GETS — Gene Expression to Spreadsheet

Documentation (web-based version)

Quick start
Gene Expression To Spreadsheet (GETS) is a tool that generates data-rich multi-coloured Microsoft Excel files from gene expression data files in plain text format. The web-based version is intended to be a front-end to run the Perl script. For a more detailed information on these files, please refer to the documentation of the Perl script.

Files
The web-based implementation of GETS accepts four distinct input files (all of them tab-delimited plain text files). Examples of inputs files can be downloaded from the **Downloads** section.

(1) The matrix file (mandatory), containing the intensity matrix which will be displayed as a heatmap. This is the equivalent to the --matrix argument in the command-line script. The first row must contain the names of the samples and the first column must contain the names of the genes/probes. To speed up the uploading process, this file must be gzip/zip compressed.

(2) The gene information file, containing gene/probe information. This is the equivalent to the --gene-info argument in the command-line script. The rows are the genes/probes and the columns contain information on each gene/probe (e.g. gene names, chromosomal locations, fold change values, *p*-values, etc.). The contents of this file will be displayed to the right side of the heatmap. To speed up the uploading process, this file must be gzip/zip compressed.

(3) The sample information file, containing sample information. This is the equivalent to the --sample-info argument in the command-line script. The contents of this file will be displayed above the heatmap. Two different formats are accepted: each column is a sample (when *Which format is the sample information file?* is set to *columns*), and each column is a row (when *Which format is the sample information file?* is set to *rows*).

(4) The colors file, containing color information for genes and samples. This is the equivalent to the --colors argument in the command-line script. The format if this file is explained in the documentation of the Perl script.

Options
(1) *Which format is the sample information file?* is the equivalent to the --sampleinfo-format argument in the command-line script. If *columns*, columns in the sample information file are the samples and the rows contain information on each sample. If *rows*, rows in the sample information file are the samples and the columns contain information on each sample.

(2) *Is gene information data already ordered?* is the equivalent to the --geneinfo-ordered argument in the command-line script. If *TRUE*, GETS assumes that the rows in the gene information file exactly match the rows in the matrix file. If *FALSE*, GETS will try to match the gene/probe names in both files. Please refer to the documentation of the Perl script.

(3) *Is sample information data already ordered?* is the equivalent to the --sampleinfo-ordered argument in the command-line script. If *TRUE,* GETS assumes that the sample order in the sample information file exactly matches the sample order in the matrix file. If *FALSE,* GETS will try to match the sample names in both files. Please refer to the documentation of the Perl script.

(4) *Palette for sample information colors* is the equivalent to the --palette argument in the command-line script.

(5) *Method for heatmap intensities* is the equivalent to the --intensity argument in the command-line script. The option sets how the heatmap color gradient is calculated.

(6) *Heatmap colors* is the equivalent to the --heatmap-colors argument in the command-line script. GBR: Green-Black-Red, BWR: Blue-White-Red.

(7) *Should expression values be centered?* is the equivalent to the --center argument in the command-line script. If *FALSE,* GETS assumes that the data in the matrix is centered around 0. If *TRUE,* GETS will center the values. Please refer to the documentation of the Perl script.

(8) *Run mode* is the only option not present in the Perl script. *Standard* is a non-verbose mode that will generate the output file without progress messages (unless an error occurred). *Verbose* will display progress and error messages.