This output shows data exploration using significance values in concert with the microarray data (if it has been loaded). The actual pattern giving rise to significance values can be observed.

From the Output Panel (table or tree), you can visualize a specific gene set by double-clicking on the row (or the *P*-value if multiple analyses have been carried out). Even if no analyses have been run, a new pop-up window will be seen after double-clicking. In this window (**Fig. 10**), if you have previously loaded your raw expression data, you will be able to visualize expression patterns for genes in your data sets that are contained in your selected gene set. If you have not loaded data, you will be prompted to choose a data file. Ensure that the file format follows the rules described earlier.

You may choose not to load a data file at this time; you will not be asked again during this session, but you may set the dataset later using the 'Analysis' option from the menu bar. With this menu option, you may also switch the gene score file. You can open as many visualization windows as you like, including multiple windows for the same gene set.

Note: Supplementary information is available via the HTML version of this article.

ACKNOWLEDGMENTS This study was supported by NIH Grant GM076990, a Michael Smith Foundation for Health Research career award and by a CIHR New Investigator award. J.G. was supported by a MIND Foundation of BC postdoctoral award.

AUTHOR CONTRIBUTIONS P.P., J.G. and M.M. prepared the protocol and the article.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

Published online at http://www.natureprotocols.com/.
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/.

1. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
2. Zeeberg, B.R. *et al.* High-Throughput GoMiner, an 'industrial-strength' integrative Gene Ontology tool for interpretation of multiple-microarray experiments, with application to studies of Common Variable Immune Deficiency (CVID). *BMC Bioinform.* **6**, 168 (2005).
3. Martin, D. *et al.* GOToolBox: functional analysis of gene datasets based on Gene Ontology. *Genome Biol.* **5**, R101 (2004).
4. Al-Shahrour, F., Diaz-Uriarte, R. & Dopazo, J. FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics* **20**, 578–580 (2004).
5. Lee, J.S., Katari, G. & Sachidanandam, R. GObar: a Gene Ontology based analysis and visualization tool for gene sets. *BMC Bioinform.* **6**, 189 (2005).
6. Huang da, W., Sherman, B.T. & Lempicki, R.A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.* **4**, 44–57 (2009).
7. Lee, H.K., Braynen, W., Keshav, K. & Pavlidis, P. ErmineJ: tool for functional analysis of gene expression data sets. *BMC Bioinform.* **6**, 269 (2005).
8. Nam, D. *et al.* ADGO: analysis of differentially expressed gene sets using composite GO annotation. *Bioinformatics* **22**, 2249–2253 (2006).
9. Subramanian, A., Kuehn, H., Gould, J., Tamayo, P. & Mesirov, J.P. GSEA-P: a desktop application for Gene Set Enrichment Analysis. *Bioinformatics* **23**, 3251–3253 (2007).
10. Wrobel, G., Chalmel, F. & Primig, M. goCluster integrates statistical analysis and functional interpretation of microarray expression data. *Bioinformatics* **21**, 3575–3577 (2005).
11. Zhang, B., Schmoyer, D., Kirov, S. & Snoddy, J. GOTree Machine (GOTM): a web-based platform for interpreting sets of interesting genes using Gene Ontology hierarchies. *BMC Bioinform.* **5**, 16 (2004).
12. Kim, S.B. *et al.* GAzer: gene set analyzer. *Bioinformatics* **23**, 1697–1699 (2007).

13. Pavlidis, P., Furey, T.S., Liberto, M., Haussler, D. & Grundy, W.N. Promoter region-based classification of genes. *Pac. Symp. Biocomput.* **6**, 151–163 (2001).
14. Breslin, T., Eden, P. & Krogh, M. Comparing functional annotation analyses with Catmap. *BMC Bioinform.* **5**, 193 (2004).
15. Basu, S.N., Kollu, R. & Banerjee-Basu, S. AutDB: a gene reference resource for autism research. *Nucleic Acids Res.* **37**, D832–836 (2009).
16. Hamosh, A. *et al.* Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res.* **30**, 52–55 (2002).
17. Rakhshandehroo, M., Hooiveld, G., Müller, M. & Kersten, S. Comparative analysis of gene regulation by the transcription factor PPARα between mouse and human. *PLoS ONE* **4**, e6796 (2009).
18. Gamper, M. *et al.* Gene expression profile of bladder tissue of patients with ulcerative interstitial cystitis. *BMC Genomics* **10**, 199 (2009).
19. Shao, L. & Vawter, M.P. Shared gene expression alterations in schizophrenia and bipolar disorder. *Biol. Psychiatry* **64**, 89–97 (2008).
20. Lai, W.S. *et al.* Akt1 deficiency affects neuronal morphology and predisposes to abnormalities in prefrontal cortex functioning. *Proc. Natl Acad. Sci. USA* **103**, 16906–16911 (2006).
21. Sequeira, A. *et al.* Global brain gene expression analysis links glutamatergic and GABAergic alterations to suicide and major depression. *PLoS ONE* **4**, e6585 (2009).
22. Fulp, C.T. *et al.* Identification of Arx transcriptional targets in the developing basal forebrain. *Hum. Mol. Genet.* **17**, 3740–3760 (2008).
23. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B* **57**, 12 (1995).