GETS - Gene Expression to Spreadsheet

Documentation

1 Overview

Gene Expression to Spreadsheet (GETS) is a visualization tool aimed at easily combine multiple types of transcriptomics data into a single multi-colored file that can be readily opened by MS Excel, minimizing manual tinkering by the user. This documentation refers mainly to the command-line version of GETS. Section 15 explains how to use GETS online.

2 Pre-requisites

The following Perl modules must be installed before running the program:

```
Excel::Writer::XLSX
Getopt::Long
List::MoreUtils
List::Util
Math::Gradient
Scalar::Util
Statistics::Descriptive
```

They can be install with:

```
$ sudo cpanm name-of-the-module
```

Required and optional arguments

The usage format of the program is:

```
$ perl gets.pl --matrix=matrix-file [--geneinfo=geneinfo-file] [--
sampleinfo=sampleinfo-file] [--output=output-prefix] [--geneinfo-
colors=geneinfo-colors-file] [--geneinfo-ordered={TRUE|FALSE}] [--sampleinfo-
ordered={TRUE|FALSE}] [--sampleinfo-format={columns|rows}] [--center={TRUE|
FALSE}] [--intensity={IQR|IDR|<number>}] [--heatmap-colors={GBR|BWR}] [--
palette={bright|muted}] [--overwrite={TRUE|FALSE}] [--verbose={TRUE|FALSE}] [--
help] [--version]
```

2 Files

All user-supplied files must be tab-delimited plain text files. All lines beginning with *!* or *#*, as well as empty lines, will be ignored.

3 Basic usage (--matrix)

The only required argument is `--matrix`, which passes the name of the file containing the intensity matrix. The intensity matrix is a tab-separated text file of *n* rows x *m* columns, where the columns represent samples and the rows represent genes (or probes). The first row must contain the names of the samples and the first column must contain the names of the genes/probes. The folder *./example1/* contains a toy matrix derived (subsetted from experiment GSE100922 at the GEO repository) with 100 probes and 40 samples:

```
$ perl gets.pl --matrix=./example1/matrix.tsv                    (1)
```

This will generate an output file named *./example1/matrix.tsv.output.xlsx* (by default), where intensities are colored in a Green-Black-Red (GBR) gradient (by default). See Figure 1.

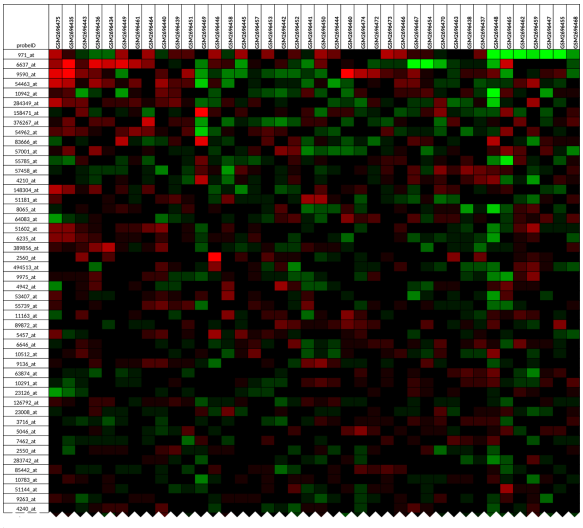

Figure 1. Output of example 1.

5 Output prefix (--output)

If we wish to assign the output a different name, we use the `--output` argument:

```
$ perl gets.pl --matrix=./example1/matrix.tsv --output=matrix                    (2)
```

This will generate an output file named *matrix.xlsx* in the working directory.

```
$ perl gets.pl --matrix=./example1/matrix.tsv --output=./example1/matrix         (3)
```

This will generate an output file named *matrix.xlsx* in the *./example1/* folder:

6 Heatmap colors

Heatmap color scheme (--heatmap-colors)

If we wish use the Blue-White-Red (BWR) color gradient for the heatmap, we use the `--heatmap-colors` argument:

```
$ perl gets.pl --matrix=./example1/matrix.tsv
--output=./example1/matrix.BWR --heatmap-colors=BWR                              (4)
```

See Figure 2.

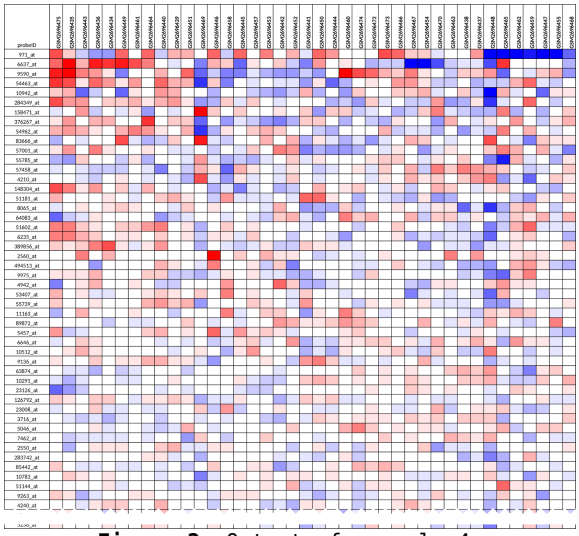

Figure 2. Output of example 4.

Both the GBR and BWR color schemes require the values in the matrix to be centered around 0. A warning message will be prompted if the input matrix does not comply with this requirement (see **Centering heatmap data** and **Troubleshooting** below).

Heatmap color intensity (--intensity)
The `--intensity` argument sets the intensity cutoff for the color gradient. Possible values are the following:

IQR	Sets the green/red limits to [*Q1-1.5IQR, Q3+1.5IQR*], being *IQR* the Inter-Quartile Range. The gradients of green/red will not be necessarily symmetrical around 0 (although they usually are in most centered expression matrices). This is the default.
IDR	Works like IDR but using the more restrictive IDR (Inter-Decile Range) instead of the IQR. Recommended if the heatmap is too dark or too bright after using `--intensity=`*IQR*.
<number>	Allow the user to define the limits of green and red [-*<number>*, *<number>*] (any intensity values lower/larger than those will be assigned the color intensity of [-*<number>*, *<number>*]).

The example (1) used the default limit of intensity (*IQR*). The following codes other values of `--intensity` to colour the heatmap:

```
$ perl gets.pl --matrix=./example1/matrix.tsv --intensity=IDR
--output=./example1/matrix.intensityIDR                                    (5)
```

```
$ perl gets.pl --matrix=./example1/matrix.tsv --intensity=2
--output=./example1/matrix.intensity2                                      (6)
```

```
$ perl gets.pl --matrix=./example1/matrix.tsv --intensity=4
--output=./example1/matrix.intensity4                                      (7)
```

The resulting heatmaps are shown in Figure 3.

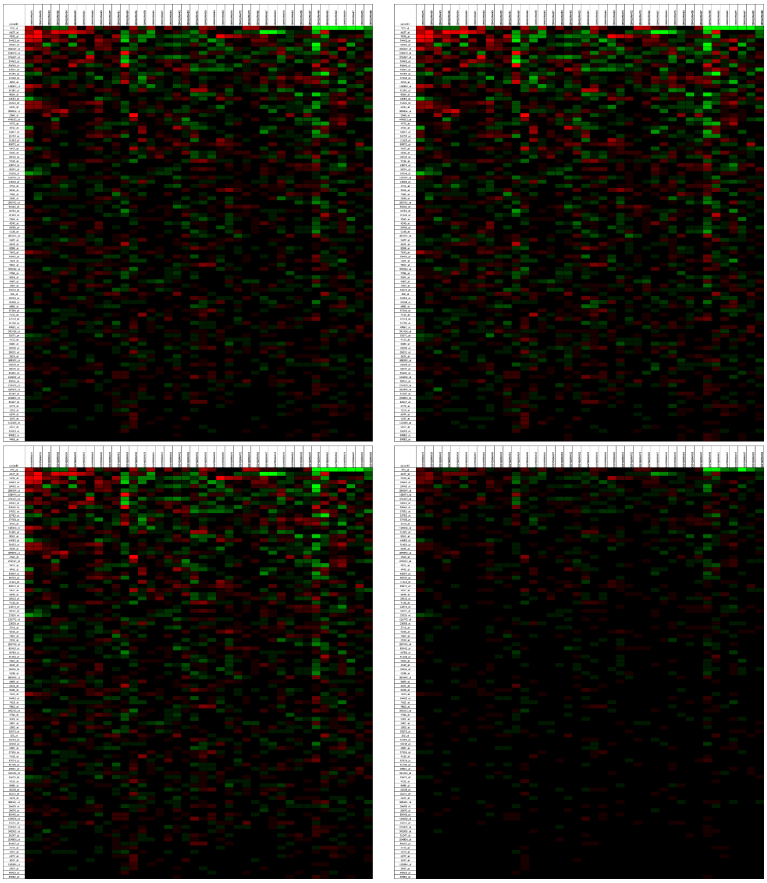

Figure 3. Outputs of (a) example 1; (b) example 5; (c) example 6; (d) example 7.

Because the values within the matrix file are largely within the [-2,2] range, there is not much difference in the heatmaps (except for example (7), where the intensity cutoff is [-4, 4]). An example of a matrix with unusually large values is provided in: *./example1/matrix_large_values.tsv*. In this case, it is advisable to use *--intensity=IDR* or even *--intensity=IQR* to avoid a matrix that is too bright. The file *./example1/matrix_large_values.tsv* was generated to illustrate this problem. Compare:

```
$ perl gets.pl --matrix=./example1/matrix_large_values.tsv  --intensity=IQR --output=./example1/matrix_large_values.intensityIQR
```
(8)

```
$ perl gets.pl --matrix=./example1/matrix_large_values.tsv --intensity=IDR --output=./example1/matrix_large_values.intensityIDR
```
(9)

```
$ perl gets.pl --matrix=./example1/matrix_large_values.tsv --intensity=2 --output=./example1/matrix_large_values.intensity2
```
(10)

```
$ perl gets.pl --matrix=./example1/matrix_large_values.tsv --intensity=4 --output=./example1/matrix_large_values.intensity4
```
(11)

The resulting heatmaps are shown in Figure 4.

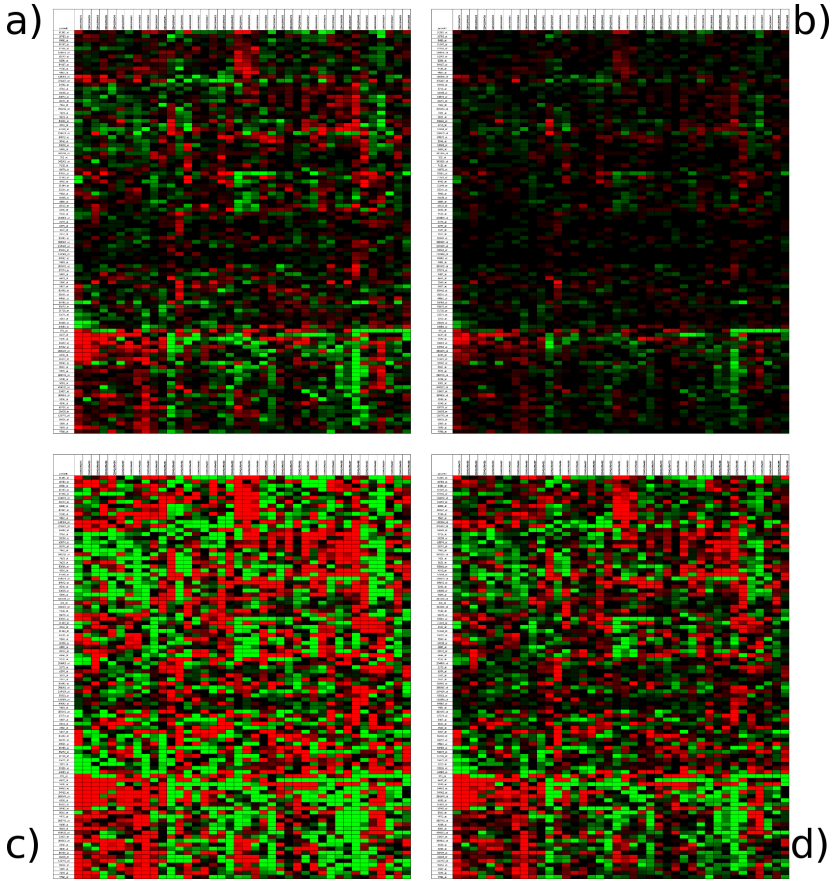

Figure 4. Outputs of (a) example 8; (b) example 9; (c) example 10; (d) example 11.

Intensities defined by IQR and IDR adjust automatically to the data range, while user-defined intensities produce very bright heatmaps because many expression values are outside the user-defined ranges.

The file *./example1/matrix_small_values.tsv* provides an example where the values in the input matrix are << |2|. The matrix will look very dark with the default `--intensity=`*`fixed`*. Compare:

```
$ perl gets.pl --matrix=./example1/matrix_small_values.tsv --intensity=IQR
--output=./example1/matrix_small_values.intensityIQR
```
(12)

```
$ perl gets.pl --matrix=./example1/matrix_small_values.tsv  --
intensity=IDR --output=./example1/matrix_small_values.intensityIDR
```
(13)

```
$ perl gets.pl --matrix=./example1/matrix_small_values.tsv --intensity=2
--output=./example1/matrix_small_values.intensity2
```
(14)

```
$ perl gets.pl --matrix=./example1/matrix_small_values.tsv --intensity=4
--output=./example1/matrix_small_values.intensity4
```
(15)

The resulting heatmaps are shown in Figure 5.

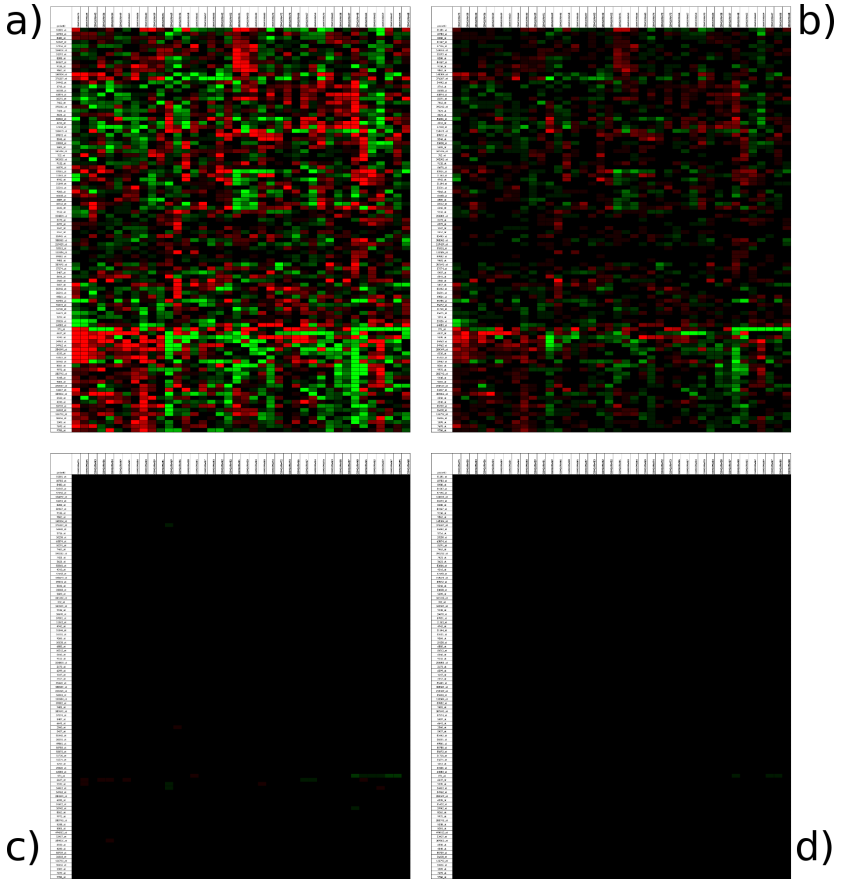

Figure 5. Outputs of (a) example 12; (b) example 13; (c) example 104; (d) example 15.

Again, intensities defined by IQR and IDR adjust automatically to the data range, while in this case user-defined intensities produce very dark heatmaps because most expression values are much lower than the user-defined ranges.

In any case, correct intensity calculation always requires the values in the matrix to be centered around 0. A warning message will be prompted if the input matrix does not comply with this requirement (see **Centering heatmap data** and **Troubleshooting** below).

7 Gene information (--geneinfo)

GETS is not only designed to write multi-colored spreadsheet-ready heatmaps but also to add gene/probe and sample information to the output file. The `--geneinfo` argument incorporates gene/probe information to the output. This argument passes the name of a tab-separated text file where the rows are the genes/probes and the columns contain information on each one of them (e.g. gene names, chromosomal locations, fold change values, *p*-values). Its contents will be added to the right of the matrix in the output file. This file must have *n* rows (i.e. one row for each gene/probe in the intensity matrix, plus the first row with the headers). By default, GETS expects the order of the genes/probes to exactly match their order in the matrix (if this is not the case, see the **Matching gene names** section). In this case, there is not need to supply the genes/probe names. The number of columns in this file is variable, depending entirely on the data that the user wishes to display. In the following example (adapted from GSE38713), we will add information on gene nomenclature and results data for three different contrasts. This file also contains some lines with comments (starting with #) that will be ignored.

```
$ perl gets.pl --matrix=./example2/matrix.tsv --
output=./example2/myoutput.withgeneinfo --geneinfo=./example2/geneinfo.tsv
```
(16)

The resulting heatmap is shown in Figure 6.

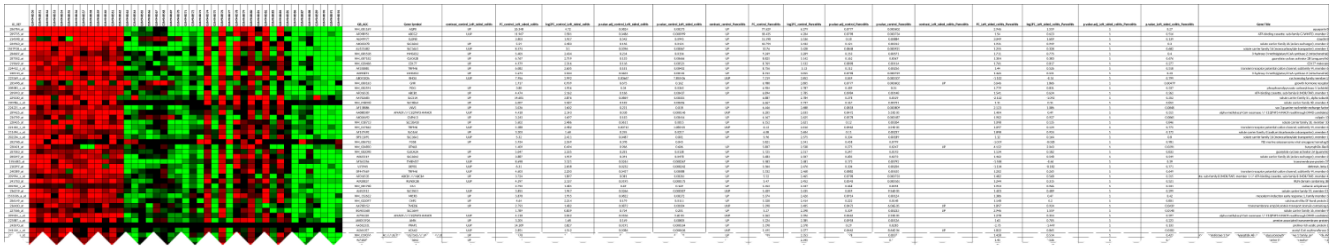

Figure 6. Output of example 16.

8 Sample information (--sampleinfo, --sampleinfo-format)

The `--sampleinfo` argument incorporates sample information to the output file. This information is supplied in a tab-separated text file that can have two formats, specified by the `--sampleinfo-format` argument. If `--sampleinfo-format=`*`columns`* (the default), columns in the `--sampleinfo` file are the samples and the rows contain information on each sample (e.g. gender, disease, treatment). In this case, the file must have *m* columns (i.e. one column for each sample in the intensity matrix, plus the first column with the headers). By default, GETS expects the order of the samples exactly match their order in the matrix (if this is not the case, see the **Matching sample names** section). In this case, there is not need to supply the sample names. In the following example, the file *./example2/sampleinfo.tsv* incorporates information on five sample features: gender, age, evolution time of the disease, extension of the disease and type of treatment. It looks like:

gender	...	male	female	female	male	female	male	female	...
age_years	...	40	50	40	50	60	53	63	...
evolution_time_years	...	--	--	13	6	22	8	10	...
disease_extension	...	--	--	Left-sided colitis	Pancolitis	Left-sided colitis	Left-sided colitis	Left-sided colitis	...
treatment	...	--	--	Azathioprine	Azathioprine	5-ASA	5-ASA	5-ASA	...

To incorporate this information to the output, we run:

```
$ perl gets.pl --matrix=./example2/matrix.tsv
--output=./example2/myoutput.withsampleinfo --
sampleinfo=./example2/sampleinfo.tsv
```
(17)

We don't specify `--sampleinfo-format=`*`columns`* because it is the default. The resulting heatmap is shown in Figure 7.

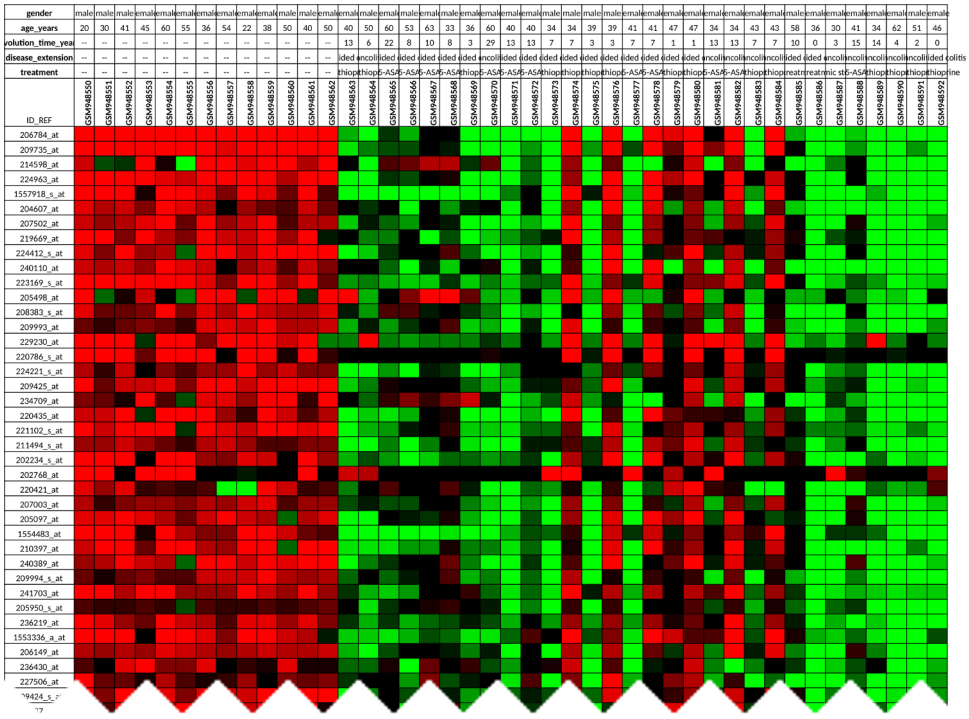

Figure 7. Output of example 17.

If `--sampleinfo-format=`*`rows`*, rows in the `--sampleinfo` file are the samples and the columns contain the sample information (i.e. it is the transposed version of the file in *columns* format). In this case, the file must have *m* rows (i.e. one row for each sample in the intensity matrix, plus the first row with the headers) and by default the order of the samples must exactly match their order in the matrix (if this is not the case, see the **Matching sample names** section).. The file *./example2/sampleinfo.rows.tsv* incorporates the same sample information but in *rows* format:

gender	age_years	evolution_time_years	disease_extension	treatment
...	...	...	...	...
male	40	--	--	--
female	50	--	--	--
female	40	13	Left-sided colitis	Azathioprine
male	50	6	Pancolitis	Azathioprine
female	60	22	Left-sided colitis	5-ASA
male	53	8	Left-sided colitis	5-ASA
female	63	10	Left-sided colitis	5-ASA
male	33	8	Left-sided colitis	5-ASA
female	36	3	Left-sided colitis	Azathioprine
female	60	29	Pancolitis	5-ASA
female	40	13	Left-sided colitis	5-ASA
female	40	13	Left-sided colitis	5-ASA
...	...	...	...	...

In the following example, we incorporate the same sample information as in (17) but using sample information file formatted as rows:

`$ perl gets.pl --matrix=./example2/matrix.tsv --output=myoutput.withsampleinforows – sampleinfo=./example2/sampleinfo.rows.tsv --sampleinfo-format=rows`	(18)

The output file will be identical to that obtained in the example (17).

Sample information can be combined with gene information into the sample output file. Here we combine the gene information of example (16) with the sample information of example (17):

`$ perl gets.pl --matrix=./example2/matrix.tsv --output=./example2/myoutput.withsampleandgeneinfo -- sampleinfo=./example2/sampleinfo.tsv --geneinfo=./example2/geneinfo.tsv`	(19)

The resulting heatmap is shown in Figure 8.

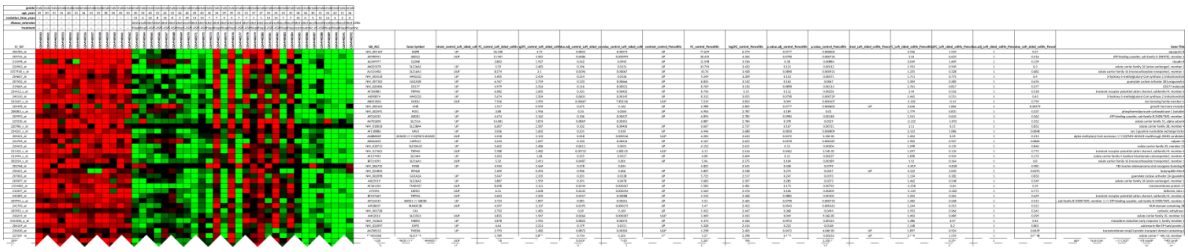

Figure 8. Output of example 19.

9 Adding colors to gene and sample information (--colors, --palette)

By default gene and sample information is displayed without colors. The argument `--colors` is used to add background colors to the information cells. This argument passes a tab-delimited text file with four columns containing (from left to right):

Column 1	{*GENEINFO*|*SAMPLEINFO*}
Column 2	Column name in the --geneinfo file (if column 1 is *GENEINFO*); Column name in the --sampleinfo file (if column 1 is *SAMPLEINFO* and --sampleinfo-format=*columns*); Row name in the --sampleinfo file (if column 1==*SAMPLEINFO* and --sampleinfo-format=*rows*)
Column 3	Value
Column 4	Color. If color is {*GRADIENT_RED*|*GRADIENT_GREEN*| *GRADIENT_TWO_COLORS*}, Column 3 will be ignored and a red/green/red-white-green gradient (centered around 0) will be displayed instead of discrete colors (non-numeric values will be shown in grey). If color is "AUTO", Column 3 will be ignored and values will be colored randomly using the palette selected with `--palette` (if the number of distinct values exceeds the number of colors, colors will be duplicated). "AUTO" is only accepted if Column 1 is "SAMPLEINFO".

For instance, in file *./example2/colors.tsv*:

SAMPLEINFO	gender	male	ORANGE
SAMPLEINFO	gender	female	CYAN
SAMPLEINFO	age_years		GRADIENT_RED
SAMPLEINFO	evolution_time_years		GRADIENT_GREEN
SAMPLEINFO	disease_extension		AUTO
SAMPLEINFO	treatment		AUTO
GENEINFO	contrast_control_Left_sided_colitis	UUP	RED
GENEINFO	contrast_control_Left_sided_colitis	UP	DARK_RED
GENEINFO	contrast_control_Left_sided_colitis	DDW	GREEN
GENEINFO	contrast_control_Left_sided_colitis	DW	DARK_GREEN
GENEINFO	contrast_control_Pancolitis	UUP	RED
GENEINFO	contrast_control_Pancolitis	UP	DARK_RED
GENEINFO	contrast_control_Pancolitis	DDW	GREEN
GENEINFO	contrast_control_Pancolitis	DW	DARK_GREEN
GENEINFO	contrast_Left_sided_colitis_Pancolitis	UUP	RED
GENEINFO	contrast_Left_sided_colitis_Pancolitis	UP	DARK_RED
GENEINFO	contrast_Left_sided_colitis_Pancolitis	DDW	GREEN
GENEINFO	contrast_Left_sided_colitis_Pancolitis	DW	DARK_GREEN
GENEINFO	membrane	yes	BLUE
GENEINFO	ER	yes	YELLOW
GENEINFO	extracellular	yes	LIGHT_ORANGE
GENEINFO	epithelium	yes	PURPLE
GENEINFO	lipid-related	yes	LIME
GENEINFO	cell-cycle-related	yes	DARK_BLUE
GENEINFO	FC_control_Left_sided_colitis		GRADIENT_TWO_COLORS
GENEINFO	FC_control_Pancolitis		GRADIENT_TWO_COLORS
GENEINFO	FC_Left_sided_colitis_Pancolitis		GRADIENT_TWO_COLORS

Row 1: the value "male" in the "gender" row of the sample information section must be shown in orange.
Row 2: the value "female" in the "gender" row of the sample information section must be shown in cyan.
Row 3: the "age_years" row of the sample information section must be shown in a gradient of reds.
Row 4: the "evolution_time_years" row of the sample information section must be shown in a gradient of greens.
Row 5: the distinct values in the "disease_extension" row of the sample information section must be shown in randomly picked colors.
Row 6: the distinct values in the "treatment" row of the sample information section must be shown in randomly picked colors.
Row 7-18: the values "UUP", "UP", "DDW" and "DW" in the columns "contrast_control_Left_sided_colitis", "contrast_control_Pancolitis" and "contrast_Left_sided_colitis_Pancolitis" will be shown in RED, DARK_RED, GREEN and DARK_GREEN, respectively.
Rows 19-24: value "yes" in columns "membrane", "ER", "extracellular", "epithelium", "lipid-related", "cell-cycle-related" will be shown in BLUE, YELLOW, LIGHT_ORANGE, PURPLE, LIME and DARK_BLUE, respectively.

Rows 25-27: values in columns "FC_control_Left_sided_colitis", "FC_control_Pancolitis" and "FC_Left_sided_colitis_Pancolitis" will be shown in a green/red gradient (centered around 0).

```
$ perl gets.pl --matrix=./example3/matrix.tsv
--output=./example3/myoutput.colors --intensity=IDR --
sampleinfo=./example3/sampleinfo.tsv --geneinfo=./example3/geneinfo.tsv --
colors=./example3/colors.tsv
```

(17)

The resulting heatmap is shown in Figure 9, and the sample information section shown in Figure 10.

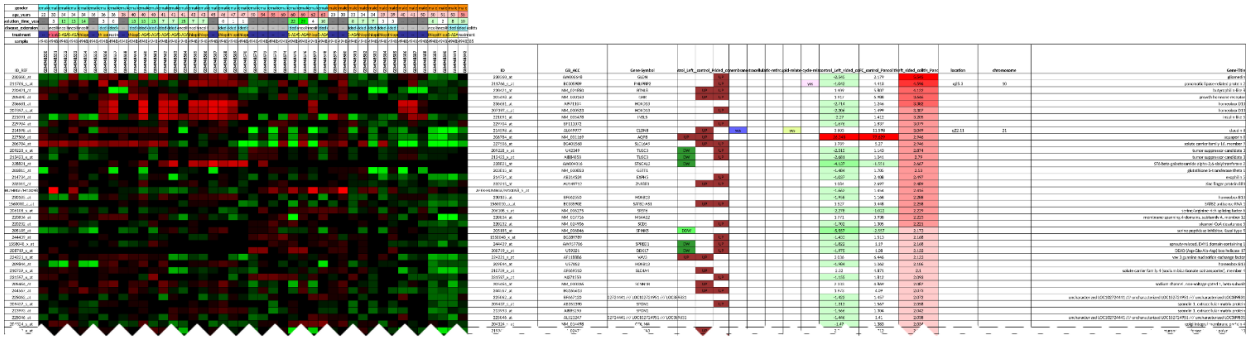

Figure 9. Output of example 17.

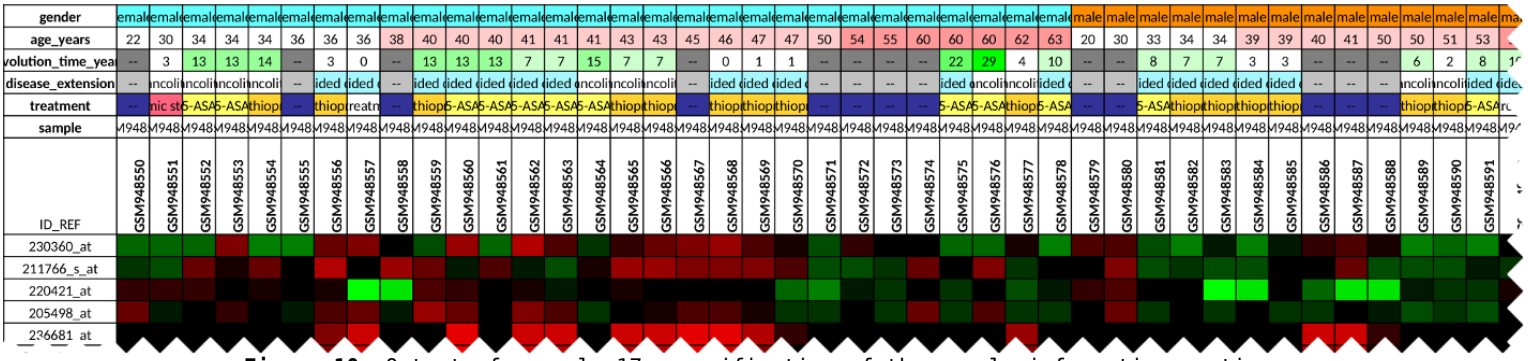

Figure 10. Output of example 17, magnification of the sample information section.

Because of the restriction in the number of colors available in MS Excel, the color list is limited to 18 colors. This color schema has two levels of brightness (bright and muted), specified by the `--palette` argument (Figure 11).

	palette	
	bright	**muted**
DARK_RED		
RED		
PURPLE		
ORANGE		
MAGENTA		
LIGHT_ORANGE		
YELLOW		
LIGHT_YELLOW		
LIME		
GREEN		
DARK_GREEN		
CYAN		
LIGHT_BLUE		
BLUE		
DARK_BLUE		
DARK_GREY		
GREY		
WHITE		

Figure 11. Available palettes.

Any unrecognized color names in `--colors` will result in an error message. Unrecognized column/row names or values in `--colors` file will be ignored.

Matching gene names (--geneinfo-ordered)

By default, GETS assumes that the rows in the gene information file exactly match the rows in the matrix file (i.e. genes/probes are in the same order in both files). If this is not the case, GETS can try to match the gene/probe names in both files if `--geneinfo-ordered=FALSE`. This option has two restrictions: (a) no duplicate gene/probe names are allowed in the matrix file, and (b) the first column of the matrix file and the first column of the gene information file must contain the gene/probe names. The data in the folder *./example3/* uses the same data as *./example2/* but the order of the rows in *geneinfo.tsv* does not match that of *matrix.tsv*.

```
$ perl gets.pl --matrix=./example3/matrix.tsv --output=./example3/output
--geneinfo-ordered=FALSE --geneinfo=./example3/geneinfo.tsv
```
(18)

This option can also be used when the gene information file contains genes that are not present in the matrix file. For instance, the file *./example4/annotation.tsv* contains annotations for all human genes. If the gene information file contains more gene/probes than the matrix file, GETS will prompt a warning message.

```
$ perl gets.pl --matrix=./example4/matrix.tsv --
output=./example4/output.annotated --geneinfo-ordered=FALSE --
geneinfo=./example4/annotation.tsv
```
(19)

10 Matching

Matching sample names (--sampleinfo-ordered)

By default, GETS assumes that the order of the samples in the sample information file exactly matches the order of the samples in the matrix file. If this is not the case, GETS can try to match the sample names in both files if `--sampleinfo-ordered=FALSE`. This option has two restrictions: (a) no duplicated sample names

are allowed in the sample information file, and (b) the last row of the sample information file must contain the sample names if `--sampleinfo-format=columns` or the last column of the sample information file must contain the sample names if `--sampleinfo-format=rows`. If `--sampleinfo-format=columns` (default):

```
$ perl gets.pl --matrix=./example5/matrix.tsv --output=./example5/output
--sampleinfo-ordered=FALSE --sampleinfo=./example5/sampleinfo.tsv
```
(20)

If `--sampleinfo-format=rows`:

```
$ perl gets.pl --matrix=./example5/matrix.tsv --
output=./example5/output.rows --sampleinfo-ordered=FALSE --sampleinfo-
format=rows --sampleinfo=./example5/sampleinfo.rows.tsv
```
(21)

As with the gene information file, this argument can be used when there are samples that are not present in the matrix file:

```
$ perl gets.pl --matrix=./example5/matrix.tsv
--output=./example5/output.extra_samples --sampleinfo-ordered=FALSE --
sampleinfo=./example5/sampleinfo.extra_samples.tsv
```
(22)

If `--sampleinfo-format=rows`:

```
$ perl gets.pl --matrix=./example5/matrix.tsv
--output=./example5/output.extra_samples.rows --sampleinfo-ordered=FALSE
--sampleinfo=./example5/sampleinfo.rows.extra_samples.tsv --sampleinfo-
format=rows
```
(23)

11 Centering heatmap data (--center)

By default, GETS assumes that the values in the matrix are centered around 0. This is useful for visualization purposes since over-expressed (above 0) and under-expressed (below 0) values will be shown in shades of red and green, respectively in the GBR color schema. If the matrix is not centered (i.e. all values are positive), they will be displayed in shades of red only. When the `--center` argument is *TRUE*, GETS automatically centers the expression values for each gene by subtracting the median, so the value for cell (i_{ij}) is calculated as:

$$i_{ij} = e_{ij} - M(e_i)$$

where e_{ij} is the expression value for gene *i* in sample *j* and $M(e_i)$ represents the median expression value for gene *i* across all samples.

The following example generates a heatmap for an uncentered matrix without centering (Figure 12a):

```
$ perl gets.pl --matrix=./example1/matrix_not_centered.tsv
--output=./example1/matrix.notcentered
```
(24)

And the following example generates a heatmap for an uncentered matrix with `--center=TRUE` (Figure 12b):

```
$ perl gets.pl --matrix=./example1/matrix_not_centered.tsv
--output=./example1/matrix.centered --center=TRUE
```
(25)

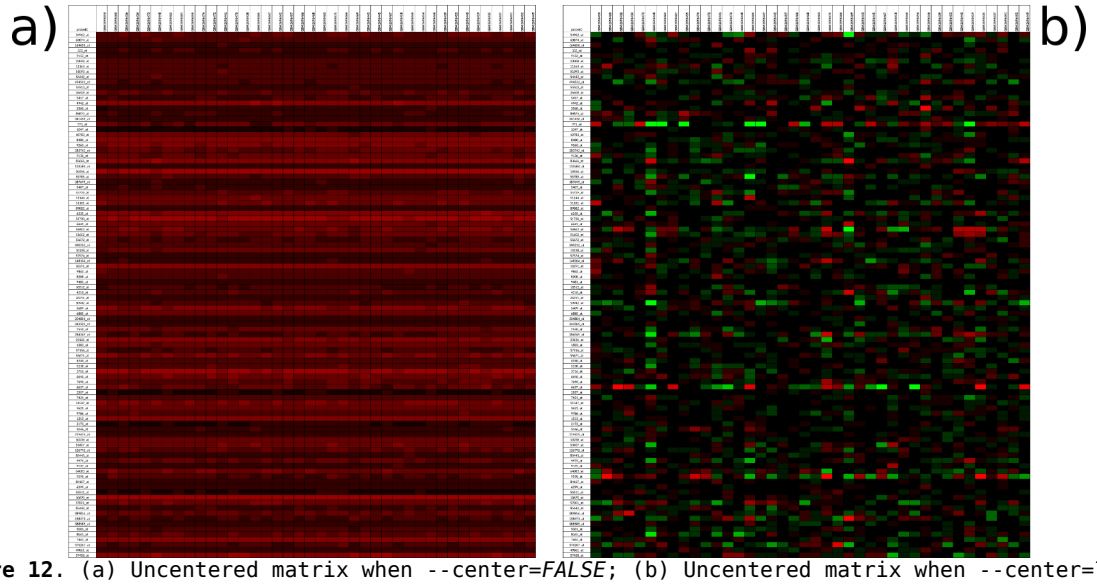

Figure 12. (a) Uncentered matrix when --center=*FALSE*; (b) Uncentered matrix when --center=*TRUE*.

12 Other options (--overwrite, --verbose)

If the `--overwrite=FALSE` (the default), the output file will not overwrite the existing one and an error message will be prompted instead.
If the `--verbose=TRUE` (the default), GETS will stdout detailed information on the progress. Otherwise, it will run silently.

13 Other examples

This example (26) is based on data from the *./example5/* folder with: (1) added gene and color information, (2) heatmap colors calculated using the *IDR* option, and (3) a muted palette for sample and gene information colors.

```
$ perl gets.pl --matrix=./example5/matrix.tsv --
output=./example5/output.complete --sampleinfo-ordered=FALSE --
sampleinfo=./example5/sampleinfo.tsv --colors=./example5/colors.tsv --
geneinfo=./example5/geneinfo.tsv --palette=muted --intensity=IDR
```
(26)

The following example (27) is based on the data from GSE94648:

```
$ perl gets.pl --matrix=./example_GSE94648/GSE94648_series_matrix.tsv --
output=./example_GSE94648/output --
geneinfo=./example_GSE94648/GSE94648_gene.info.tsv --center=TRUE --
sampleinfo=./example_GSE94648/GSE94648_series_samples.rows.tsv --
sampleinfo-format=rows --sampleinfo-ordered=FALSE --palette=muted --
colors=./example_GSE94648/GSE94648_colors.tsv --intensity=IDR
```
(27)

The following example (28) is based on data from GSE51601 experiment from GEO database. Because the data in the matrix is not centered around 0, we use `--center=TRUE`:

```
$ perl gets.pl --matrix=./example_GSE51601/GSE51601_series_matrix.tsv --
output=./example_GSE51601/output --
```
(28)

```
geneinfo=./example_GSE51601/GPL17826_040666_D_GEO_20131022.tsv --
center=TRUE
```

14 Using GETS online

1 Getting started

The home webpage of GETS contains an example that allows the user to visualize a simple dataset. This example contains four different types of file:

-the gene expression matrix
-the gene information data
-the sample information data
-the colors data

All files are available by clicking on the icon present under each section. The only mandatory file is the gene expression matrix (Figure 13).

2 Data upload

All four files must be uploaded to their correct area (Figure 13). Refer to sections xxx for examples of these file formats.

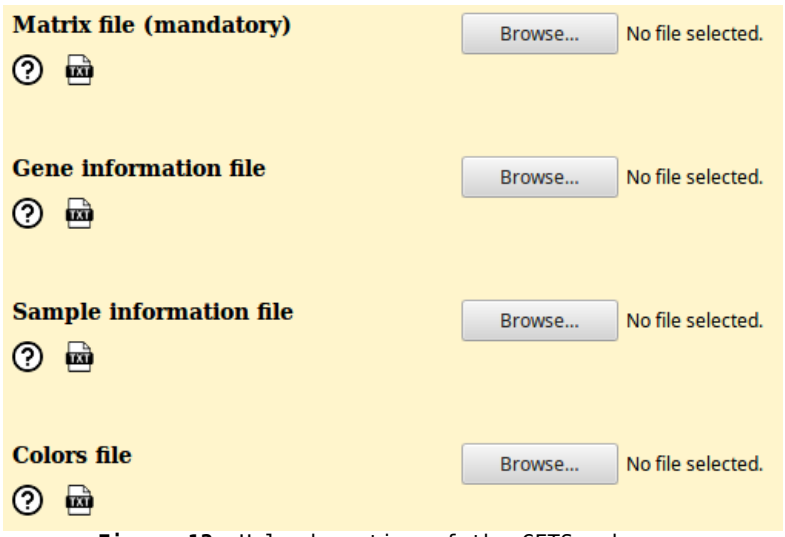

Figure 13. Upload section of the GETS webpage.

3 Options

After uploading the file(s), the user must select the options that match the format of the input files (Figure 14). Refer to sections xxx for examples of these file formats.

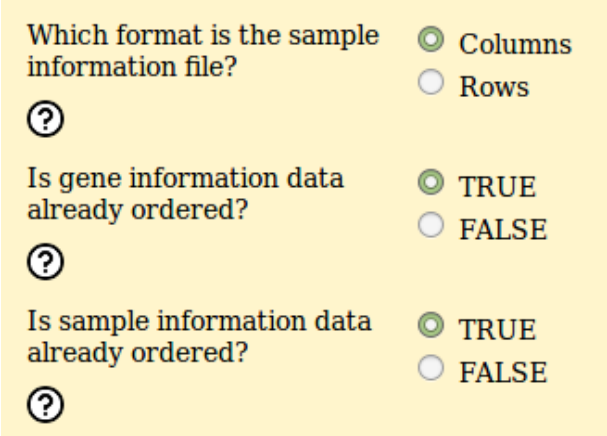

Figure 14. File format section of the GETS webpage.

Also, the user can choose the color palette to be used to color gene and sample data (*Palette*; Figure 15), the method to calculate the heatmap intensity limits (*Limit for heatmap intensities*; Figure 15), the heatmap color scheme (*Heatmap colors*; Figure 15), and whether or not the values in the matrix should be centered (*Should expression values be centered?*; Figure 15).

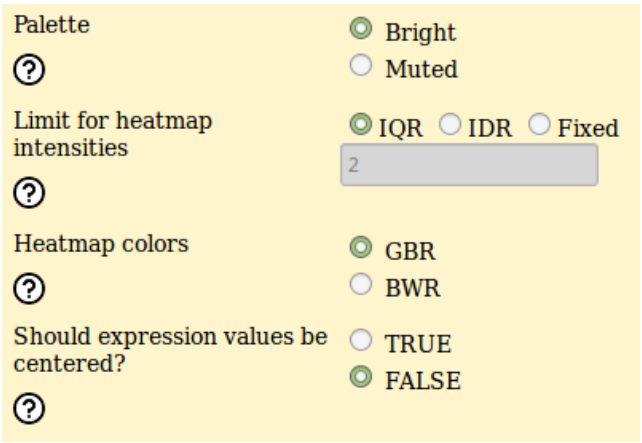

Figure 15. Colors and centering section of the GETS webpage.

4 Output

The "Submit Form" button will process the input data (Figure 16). Selecting Verbose run mode will output an extended description of the input data and the progress of GETS. Otherwise, the output will be only the link to download the output file (Figures 17 and 18).

Figure 16. Run mode and Submit Form section of the GETS webpage.

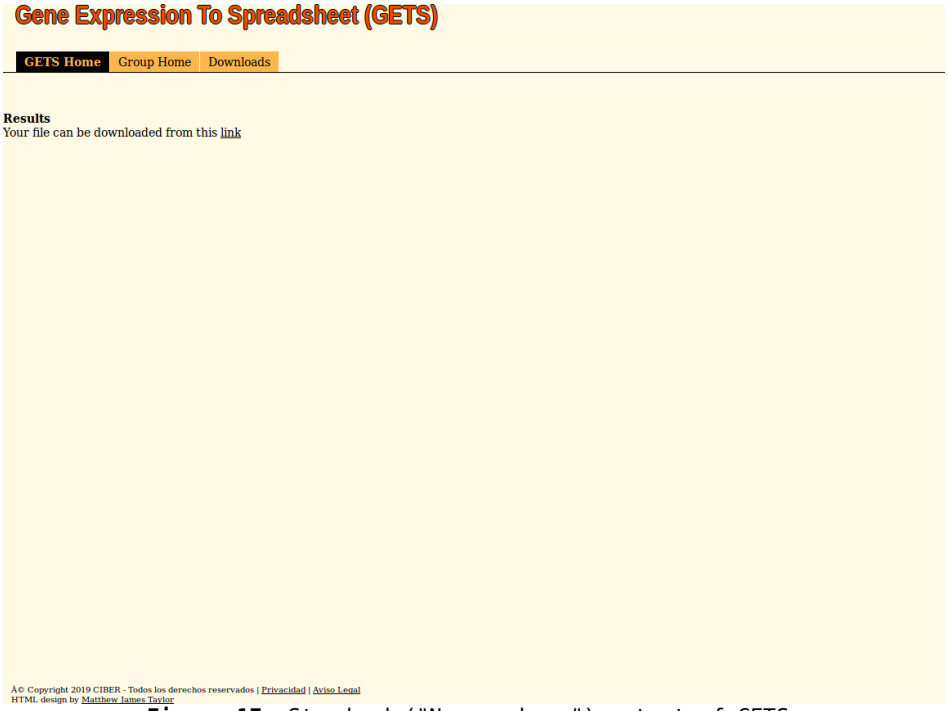

Figure 17. Standard ("Non-verbose") output of GETS.

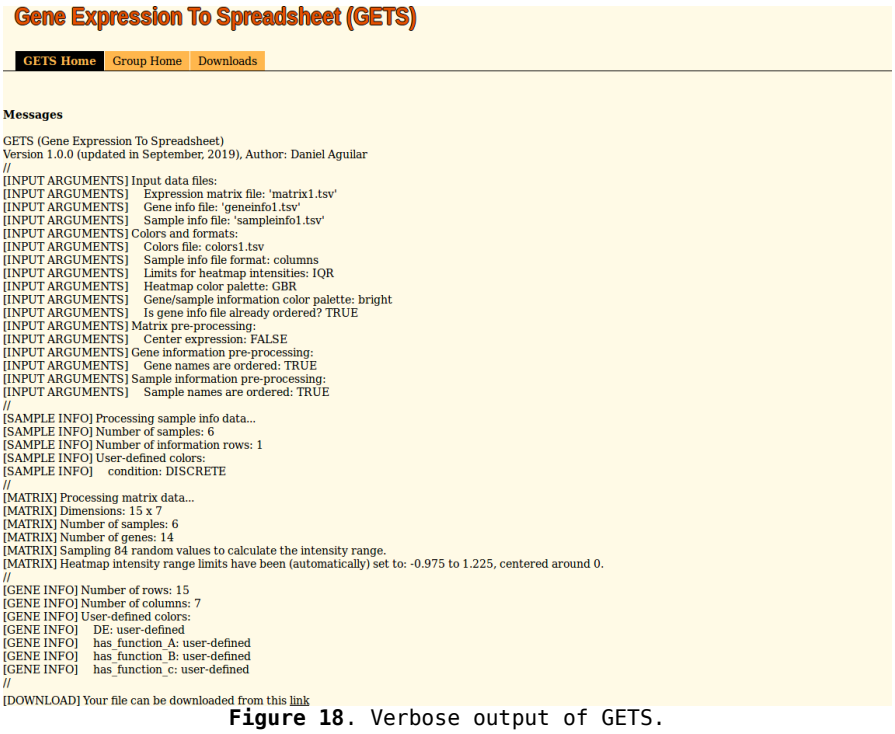

Figure 18. Verbose output of GETS.

15 Troubleshooting

The heatmap colors are very bright

The default `--intensity=IQR` works well for the majority of centered gene expression matrices. If the heatmap looks too bright, `--intensity=IDR` will darken the colors. If some cells still look too bright, it may be because there are some extreme outliers or because the values are not symmetrically

distributed around zero (which is rare). In these cases, try `--intensity=<number>`.

The heatmap is all red
Most likely the data in the matrix was not centered. Try `--center=TRUE`.

Some cells in the heatmap are too bright
If some cells in the heatmap are very bright and the rest are very dark (even after centering with `--center=TRUE`), chances are that the expression matrix was not normalized. Some experimental methodologies (such as RNA-Seq) generate expression matrices with extremely large expression values that must be normalized before any analysis or visualization. GETS does not incorporate any method for matrix normalization, although normalization methods can be found in popular R packages such as *limma* and *edgeR*.

The sample information section is (mostly) empty
This very in very specific cases where `--sampleinfo-ordered=FALSE` and `--sampleinfo-format` is incorrectly set as `--sampleinfo-format=rows`.

16 License