# Gene Expression Analysis Using DVX

**Haim Bar**[1], haim.bar@uconn.edu

**Elizabeth D. Schifano**[1], elizabeth.schifano@uconn.edu

[1] Department of Statistics, University of Connecticut

February 7, 2018

## Contents

Introduction ..... 3
Installing and starting the DVX software ..... 3
The Input Data ..... 4
DVX tabs ..... 5
The Summary tab ..... 5
The Plots tab ..... 6
Boxplot ..... 6
Histogram ..... 6
Flat Histograms ..... 6
Principal Components ..... 6
Heat Map ..... 7
Means, Variances, and Coefficients of Variation ..... 7
Mean vs. Variance ..... 8
Bland Altman ..... 8
Notes ..... 8
The Filter/Transform Tab ..... 8
The Save Tab ..... 11
The Analyze Tab ..... 11
The Results Tab ..... 13
The Save Report Tab ..... 15
Quitting the program ..... 15
Case Studies ..... 15
The REST dataset ..... 15
A brief description of the data - Geo Data Set 5204 ..... 15
Descriptive statistics ..... 16

Plots ..... 17
Filtering and/or transforming the data ..... 19
Differential analysis ..... 21
Saving Results ..... 24
Autism and Copy Number Variation of human 16p11.2 - Geo DataSet 4430 ..... 25
A brief description of the data ..... 25
Descriptive statistics ..... 25
Plots ..... 27
Filtering and/or transforming the data ..... 30
Differential analysis ..... 31
Data from other platforms ..... 34
Citations ..... 35

## Introduction

DVX is an interactive program written in R, which can be used to perform **D**ifferential **V**ariation and e**X**pression analysis of gene expression (or similar) data.

DVX uses two mixture distributions within a linear model framework to assess both differential dispersion and differential expression. DVX additionally provides many graphical visualization options, as well as several common data pre-processing options. To aid in model comparison, the package can also run the LIMMA model (G. K. Smyth 2004) for analyzing differential expression between two treatment groups.

This guide provides information on the DVX input data types; DVX installation; DVX preprocessing, visualization, and analysis tools; and two case studies. It also provides information on how to handle data from other platforms (e.g. count data from next-generation sequencing platforms).

The DVX homepage is https://haim-bar.uconn.edu/software/DVX/ . For questions, comments, and suggestions, please contact Haim Bar

## Installing and starting the DVX software

To use DVX, you must first install R (version >= 3.3.2) (R Core Team 2016) which is available from https://www.r-project.org/ and RStudio (version >= 1.0.136) (RStudio Team 2015) which is available from https://www.rstudio.com/products/rstudio/download/

DVX requires the following packages:

- Biobase (Huber et al. 2015)
- DT (Xie 2016)
- limma (M. E. Ritchie et al. 2015)
- qvalue (Andrew J. Bass, Dabney, and Robinson 2015)
- rtf (Schaffer 2013)
- shiny (Chang et al. 2017)

These packages should be installed prior to running DVX. Once R and RStudio are installed, start RStudio and in the Console, type the following:

```
source("https://bioconductor.org/biocLite.R")
biocLite("Biobase")
biocLite("limma")
biocLite("qvalue")
install.packages("shiny")
install.packages("DT")
install.packages("rtf")
```

There are two ways to run DVX - locally or remotely. To run it locally, download the file https://haim-bar.uconn.edu/wp-content/uploads/sites/1740/2018/02/DVX.zip and

unzip it in a folder on your computer. Then, start RStudio and use the **File > Open File…** menu to open the file server.R under the DVX folder. Click on the small triangle to the right of the "Run App" button, and select "**Run External**", and then click on "Run App". Alternatively, in the console, type the following:

```
runApp('DVXdir',display.mode="no", launch.browser=FALSE, port=2197)
```

replace DVXdir with the name of the folder where you saved the DVX files. Click here to see a screenshot that shows the two ways to run DVX locally from the RStudio user interface.

You may also run DVX *remotely*, which means that you do not have to download and unzip the source code on your computer. Simply type the following in the RStudio Console:

```
runUrl(https://haim-bar.uconn.edu/wp-content/uploads/sites/1740/2018/02/DVX.zip')
```

After that, the DVX user interface will appear on your default web browser and the "New Project" tab will be displayed. Click here for a screenshot. In the sidebar on the left, click the "Browse" button and select the saved ExpressionSet file. The sidebar will now show a link to "Load a different data set", which takes you back to the New Project screen.

When you open an ExpressionSet dataset, the main panel has 7 tabs at the top of the page. These tabs are described in detail below.

## The Input Data

The DVX software uses as input an *ExpressionSet* object from the Biobase package (Huber et al. 2015). In this documentation, we use datasets from the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO), which is a public repository for array- and sequence-based data. As of November 2017, the repository contains more than 4,300 datasets.

To create an ExpressionSet dataset on your own, please follow the instructions in the Bioconductor's introduction to ExpressionSet document. To analyze data from the NCBI GEO repository, first download the data file from the GEO DataSets (GDS) web page. For example, you may enter lipoprotein mice and press the Search button to see available items, or, if you know the accession or dataset number, e.g., GDS6176, enter it in the search box. Click on the "Download Data" link, and then right-click (ctrl+click on a Mac) on the link to the DataSet SOFT file (e.g., "GDS6176.soft.gz") to save it to a folder of your choice. Alternatively, you can use R to download the GEO dataset file, using the *GEOquery* package, (S. Davis and Meltzer 2007) like this:

```
library("GEOquery")
gds <- getGEO("GDS6176", destdir = "~/Desktop")
```

Once the data is stored on your computer, you have to save it as an ExpressionSet file. To do that, use the GDS2eSet function:

```
eset <- GDS2eSet(gds)
save(eset, file="~/Desktop/eSet6176.RData")
```

For data in GSE format, the data is returned as a list. You may then do the following:

```
gse <- getGEO("GSE11675")    # very small dataset
eset = gse[[1]]              # ExpressionSet
save(eset, file="~/Desktop/eSetGSE11678.RData")
```

More details about GEOquery are provided here.

# DVX tabs

## The Summary tab

This tab includes a short summary of the dataset currently being used. It consists of three tables.

The first table contains the ExpressionSet metadata, including:

- The name of the contributor
- The lab that contributed the data.
- Contact information
- Title
- URL
- PubMed IDs
- The number of samples
- The number of features (genes)
- Abstract

Note that some datasets may have only a subset of these metadata fields.

The second table contains gene expression statistics, across all samples, including the five number summary of 'expression' values - minimum, maximum, Q1=first quartile, Q2=median, Q3=third quartile. The table also shows the overall mean, and the number of missing expression values. In this version, missing values are not imputed and in the subsequent tabs only complete cases are used.

The third table contains summaries for phenotype data. For factors, it shows the different levels and the total number of samples in each level, and for numeric variables (other than gene expression) it shows the same summary statistics as for the gene expression.

# The Plots tab

## Boxplot

Displays horizontal boxplots by subject for all subjects. Subject-specific boxplots are color-coded by the levels of the selected factor. The factor (and subsequent color-coding) can be changed using the "Plot by Factor" drop-down menu.
Each subject-specific boxplot shows the distribution of that subject's features (e.g. gene expression values across all genes). In each boxplot, the horizontal length of the box corresponds to the interquartile range, with the median indicated with a solid vertical line within the box. Individual points depicted beyond the whiskers in either direction are indicative of extreme observations.

## Histogram

Displays a frequency histogram of features (e.g., gene expression values) for a given subject, as indicated by the value under "Select Subject ID".
The subject may be changed using the "Select Subject ID" drop-down menu.
These plots provide an indication of the shape of the distribution of features. For example, the histograms show whether the distribution of features is unimodal or multimodal, and whether it is symmetric or skewed.

## Flat Histograms

Displays "flattened histograms" by subject for all subjects. Each colored horizontal line in the color-matrix corresponds to one subject, and can be interpreted as follows: suppose you were to construct a frequency histogram for each subject. For a given subject, rather than plotting the frequencies as the heights in the frequency histogram, the frequencies are converted into a color intensity and plotted in a single horizontal line, with red indicating high frequency and yellow/white indicating low frequencies. The colored horizontal lines corresponding to the frequency intensities are stacked for all subjects, to give the appearance of a color-matrix. Like the Histogram option, these plots provide an indication of whether or not the distribution of features for each subject is unimodal or bimodal, but unlike the Histogram option, we can view the modality of feature distributions for all subjects simultaneously.

## Principal Components

Principal components analysis can be used to reduce the dimensionality of multivariate datasets through the identification of a reduced number of components that retain most of the variability in the original dataset. The principal components plots shows the second principal component (PC2) against the first principal component (PC1).
Each point in the plot represents one subject, where the subjects are color-coded and labeled according to the levels of the selected factor. The factor (and subsequent color-coding) can be changed using the "Plot by Factor" drop-down menu. The principal component plot is generally of most use in identifying genetic relationships between subjects; subjects with points located near each other in the plot are more genetically

similar than subjects with points in the plot that are far away. If points corresponding to a factor level are clustered together, it may suggest that there is a systematic difference in feature values between the factor levels.

## Heat Map

Displays the feature values, color-coded by intensity of the measurement (low=yellow, high=red), for each feature (row) and subject (column). The columns are reordered, so that similar subjects appear close to each other. The order of the features is the same as in the input data. If there are rows with uniformly low intensity, then the corresponding features may be indistinguishable from background noise, and data filtering may be necessary. Similarly, rows with uniformly high intensity may correspond to generally abundant genes (e.g., housekeeping genes).

Similarly, columns with uniformly low or high intensity may suggest that there are subject specific effects which have to be accounted for (either via transformation, or by including predictors in the model used for the DE analysis.)

The dendrogram along the top shows how the subjects (columns) are clustered, according to their overall similarity across all features.
The colored bars between the dendogram and the heatmap represent the subjects, color-coded according to levels within the selected factor. If bars of the same color are clustered together, it may suggest that there is a systematic difference in feature values between the levels of the factor. The factor and subsequent color-coding can be changed using the "Plot by Factor" drop-down menu.

## Means, Variances, and Coefficients of Variation

For each type (Means, Variances, or Coefficients of Variation), two plots are available: Scatterplot and Histogram. The displayed statistics are the sample means, natural logarithm of the sample variances, and the ratios of the sample standard deviation to the sample mean for the Means, Variances, and Coefficients of Variation types, respectively.

### Scatterplot

Displays the statistics for each feature (computed across subjects) for a particular level (as per "Select Level") of a particular factor variable (as per "Plot by Factor"). Features appear in the plot along the horizontal axis in the same order that they appear in the gene expression data. Levels and factors may be changed using the drop-down menus. Features with unusually high or low statistics may need to be investigated further.

### Histogram

Displays a frequency histogram for the statistics of each feature (computed across subjects) for a particular level (as per "Select Level") of a particular factor variable (as per "Plot by Factor"). Levels and factors may be changed using the drop-down menus. Features with unusually high or low statistics may need to be investigated further.

### Mean vs. Variance

Displays the mean-log(variance) relationship. Each point represents the sample mean and log(variance) for each feature (computed across all subjects sharing the selected level of the selected factor).
If there is no trend in the data, then the variance remains constant as the mean changes. If a trend is observed (for example, the variances increase as the means increase or decrease), then a transformation of the data may have to be considered.

### Bland Altman

This plot is used to assess agreement between two sets of measurements (Bland and Altman 1986, 2003), and is used here to assess agreement between two levels of a factor. For a given pair of levels (as per the "Select first level" drop-down option and "Select second level" drop-down option) within a given factor (as per "Plot by Factor"), we compute the mean expression values across all subjects in the two selected levels. Denote them as m1 and m2. Then the horizontal axis represents the average of m1 and m2, and the vertical axis shows the difference between m1 and m2. Each feature corresponds to one point in the scatterplot.
Levels and factors may be changed using the drop-down menus. The plot shows the amount of disagreement between the two levels (via the differences) and displays how this disagreement relates to the magnitude of the measurements (via the averages). If the overall average difference between two levels is not 0, it may suggest that there are systematic differences between the features values across the levels of the selected factor. Patterns in the plot (e.g. the differences increase as the averages increase or decrease) may suggest that a transformation is needed. Extremely large or small differences (along the vertical axis) suggest that there may be features which have significantly different variances for different levels of the selected factor.

### Notes

To save a plot, check the "Include in report" box at the bottom of the sidebar. Then, use the "Save Report" tab to export all the selected plots to a Rich-Text Format (RTF) file.

## The Filter/Transform Tab

This tab allows the user to manipulate the data by applying certain transformations and choosing filtering criteria. The sidebar has three parts, which allow users to

- exclude subjects
- exclude features
- transform the feature data

The plot area contains the flat histogram based on the current selection of filtering criteria and feature data transformation.

### Exclude subjects

Removes subjects with missing values (coded as NA) in particular variables (as selected from the drop-down list) from the dataset. If you choose the "Any" option, all subjects with NA in any of the variables will be excluded.

### Exclude features

Exclusion of features is done by comparing a function of the expression data with some threshold. In other words, the general form of the exclusion criterion is

F(feature data) OPERATOR threshold

The function, F, is one of the following

- **None**: do not remove any features. With this option, any operator option or threshold value can be specified as they will be ignored.
- **min**: for each feature the minimum of all the expression values across all subjects is compared with the given threshold.
  For example, in conjunction with operator ">=" and threshold=15, "min >= 15" would remove all features with a minimum value (across all subjects) greater than or equal to 15. This can be used to remove features with exceptionally high expression levels across all subjects.
- **max**: for each feature the maximum of all the expression values across all subjects is compared with the given threshold.
  For example, in conjunction with operator "<=" and threshold=6, "max <= 6" would remove all features with a maximum value (across all subjects) less than or equal to 6. This can be used to remove features with exceptionally low expression levels across all subjects.
- **mean**: for each feature the mean of all the expression values across all subjects is compared with the given threshold.
  For example, in conjunction with operator "<" and threshold=10, "mean < 10" would remove all features with a mean value (across all subjects) less than 10. This can be used to remove features with exceptionally high or low average expression levels.
- **median**: for each feature the median of all the expression values across all subjects is compared with the given threshold.
  For example, in conjunction with operator ">=" and threshold=12, "median >= 12" would remove all features with a median value (across all subjects) greater than or equal to 12.
- **IQR**: for each feature the inter-quartile range of the expression levels, across all subjects, is compared to the given threshold. For example, in conjunction with operator "==" and threshold=0, "IQR == 0" would remove all features with an IQR equal to 0. This situation may occur if the feature is either hard to detect, and a lower bound ("feature detection level") is provided for at least half the subjects, or it is highly abundant, and an upper bound ("saturation level") is provided for at least half the subjects.
- **var**: for each feature the variance of the expression levels, across all subjects, is compared to the given threshold. For example, in conjunction with Operation "==" and Threshold=0, "var == 0" would remove all features with a sample variance equal

to 0. This situation may occur if the feature is either hard to detect, and a lower bound ("feature detection level") is provided for all features, or it is highly abundant, and an upper bound ("saturation level") is provided for all features.

The available operators are:

- **<=** : less than or equal to
- **=>** : greater than or equal to
- **<** : less than
- **>** : greater than
- **==** : equal
- **!=** : not equal

Threshold values: can be any real number. Use the flat histogram plot to choose a reasonable threshold value.

### Transform feature data

Transforms feature data according to one of three options, and is to be used in conjunction with a "Parameter c" value.

- **identity**: no transformation is performed; any Parameter c value may be specified as it will be ignored.
- **log2(c + x)**: performs a logarithm transformation (base 2) on the feature data shifted by c units, as specified through the Parameter c value.
  Setting Parameter c to 0 will result in the usual logarithm transformation (base 2) on the feature data. The parameter c is used to prevent taking the logarithm of non-positive values. If the selected parameter leads to taking the logarithm of negative values, the software adds a constant to the user's choice.
- **Add Gaussian noise(0, sd=c)**: Add random noise to the feature data, where the noise is independently Normally distributed for each feature with mean 0 and standard deviation c, as specified through the Parameter c value.
- It is also possible, and indeed, recommended, to **set equal medians across subjects**, to reduce the subject-specific effect.

### Notes

- After all the filtering and transformation criteria have been selected, press the Apply button, and check the updated flat histogram. After you click Apply, the transformed data is used in subsequent plotting and analyses during the active session, even if you don't save the transformed data as a new dataset. You can undo this action by removing the filter/transform criteria and click Apply again.
- Only one combination of subject exclusion criterion, feature exclusion criterion, and transformation, can be done. If more exclusion criteria or tranformations are needed, perform one at a time, and use the Save tab to create intermediate versions of the filtered/transformed data.

## The Save Tab

This tab is used to save modified datasets formed via filtering, transforming, subsetting variables from the originally loaded dataset (via the Filter/Transform tab).

**Save as**: Specifies file name of dataset to be saved. Type in [filename].RData, substituting [filename] with any name of your choice. The default file name is temp.RData.

**Predictors**: Uncheck any predictor variable that you do not want to keep in the dataset to be saved. If you want to keep all covariates, leave all covariates checked.

**Subject**: Uncheck any subjects whose data you do not want to keep in the dataset to be saved. This is potentially useful if a certain subset of subjects corresponds to a treatment or variable that you are not interested in analyzing.

**Description box**: Enter (optional) annotations for the dataset you want to save. For example, you may wish to make a small note of the transformations made to the original dataset.

**Save button**: Saves [filename].RData to your current working directory

If a file by the same name exists, clicking "Save" will over-write it. To load a saved dataset, click on the Summary tab, and in the sidebar, click on "Load a different data set."

## The Analyze Tab

This tab is used to fit a statistical model to normalized gene expression data in order to detect genes with either different expression levels or different variances in two groups. In this tab the user can choose the differential factor, the levels of that factor which are to be compared, and the statistical model. It is also possible to include control variables which are assumed to have a significant effect on gene expression levels.

**Select a differential factor**: To fit the statistical model to the data (possibly transformed and/or filtered via the Filter/Transform tab), one first has to decide which factor is "differential", in the sense that there could be features which either have different variances and/or different means, when compared with the baseline level of this differential factor. Only a categorical variable may be selected as the differential factor.

**Select baseline level**: Specifies which level of the differential factor serves as the baseline for pairwise comparisons. It is possible to combine multiple levels as the baseline by selecting multiple levels from the drop-down list.

**Select treatment level**: Specifies which level of the differential factor serves as the 'treatment' for pairwise comparisons. It is possible to combine multiple levels as the treatment by selecting multiple levels from the drop-down list.

Being able to select multiple levels as baseline or treatment provides a convenient way to test different contrasts.

Once a differential factor is selected and the baseline and treatment groups are defined, the main panel will show two histograms - the top one depicts the distribution of the differential expression between the two groups, and the bottom one shows the logarithm of the ratio between the variances in the two groups.

**Method**: DVX allows to fit three different models, all of which are based on a mixture model in which most of the genes are assumed to be non-differential, and the distribution of the appropriate test statistics for non-differential genes can be assumed to be normal with mean 0.

To be more specific, let $M_{t,g}$ be the mean of the $g^{th}$ feature in the treatment group, let $M_{0,g}$ be the mean of the $g^{th}$ feature in the baseline group, and denote $d_g = M_{t,g} - M_{0,g}$. All three methods assume that $d_g \sim N(0, \sigma_g^2)$ for any gene which is not differentially expressed.

The three methods differ in how $d_g$ are modeled for differentially expressed ('non-null') genes. In the **L2N** method, $d_g$ of the differentially expressed genes are assumed to follow a mixture of two log-normal distributions, such that $d_g \sim LN(\mu_1, \tau_1^2)$ if $d_g > 0$, and $-d_g \sim LN(\mu_2, \tau_2^2)$ if $d_g < 0$ (Bar and Schifano (2018)). The **N3** method also consists of a three-component mixture model, except that the two non-null components are normally distributed (Bar, Booth, and Wells 2014). That is, $d_g \sim N(\mu_1, \tau_1^2)$ if $d_g > 0$, and $d_g \sim N(\mu_2, \tau_2^2)$ if $d_g < 0$, where $\mu_1 > 0$ and $\mu_2 < 0$. The **limma** method (G. K. Smyth 2004) is a two-component mixture model in which the non-null component is also normally distributed with mean 0, but with variance $v_0\sigma_g^2$ where $v_0 > 1$.

Note that the L2N and N3 methods also allow to test for differential variation between the treatment and baseline groups. let $V_{t,g}$ be the variance of the $g^{th}$ feature in the treatment group, and let $V_{0,g}$ be the variance of the $g^{th}$ feature in the baseline group. Denote $v_g = \log(V_{t,g}/V_{0,g})$. Applying a bias correction transformation (Bar, Booth, and Wells 2014), we can assume that $v_g$ follow a normal distribution with mean 0 for all the genes which have the same variance in both groups. Differentially dispersed genes are assumed to follow either the L2N or the N3 model, depending on the user's selection.

For simplicity, we previously defined $d_g$ as the difference between the means in the two groups, but in practice, it is more generally defined as the difference conditional on some predictors. Furthermore, if one of L2N or N3 is used, then $d_g$ is standardized by dividing the conditional difference between the groups by the gene-specific estimate of the posterior standard deviation of the difference. In the fitted-distribution plot for limma, $d_g$ is denoted by dE (differential expression), whereas if L2N or N3 are used, we denote the standardized differential expression in the plot by dEv, to highlight the fact that the differential expression is scaled by the gene-specific standard deviation.

Which of the three models to choose depends on the properties of the data and the user's preference. It is possible to fit all three models, and use the plots shown in the Results tab to determine which method may provide the best fit. Root MSE (rMSE) values may also be

compared across methods, and are provided for the fitted model as part of the title of the histograms in the Results tab.

**Trim percentile**: this is an optional parameter which determines whether the diagnostic plot in the Result tab (after fitting the selected model) will show the entire range of values of the test statistics $d_g$ and $v_g$ (trim=0) or be trimmed to drop the extreme percentiles on both sides. The trimmed view may be preferred if the tails are too long and one wants to see how well the selected mixture model fits the data where the majority of the data is concentrated.

**Predictors**: it is possible to include control variables in the model, in addition to the differential factor. Control variables are used to obtain better estimates for the mean expression levels in the two groups of interest. For all three methods, the adjusted mean expression levels are obtained by using limma's *lmFit* function. These predictors are especially useful if there is reason to believe that there may be a 'batch effect', and by including these predictors we may control for the undesired effect.

The control variables are used only to adjust the group means in the test for differential expression, and not for differential variation.

When all the model parameters are set, click on the Run button. While the fitting algorithm is running, a message will appear in the bottom-right corner of the screen. When the selected fitting algorithm converges, the Results tab will automatically be shown.

## The Results Tab

This tab is used to show the results of all previously executed analyses. By default, the most recent analysis is shown, but it is easy to switch to a previous analysis by selecting from the "**Select an analysis**" drop-down menu in the side-bar.

Also in the side-bar, the model specification is shown. This includes the name of the dataset, the differential variable, the baseline and treatment groups, the selected model, and whether any control predictors were included. Below the model specification, the user can select the following:

**Results to show**: this determines what is shown in the main panel. The options are:

- **Histograms**: show the histograms of the test statistics for $d_g$ (the differential expression) and $v_g$ (differential variation), along with the fitted distribution, per the selected model. The red curve depicts the null distribution, the green curve(s) depict the non-null components, and the dashed blue curve shows the overall fit of the mixture model. The title of each plot includes the root mean-squared error for the fitted model, to assess the goodness of fit. Note that since limma is not fitting a mixture model for differential variation, when the limma model is selected only the differential expression plot is shown.
- **Genes**: **Variance** (only available for L2N and N3): choosing this option will show a table of all the genes and their differential variation statistics, log(V1/V0), the p-

value (based on the null distribution), the False Discovery Rate (Benjamini and Hochberg 1995), the q-value (Storey 2002), and the (Bayesian) posterior probability that a gene has a significantly greater variance in the treatment (control) group, denoted by b2g (b1g). It is possible to change the number genes shown on each page, and to sort by any column in the table. The search box also allows to find specific genes of interest. Above the table there is a button labeled "save genes", which allows to export the entire list to a csv file, which can be viewed and edited with Excel.

- **Genes: Mean**: choosing this option will show a table of all the genes and their differential expression statistics, dE or dEv, the p-value (based on the null distribution), the False Discovery Rate (Benjamini-Hochberg), the q-value, and the (Bayesian) posterior probability that a gene has a significantly more (less) expressed in the treatment (control) group, denoted by b1g (b2g). It is possible to show more genes per page, and sort by any column in the table. The search box also allows to find specific genes of interest. Above the table there is a button labeled "save genes", which allows to export the entire list to a csv file, which can be viewed and edited with Excel.
- **p-values**: this option shows the distribution of the p-values as a histogram (on the right) and as a scatterplot of -log10(p) (on the left). The scatterplot also shows reference lines at -log10(0.05), -log10(0.01), and -log10(0.001), as well as at -log10(0.05/G) where G is the total number of genes. The latter (shown as a dotted orange line) represents the Bonferonni threshold for determining significance, while accounting for multiple testing, with $\alpha = 0.05$. Note that the p-values in the plots are the raw values, and are not adjusted for multiple testing.

**Analysis Name**: the analyses are automatically named by the program (sequentially - "Analysis 1", "Analysis 2", etc.). However, it is possible to give a more descriptive name to each model by filling this box and clicking on the "Rename" button.

**Add an annotation**: this text box and the "Add an annotation" button may be used to add a short description to the analysis.

**Delete Analysis**: this button may be used to delete an analysis from the history.

**Analyses to include in report**: the results of each analysis may be included in a report (see next tab) by checking the corresponding check-boxes. The report will include the model specification, the histograms of the statistics and the fitted mixture models, and the p-values plots. The complete lists of genes have to be exported to a separate csv file, as described above.

### Notes

- The software does not create duplicate analyses - if the selected model specifications (differential factor, baseline/treatment levels, method, and predictors) are determined to be identical to a previously run analysis, the previous analysis will be shown.

- When a new dataset is loaded, or when the user closes the program, all analyses are purged.

## The Save Report Tab

This tab is used to download all the selected plots and analyses. The report, which also includes the content of the Summary tab, is saved as an RTF file (rich text format) and can be viewed and edited using Microsoft Word. Note that for selected analyses, the report includes the model specification, the histograms of the statistics and the fitted mixture models, and the p-values plots.

## Quitting the program

To quit the program, simply press the Quit checkbox in the sidebar. You will be prompted to confirm this action. If you clicked it by mistake, simply uncheck the "Quit" box. If you want to save your plots, analyses, or transformations, you must do that before quitting the program.

Note that the Quit action stops the Shiny server, and in some browsers it closes the tab where the DVX interface was disaplyed, but in some browsers the browser's tab has to be closed manually.

# Case Studies

## The REST dataset

### A brief description of the data - Geo Data Set 5204

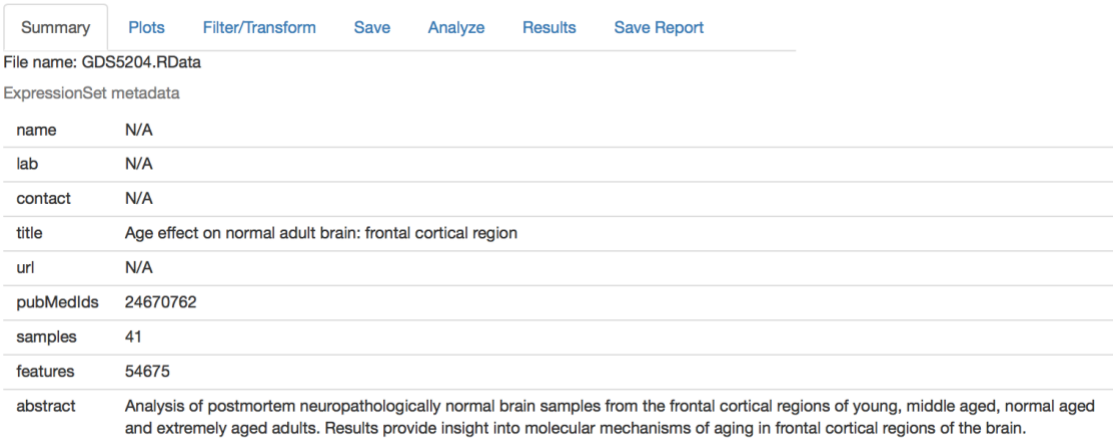
Understanding the mechanisms that preserve normal neuronal functionality is very important for treating Alzheimer's disease (AD) patients. REST/NRSF (repressor element 1-silencing transcription/neuron-restrictive silencer factor) is known to regulate neuronal genes during embryonic development, and Lu et al. (2014) showed that it is "induced in the aging human brain and regulates a network of genes that mediate cell death, stress resistance and AD pathology." Lu et al. (2014) observed that REST is lost from the nucleus of cells among AD and mild cognitive impairment (MCI) patients, which leads to dysregulation of this gene network.

Gene expression levels were obtained from 41 people, in four groups: young (<40yr) (n=12), middle aged (40-70yr) (n=9), normal aged (70-94yr) (n=16), and extremely aged (95-106yr) (n=4). There are 21 females and 20 males in this sample. The data has been deposited with GEO accession number GSE53890. To access the data on the NCBI web site, click here. The Lu et al. (2014) paper is available here.

We use DVX to perform differential expression and differential variation analysis between age groups.

## Descriptive statistics

To start the analysis, click on the Browse button, and select the GDS5204.RData dataset. The right panel in your web browser will show a summary of the data set. The top part shows the ExpressionSet metadata, and in this case it looks like this:




The bottom part of the panel shows gene expression and phenotype summary statistics. For the gene expression data, the summary includes the minimum and maximum, the first, second, and third quartiles, the mean, and the total number of missing values in the gene expression matrix. These statistics can indicate whether the expression data has been transformed. The plots in subsequent sections may be used to determine if further transformation or filtering is needed, but this table should give some guidance for any additional data processing steps. For example, if a log-transformation is needed, knowing the minimum value will help prevent applying the logarithm function to negative values. Also, if we later want to remove any genes with low-abundance across most or all samples, it is useful to know the range of expression values. See more about this in the description of the Filter/Transform tab.

Note that the statistical analysis is applied only to complete cases. Thus, any gene (a row in the gene expression matrix) which has at least one missing value, is removed. If one wishes to impute missing values, it must be done prior to loading the ExpressionSet data to DVX.

The summary of phenotype data is provided at the bottom of the page. For variables of type ‘factor’ a list of the levels and the sample size for each level are given. If the phenotype data contains numeric variables, their summary is provided in terms of the minimum, maximum, quartiles, mean, and number of missing values.

DVX performs differential analysis between two groups, so it is necessary to select a categorical variable (which we will call the treatment factor) with at least two levels. Other variables, categorical or continuous, may be included in the model if they are believed to be associated with expression levels in general (but their effect on expression is not different

across treatment groups). In this case “age” is our treatment factor, and we may want to include gender as an additional explanatory variable in the statistical model.

Gene expression statistics

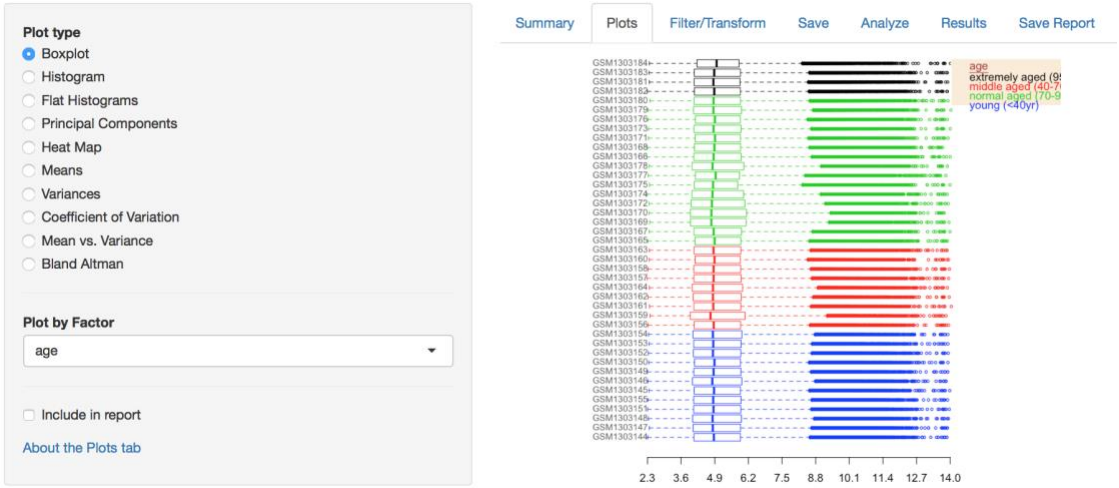
| min | Q1 | Median | Q3 | max | Mean | Missing |
|---|---|---|---|---|---|---|
| 2.27 | 4.08 | 4.85 | 5.92 | 14.04 | 5.15 | 0 |

Phenotype data (factors)

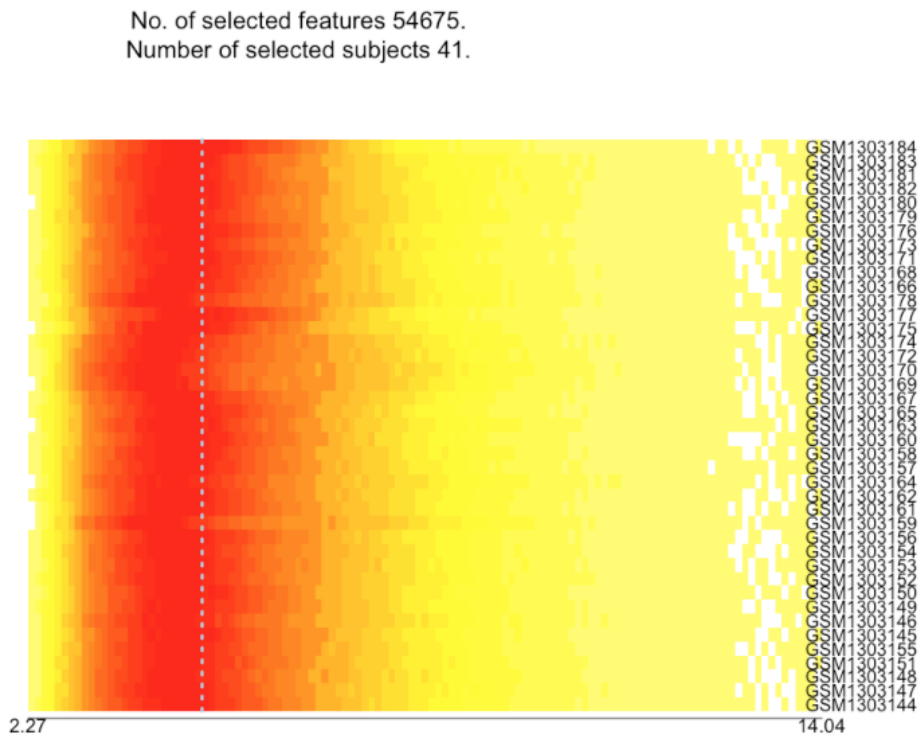
| | |
|---|---|
| age | extremely aged (95-106yr) (4), middle aged (40-70yr) (9), normal aged (70-94yr) (16), young (<40yr) (12) |
| gender | female (21), male (20) |

## Plots

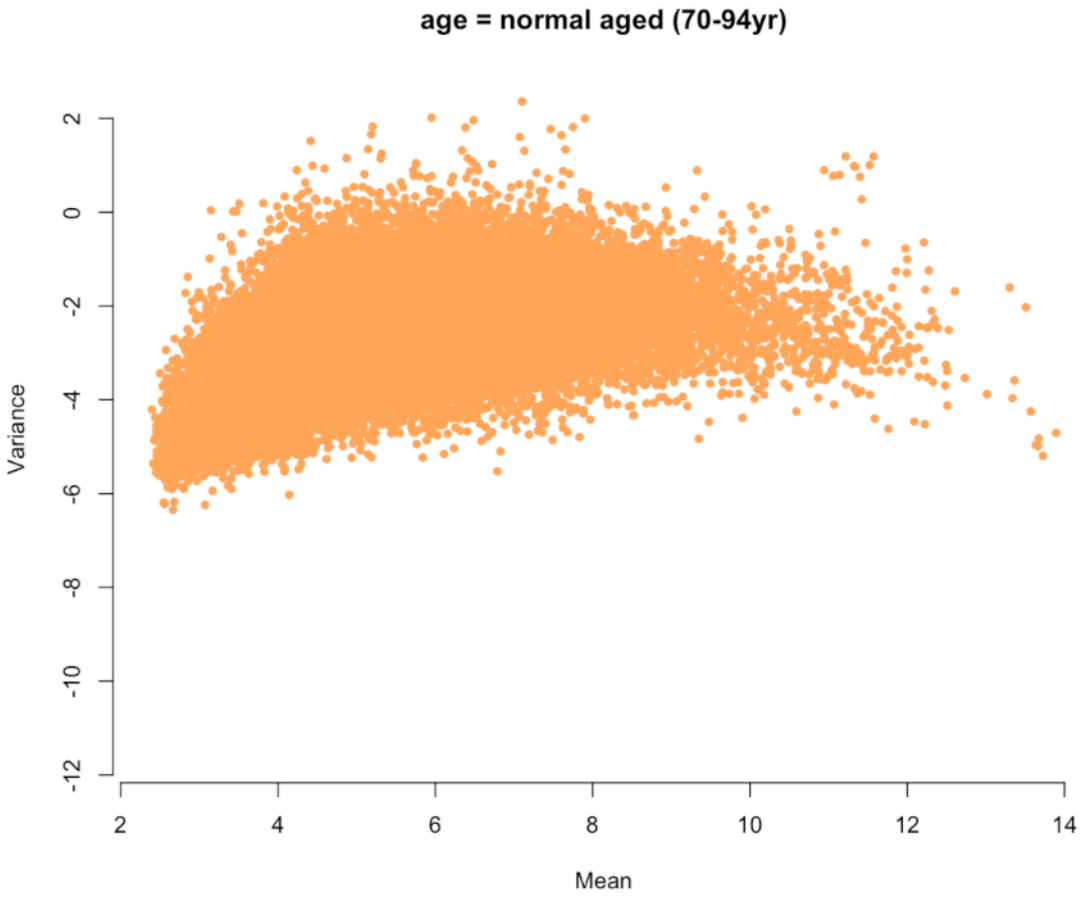
Next, it is a good idea to look at the different graphical representations of the data by clicking the Plots tab. The first option is Boxplot, which shows horizontal boxplots that depict the distribution of the expression levels for each sample. The samples are color-coded so that each color corresponds to a level of the user-selected factor. In the following plot, we selected ‘age’ from the drop-down menu. From this plot it is possible to see if the data needs to be transformed. The data may be somewhat skewed, since whiskers on the right are longer, and the only outliers appear to be on the right.




The ‘flat histogram’ plot serves a similar purpose, but it provides a more detailed view of the distributions of gene expression across samples. The dark red rectangles represent high-density regions - expression values which appear with high probability. The bright yellow rectangles correspond to expression values which have been observed for a small number of genes. The dashed vertical line represents the overall median. Like the boxplots, the flat-histogram shows if some samples have distinctly different distributions than others. In general, it is recommended to eliminate any subject-specific effects. One common way to do that, is to equalize the medians across all samples, which can be done in the Filter/Transform tab. The perceived skewness in this plot may be corrected by a log transformation, or by filtering low-abundance genes, as we will see in the next section.




In addition to the boxplots and the flat histogram, we will also use the “Mean vs. Variance” plot to check whether a data transformation is needed. The following plot shows the log-variance of genes versus the mean expression for the group “normal aged (70-94)”. There seems to be an upward trend, which suggests that the mean and variance are not independent.




## Filtering and/or transforming the data

The statistical models used by DVX to test for differential expression and/or variation rely on the assumption that the expression data has been normalized. Since we have noticed skewness in the flat histogram and a trend in the Mean-Variance plot, we may conclude that the assumption is not valid. One possible transformation in this case may be to take the logarithm of the expression data. However, from the documentation of the dataset we know that the expression values were already log-transformed. This can also be inferred from the information in the Summary tab - the values in the "Gene expression statistics" table are typical for log-transformed expression data.

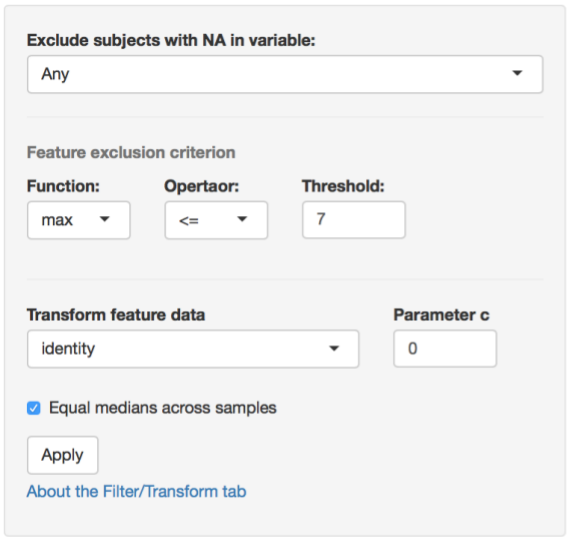
The pattern that we observed in the flat histogram and mean-variance plots can also be explained by a large number of low-abundance genes. Eliminating such genes may be desired because they are indistinguishable from 'background noise'.

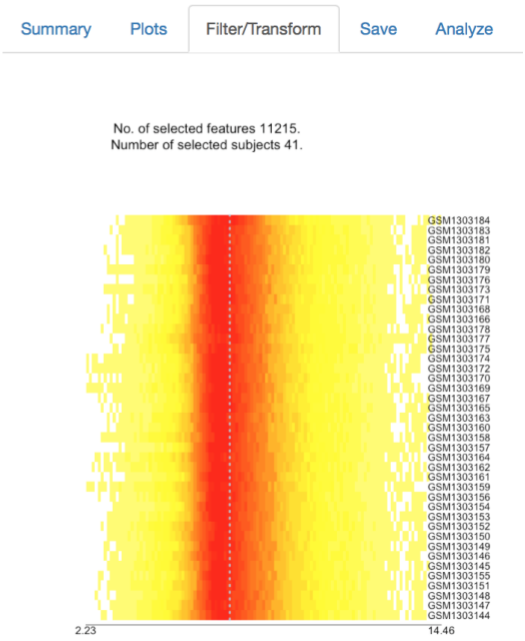
To transform the data, click on the "Filter/Transform" tab. It will show the flat histogram in the main panel, and the filtering and transformation functions will appear in the sidebar. Suppose that we choose to define a gene to be 'low-abundance' if its overall log-expression value is less than or equal to 7. We can use the drop-down menu to choose whether the mean (across all subjects) is less than or equal to 7, but we can also use the median, or the maximum value, as we did for the next screenshot.

If we also wanted to log-transform the data, we would click on the drop-down menu "Transform feature data" and choose "log2(c+x)". Since we saw that the minimum expression level is 2.27, we would leave c at its default value (0).

We recommend checking the "Equal medians across samples" box, to remove subject-specific effects. Click the Apply button to finish the transformation/filtering step.

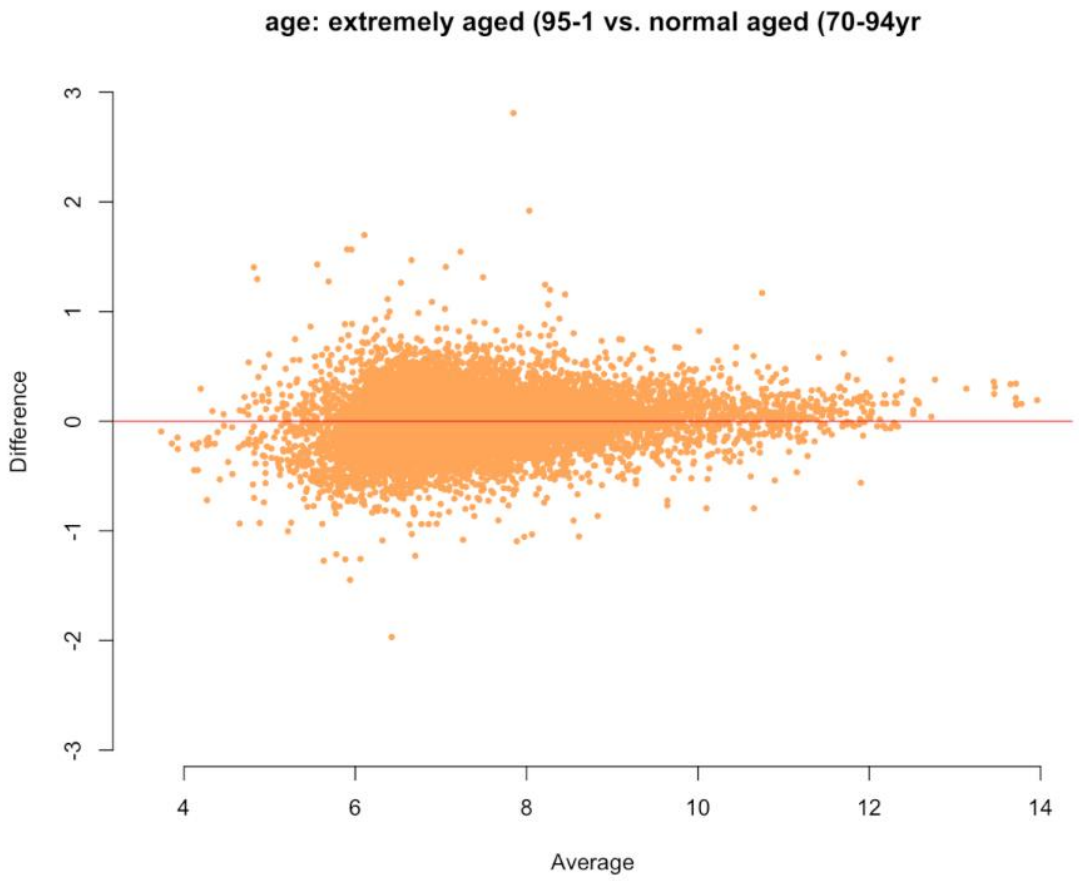
It is a good idea to save the filtered dataset, so we may skip the filtering step when we use the reduced data set in the future. To do that, click on the Save tab, and choose a file name (e.g. GDS5204filtered.RData).




We may now go back to the Plots tab and check whether the transformation yielded the desired results. For example, choose the "Bland Altman" plot, which shows the difference in expression levels between two levels of a factor versus the average expression levels. In principle, the plot should not show a relationship between the two dimensions. For example, if the differences increase (or decrease) as the average increases, it may indicate a deviation from the normality assumption. Regions with non-zero mean difference are also problematic, and may suggest that there might be a lurking confounding factor (or factors).

For example, in the following Bland Altman plot we used the extremely aged and the normal aged groups. There is nothing unusual about this plot, or for other pairs of age groups.



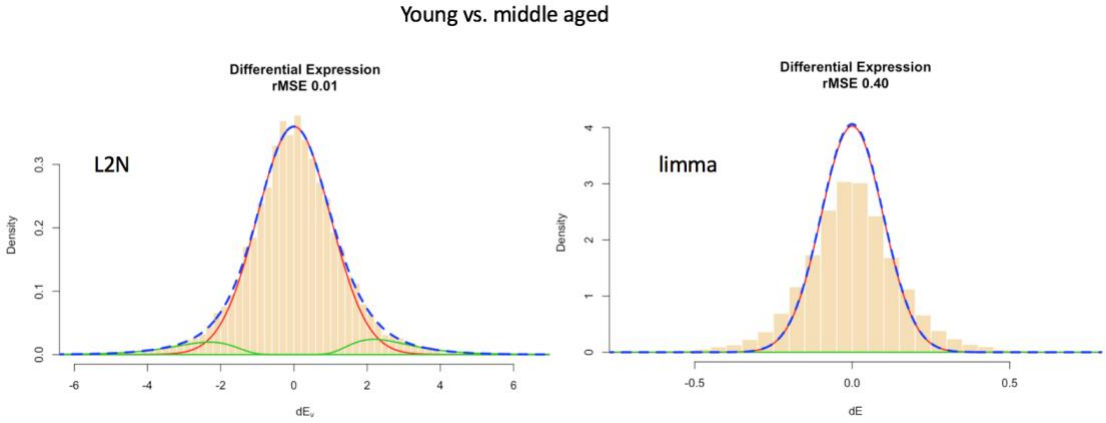

## Differential analysis

Next, we move to the Analyze tab and perform the statistical analysis. We want to test which genes are differentially expressed between age groups. In principle, we may combine multiple levels to be the baseline and/or the treatment, but in this case we will focus on pairwise comparisons. We may also control for other factors or covariates, by adding them in the Predictors text box. The only other variable in this dataset is gender, and we will add it as a predictor.

Since the change in neuronal condition is known to deteriorate gradually over time for adults, we will perform three comparisons: young vs. middle aged, middle aged vs. normal aged, and normal aged vs. extremely aged. We can run the differential analysis using all three available methods, namely, L2N, N3 (Bar, Booth, and Wells 2014), and limma (M. E. Ritchie et al. 2015). The fitted model is presented graphically in the Results tab, once the fitting algorithm has converged.

The following plots show the fitted distribution for the Young vs. Middle aged comparison for the L2N model (left) and limma (right). The baseline group was set as “young” and treatment group was set as “middle aged”. The red curve represents the distribution of the ‘null’ (non differentially expressed) genes, and the green curve(s) show the distribution of the non-null genes, per the selected model. Note that limma, by default, uses Pr(non-null) = 0.01. The dashed blue line is the fitted mixture distribution. These plots also show the

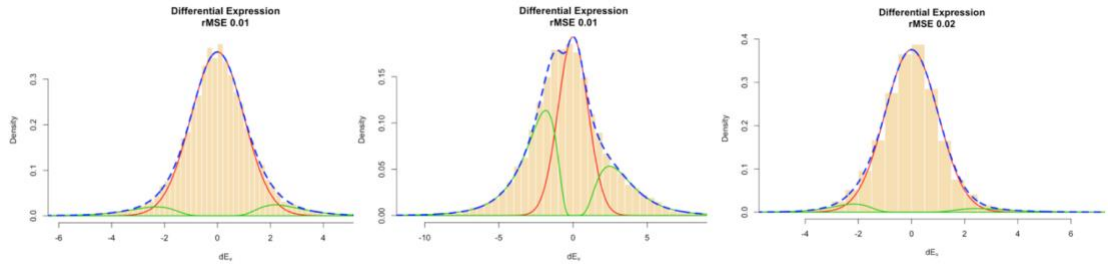
estimated goodness of fit, in terms of the root mean squared error (rMSE). In this case, the L2N model yields a better fit than limma (0.01 vs. 0.4).




Note that the scales on the x-axis are different for limma and L2N and N3. For limma, dE is the estimated contrast between the two groups, accounting for predictors. If no predictors are included in the model, dE is just the difference between the mean expression level in the treatment group and that in the control group. In N3 and L2N, the x-axis is labeled dEv which is the estimated standardized contrast between the two groups, accounting for predictors. The standardized contrasts are obtained by dividing dE by the estimated gene-specific standard deviations

With this dataset we get better fit with the L2N model, and so in the remainder of this section we present results obtained by using the L2N model. The following three plots show the fitted distributions for each of the three comparisons: young vs. middle aged (left), middle aged vs. normal aged (middle), and normal aged vs. extremely aged (right).

It is clear from the plots that in the comparisons young vs. middle aged and normal aged vs. extremely aged, the vast majority of genes are not differentially expressed, whereas in the comparison between middle aged vs. normal aged many genes are estimated to be differentially expressed.




Note that L2N and N3 also test for differential variation, and similar plots (not shown) are generated for the statistics $\log(V_{1g}/V_{0g})$ where $V_{1g}$ is the variance for gene g in the treatment group, and $V_{0g}$ is the variance for gene g in the baseline group. With this dataset and with $q < 0.01$, no genes are differentially dispersed in the Young vs. middle aged and in the middle aged vs. normal aged comparisons, and two genes have a significantly higher

variance in the extremely aged group when compared with the normal aged (205737_at and 207614_s_at).

In addition to the goodness of fit plot, the sidebar of Results tab offers two additional options to view the results of an analysis. You can view the results as a table, which includes the gene ID, the test statistic, the p-value (unadjusted), the Benjamini-Hochberg adjusted p-value (Benjamini and Hochberg 1995), and the q-value (Storey 2002). For L2N and N3 the table also contains the posterior probabilities of genes being in the two non-null components. As example is provided below for a limma analysis of the complete data set.

The table can be sorted by clicking on a column name. The table is also searchable, which can be useful if one is interested in the outcome of specific genes. The list can be exported in its entirety to a comma separate file, by clicking on the "save genes" button at the top of the panel. The exported list contains any available feature data, such as "Gene title", "Gene symbol", "GO function", etc.

The following screenshot shows the top 10 differentially expressed genes in the "young vs. middle aged" comparison using the L2N method, sorted by the q-values. Since "young" was set as the baseline group, the genes with positive (negative) dE are overexpressed (underexpressed) in the middle-aged group as compared to the young-aged group.

**Differential expression analysis**

young vs. middle aged - L2N save genes

Show 10 entries Search:

| | dE | p.val | bh | q.val | b1g | b2g |
|---|---|---|---|---|---|---|
| 1554633_a_at | -5.3473 | 0 | 0 | 0 | 0 | 0.9998 |
| 201039_s_at | 6.0305 | 0 | 0 | 0 | 1 | 0 |
| 201843_s_at | 6.658 | 0 | 0 | 0 | 1 | 0 |
| 202543_s_at | -6.1368 | 0 | 0 | 0 | 0 | 1 |
| 202581_at | -5.9598 | 0 | 0 | 0 | 0 | 1 |
| 202917_s_at | 5.9464 | 0 | 0 | 0 | 1 | 0 |
| 203337_x_at | -8.6314 | 0 | 0 | 0 | 0 | 1 |
| 203973_s_at | 5.482 | 0 | 0 | 0 | 1 | 0 |
| 204326_x_at | 5.7224 | 0 | 0 | 0 | 1 | 0 |
| 206001_at | -5.8601 | 0 | 0 | 0 | 0 | 1 |

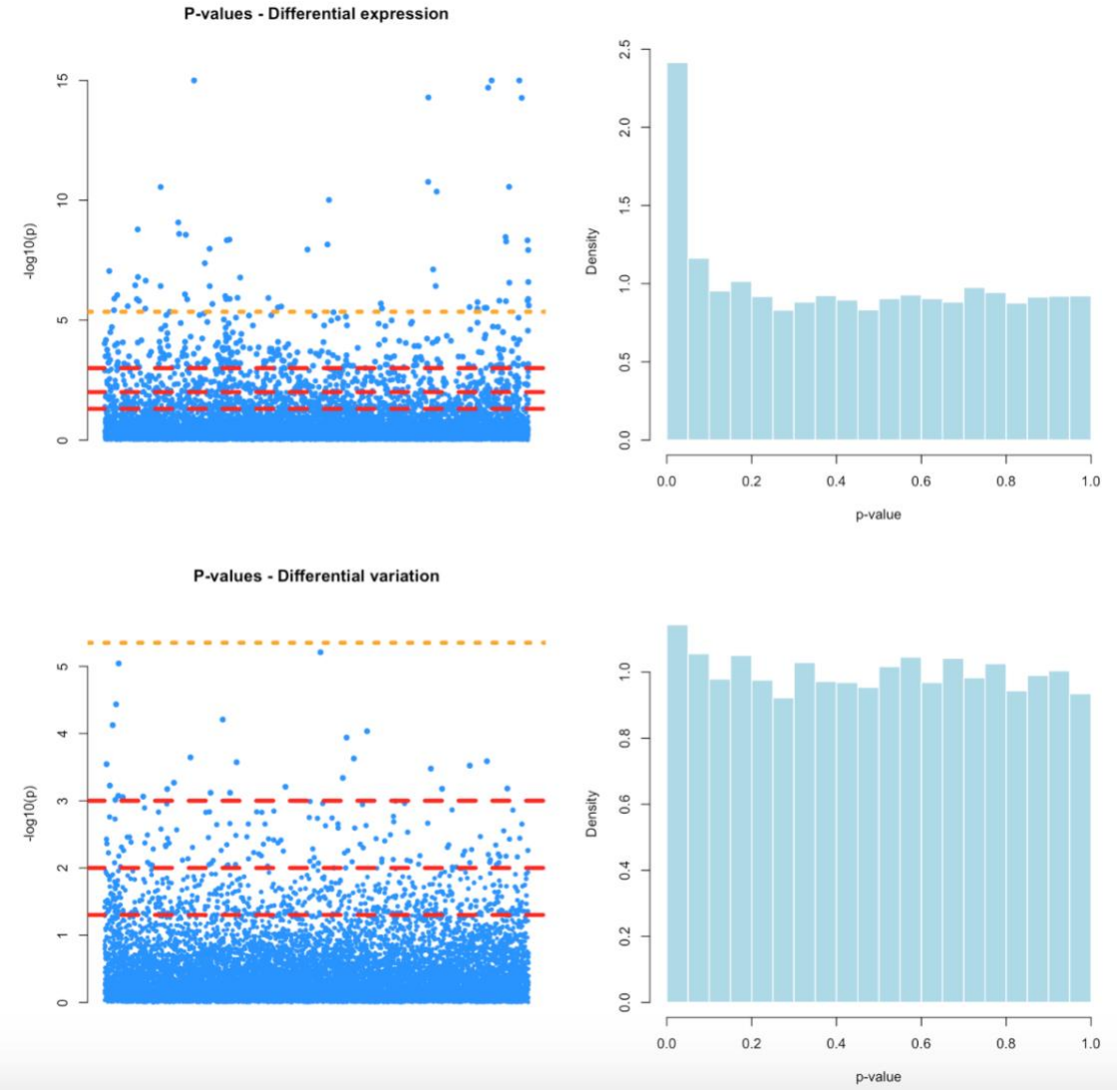
Showing 1 to 10 of 11,215 entries Previous 1 2 3 4 5 ... 1122 Next

Using a q-value threshold of 0.01, the L2N method finds 160 differentially expressed (DE genes) between middle aged and young, 2842 DE genes between middle aged and normal aged (middle), and 67 DE genes between normal aged vs. extremely aged (out of 11,215 genes that remained in the filtered dataset.)

The distribution of the p-values may also be of interest for diagnostic purposes. The following plots, generated from the "young vs. middle aged" comparison using the L2N method, show the scatterplot (left) and histogram (right) of p-values for the differential

expression analysis (top) and the differential variation analysis (bottom). The p-values are not adjusted for multiple testing. The dotted orange line represents the significance level for a Bonferroni correction for multiple testing (-log10(0.05/G) where G is the number of genes.)

The histogram for the differential analysis is approximately uniform for the larger p-values, which is the expected distribution under the null hypothesis (no genes with differential variation). The top scatterplot shows that there are numerous genes with p-value above the dashed orange line which means that they are sufficiently small to be declared differentially expressed, even if we used the conservative Bonferroni adjustment to the p-values (using the 0.05 level.)




## Saving Results

To save the selected results and plots, click on the Save Report tab and then click on the Save button. The information from the Summary tab will also be included in the report, which can be viewed and edited using Microsoft Word. A sample report for this data set is

provided here as a pdf file. The report is annotated and reorganized to improve the presentation (e.g., some plots were resized in order to fit side by side.) Detailed results from the differential analysis can be saved to a separate csv file, as described above.

## Autism and Copy Number Variation of human 16p11.2 - Geo DataSet 4430

### A brief description of the data

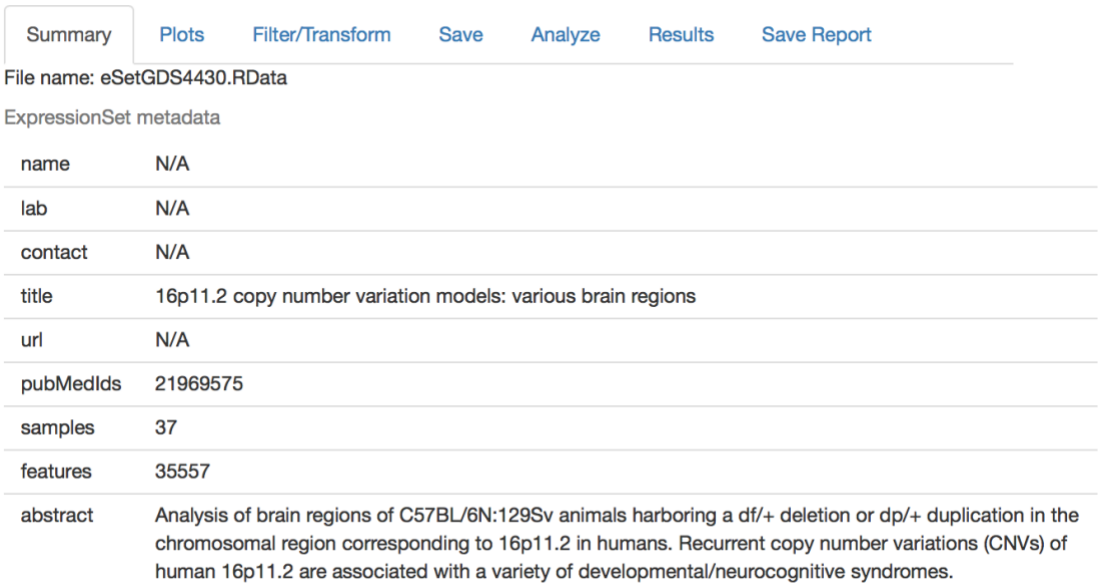
We use DVX to perform differential expression and differential variation analysis between three groups, defined by the copy number variations (CNV) in mouse chromosome 7 that are syntenic to human 16p11.2, genotypes known to be associated with multiple developmental/neurocognitive syndromes. For more information about the experiment, see Horev et al. (2011).

Samples were collected from four regions in the brain (the factor 'tissue'), from eight cloned mice (the factor 'individual'). The purpose of this case study is to identify differential genes when comparing different genotypes/variations. In particular, we are interested in comparing the '+/+ wild type' group (mice having 2 copies of the allele, n=15) with the 'df/+ deletion' group (1 copy, n=10), and with the 'dp/+ duplication' group (3 copies, n=12).

The data for this section may be obtained from
https://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS4430

### Descriptive statistics

To start the analysis, click on the Browse button, and select the eSetGDS4430.RData dataset. The right panel in your web browser will show a summary of the data set. The top part shows the ExpressionSet metadata, and in this case it looks like this:




The bottom part of the panel shows gene expression and phenotype summary statistics. For the gene expression data, the summary includes the minimum and maximum, the first, second, and third quartiles, the mean, and the total number of missing values in the gene expression matrix. These statistics usually indicate whether the expression data has been transformed. The plots in subsequent sections may be used to determine if further transformation or filtering is needed, but this table should give some guidance for any additional data processing steps. For example, if a log-transformation is needed, knowing the minimum value will help prevent applying the logarithm function to negative values. Also, if we later want to remove any genes with low-abundance across most or all samples, it is useful to know the range of expression values. See more about this in the description of the Filter/Transform tab.

Note that the statistical analysis is applied only to complete cases. Thus, any gene (a row in the gene expression matrix) which has at least one missing value, is removed. If one wishes to impute missing values, it must be done prior to loading the ExpressionSet data to DVX.

The summary of phenotype data is provided at the bottom of the page. For variables of type ‘factor’ a list of the levels and the sample size for each level are given. If the phenotype data contains numeric variables (e.g., age), their summary is provided in terms of the minimum, maximum, quartiles, mean, and number of missing values.

Note that DVX performs differential analysis between two groups, so it is necessary to select a categorical variable (which we will call the treatment factor) with at least two levels. Other variables, categorical or continuous, may be included in the model if they are believed to be associated with expression levels in general (but their effect on expression is not different across treatment groups). In this case “genotype/variation” is our treatment factor, and we may want to include tissue and/or individual as additional explanatory variables in the statistical model.

Gene expression statistics

| min | Q1 | Median | Q3 | max | Mean | Missing |
|---|---|---|---|---|---|---|
| 3.29 | 5.84 | 7.40 | 9.22 | 14.54 | 7.59 | 37 |

Phenotype data (factors)

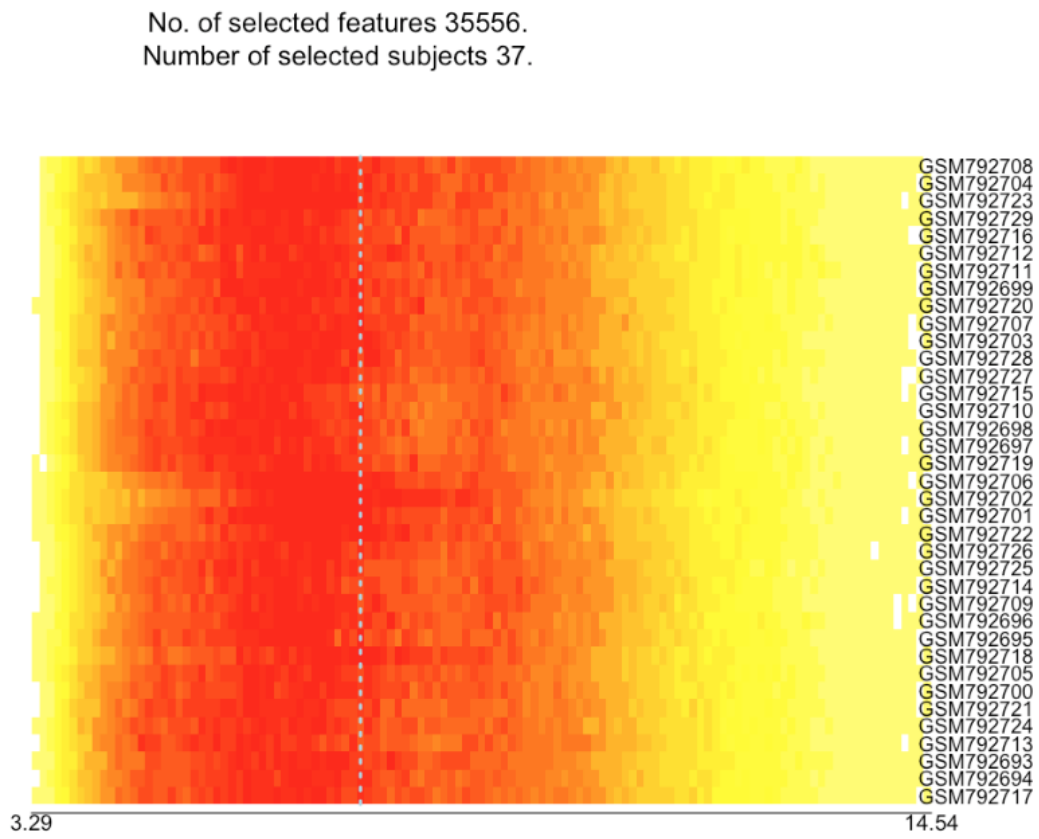
| | |
|---|---|
| tissue | brain stem (8), cerebellum (11), cortex (9), olfactory (9) |
| genotype/variation | +/+ wild type - 2 copies (15), df/+ deletion - 1 copy (10), dp/+ duplication - 3 copies (12) |
| individual | 101 (6), 102 (5), 121 (4), 62 (5), 63 (4), 88 (4), 89 (3), 90 (6) |

## Plots

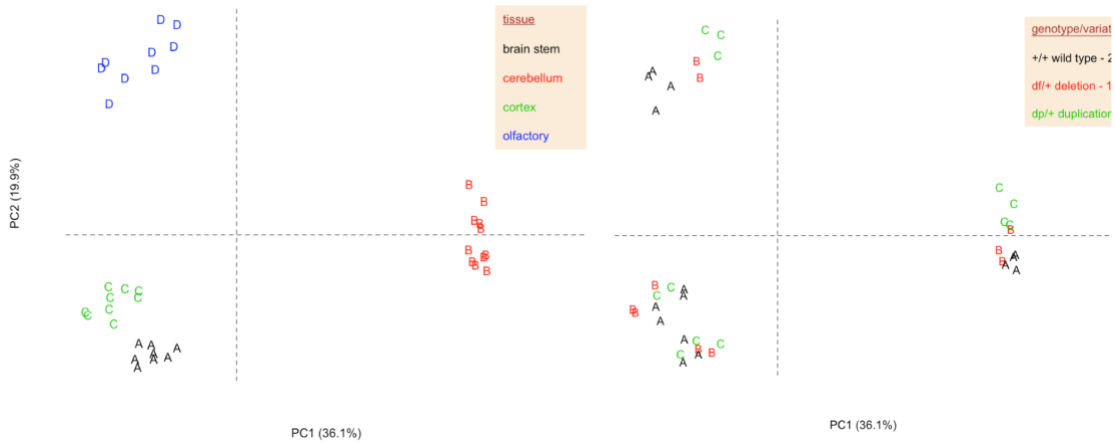
Next, it is a good idea to look at the different graphical representations of the data by clicking the Plots tab. The first option is Boxplot, which shows horizontal boxplots that depict the distribution of the expression levels for each sample. The samples are color-coded so that each color corresponds to a level of the user-selected factor. In the following plot, we selected 'genotype/variation' from the drop-down menu. From this plot it is possible to see if the data has to be transformed. In this example, the data may be somewhat skewed, since whiskers on the right are longer, and the only outliers appear to be on the right.


The 'flat histogram' plot serves a similar purpose, but it provides a more detailed view of the distributions of gene expression across samples. The dark red rectangles represent high-density regions - expression values which appear with high probability. The bright yellow rectangles correspond to expression values which have been observed for a small number of genes. The dashed vertical line represents the overall median. Like the boxplots, the flat-histogram shows if some samples have distinctly different distributions than others. In general, it is recommended to eliminate any subject-specific effects. One common way to do that, is to equalize the medians across all samples, which can be done in the Filter/Transform tab. The perceived skewness in this plot may be corrected by a log

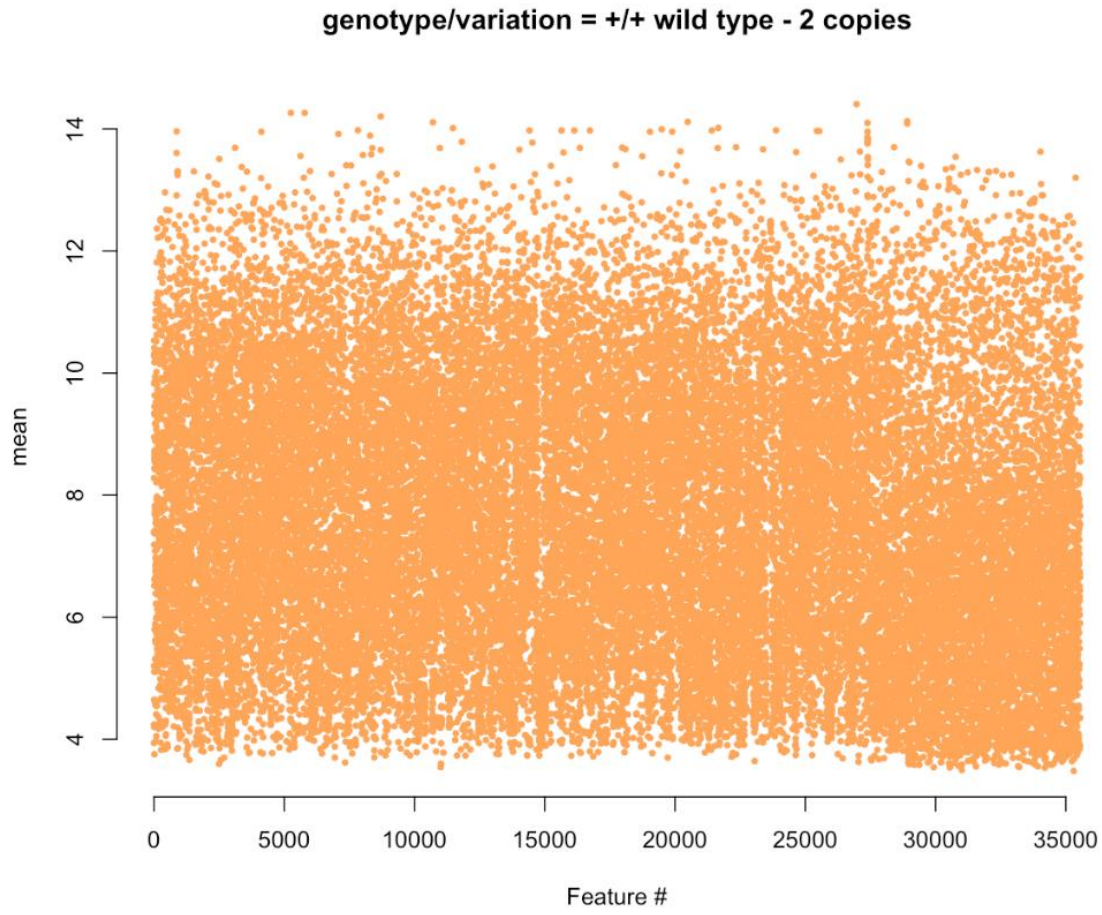
transformation, or by removing genes with low-expression levels in all subjects. We will discuss this further when we demonstrate the Means plot below.



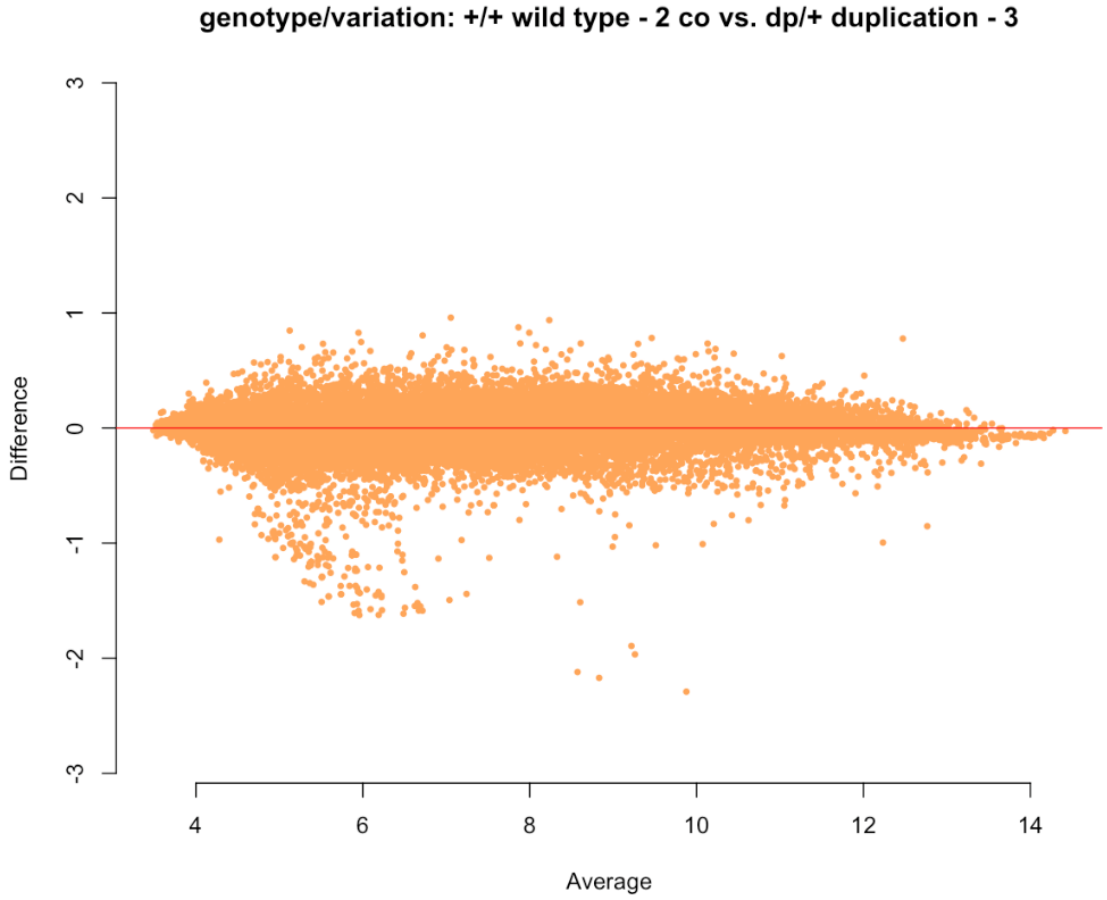

Next on the Plots tab is the Principal Components option, which shows the arrangement of the samples by the first two principal components. By changing the selected factor in the drop down menu, we can see whether the data are clustered in ways which should be considered when fitting a statistical model. For example, it is clear from the left plot below that the four tissue types are clustered separately, suggesting that the expression levels vary systematically across brain regions. When we change the selected factor to be the genotype, from the plot on the right we can see that the three genotypes are represented almost equally in each region. Since we are interested in the comparisons between genotypes, the PC plot suggests that if the expression levels are affected by the tissue type, this effect is likely to be cancelled out. The olfactory region (D) is the only one in which the three genotypes form distinct sub-clusters, so we may want to investigate this region separately. However, the distances within this cluster are small so for the purpose of this case study we will not pursue this direction.




DVX can be used to produce detailed plots of means, variances, and coefficients of variation by gene, for selected factors and levels. These plots can be used to determine if additional filtering or transformation is needed. The following example shows the mean expression across all samples in the wild-type group for each of the 35,556 genes. Among the ~7,000 genes on the right hand side there appear to be many with low mean expression levels. We may want to investigate this further, or use this information to determine a threshold for removing genes from the analysis.



The statistical models used here to test for differential expression and/or variation rely on the assumption that the expression data has been normalized. To check if there is evidence that suggests that the assumption is not valid, two types of plots may be used, namely the "Mean vs. Variance" plot, and the "Bland Altman" plot. The latter is demonstrated below. It shows the difference in expression levels between two levels of a factor versus the average expression levels. In principle, the plot should not show a relationship between the two dimensions. For example, if the differences increase (or decrease) as the average increases, it may indicate a deviation from the normality assumption. Regions with non-zero mean difference are also problematic, and may suggest that there might be a lurking confounding factor (or factors). While the plot below does not suggest a clear violation of the normality assumption or a potential lurking variable, we notice that many of the genes with large differences between the groups (in absolute value) also have a relatively small average. We may consider running the analysis with a subset in which we have excluded low-abundance genes. We explain how to do this in the next section.




## Filtering and/or transforming the data

The boxplots and the flat histograms showed some skewness in the data, but from the documentation of the data set we know that the expression values were already log-transformed. We also saw that the large differences between the genotypes occurred mostly for genes with overall low expression values. We may choose to perform the

differential analysis using only genes which are suffciently expressed across samples. We switch to the Filter/Transform tab and select Function=mean, Operator="<=", and Threshold=9. We check the "Equal medians across samples" box, and click the Apply button. The resulting dataset has 9,856 genes and the distribution of each sample appears to be more symmetric. The amount of variance explained by the first two components increases from 46% to 68.4% after the filtering. Also, within each cluster, the three genotypes are mixed well, so we expect that any effect the tissue has on expression levels will be cancelled out when we perform the differential analysis.

If we want to save the filtered dataset, we can click on the Save tab, and choose a file name (e.g. eSetGDS4430mean9.RData). This way, we may skip the filtering step if we want to return to the reduced data set in the future.

## Differential analysis

Next, we move to the Analyze tab and perform the statistical analysis. We select the differential factor to be "genotype/variation", the baseline level to be "+/+ wild type - 2 copies", and the treatment level to be "df/+ deletion - 1 copy".

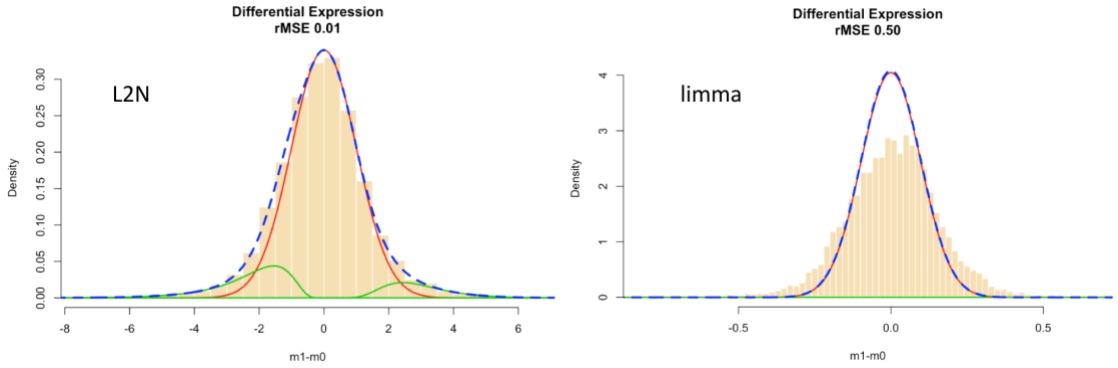
In principle, we may combine multiple levels to be the baseline and/or the treatment, but in this case it does not make sense, because the "1 copy" group is expected to be very different from the "3 copies" group.

We control for the individual effect by including this factor as a predictor. We do not include the tissue as a predictor, since we observed previously that its effect on expression levels is likely to cancel out in the differential analysis.

We run the differential analysis using all three available methods, namely, L2N, N3 (Bar, Booth, and Wells 2014), and limma (M. E. Ritchie et al. 2015). The fitted model is presented graphically in the Results tab, once the fitting algorithm has converged. The plots for the L2N and limma methods are shown below. The red curve represents the distribution of the 'null' (non differentially expressed) genes, and the green curve(s) show the distribution of the non-null genes, per the selected model. Note that limma, by default, uses Pr(non-null) = 0.01. The dashed blue line is the fitted mixture distribution.

These plots also show the estimated goodness of fit, in terms of the root mean squared error (rMSE). In this case, the L2N model yields a better fit than limma (0.01 vs. 0.5).

Note that L2N and N3 also test for differential variation, and similar plots (not shown) are generated for the statistics $\log(V_{1g}/V_{0g})$ where $V_{1g}$ is the variance for gene g in the treatment group, and $V_{0g}$ is the variance for gene g in the baseline group.




In addition to the goodness of fit plot, the sidebar of Results tab offers two additional options to view the results of an analysis. You can view the results as a table, which includes the gene ID, the test statistic, the p-value (unadjusted), the Benjamini-Hochberg adjusted p-value (Benjamini and Hochberg 1995), and the q-value (Storey 2002). For L2N and N3 the table also contains the posterior probabilities of genes being in the two non-null components. As example is provided below for a limma analysis of the complete data set.

The table can be sorted by clicking on a column name. The table is also searchable, which can be useful if one is interested in the outcome of specific genes. The list can be exported in its entirety to a comma separate file, by clicking on the “save genes” button at the top of the panel. The exported list contains any available feature data, such as “Gene title”, “Gene symbol”, “GO function”, etc.

In our case study, limma detects 22 differentially expressed genes with FDR < 0.05 (and 21 of these are underexpressed in the treatment group, “df/+ deletion - 1 copy”.) L2N and N3 detect 385 and 360 differentially expressed genes with FDR < 0.05, respectively.

L2N and N3 also perform differential variation analysis, and in this case L2N detects 6 genes with a Benjamini-Hochberg adjusted p-value less than 0.05, while N3 detects 7.

Summary | Plots | Filter/Transform | Save | Analyze | Results | Save Report

## Differential expression analysis

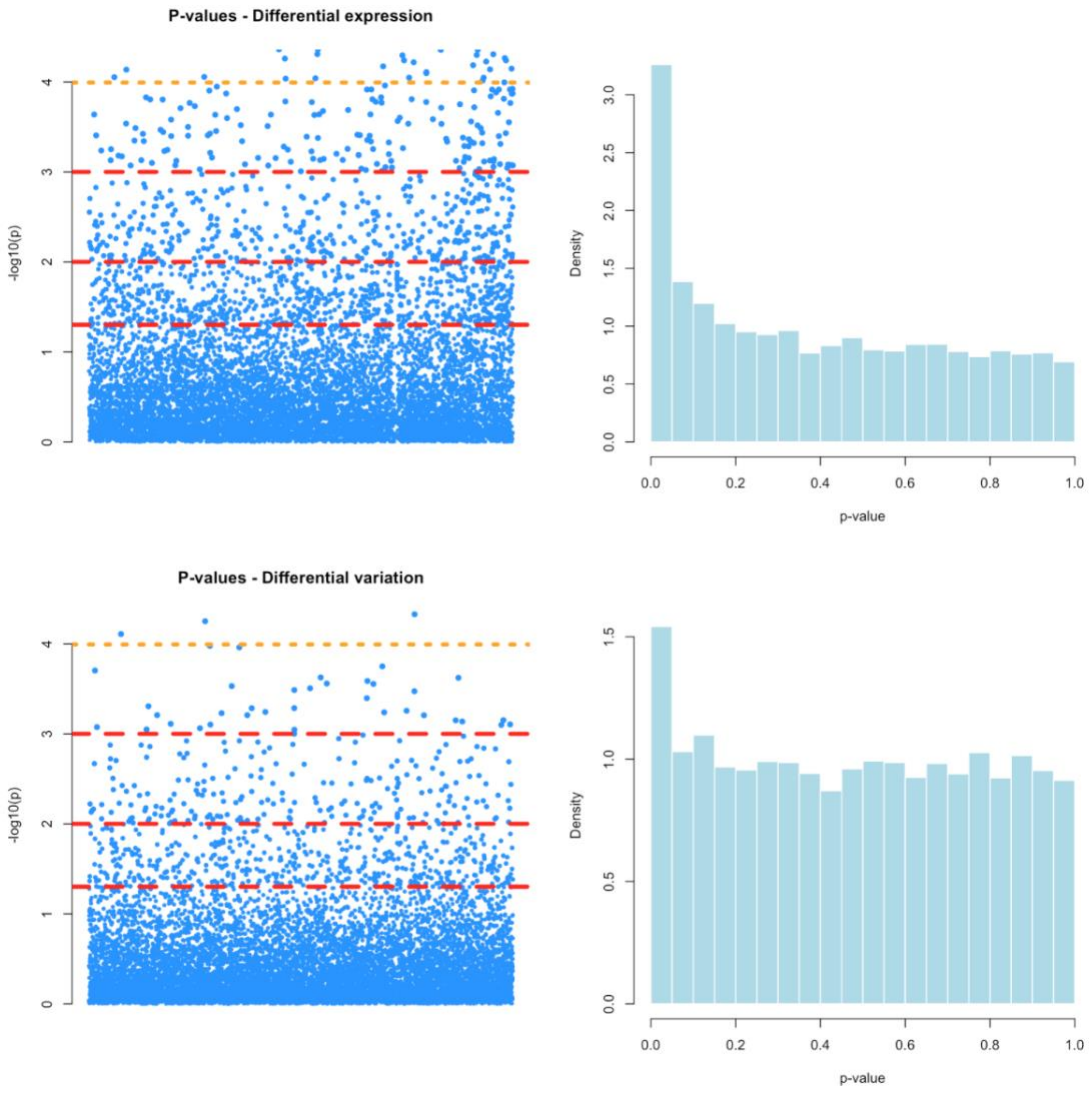
eset4430limma [save genes]

Show 10 entries Search:

| | m1-m0 | p.val | bh | q.val |
|---|---|---|---|---|
| 10344614 | 0.9203 | 0.3665 | 0.9489 | 0.9253 |
| 10344616 | -2.2947 | 0.0307 | 0.5821 | 0.5676 |
| 10344618 | -1.1089 | 0.2784 | 0.9126 | 0.8899 |
| 10344620 | 1.1824 | 0.2485 | 0.9016 | 0.8792 |
| 10344622 | 2.1933 | 0.0381 | 0.6204 | 0.605 |
| 10344624 | -2.1432 | 0.0423 | 0.6354 | 0.6196 |
| 10344633 | -1.8644 | 0.0744 | 0.6976 | 0.6802 |
| 10344637 | -0.5154 | 0.6109 | 0.9847 | 0.9602 |
| 10344653 | 0.1187 | 0.9065 | 0.9965 | 0.9717 |
| 10344658 | 2.9369 | 0.0072 | 0.3907 | 0.381 |

Showing 1 to 10 of 35,556 entries Previous 1 2 3 4 5 … 3556 Next

The distribution of the p-values may also be of interest for diagnostic purposes. The following plot shows the scatterplot (left) and histogram (right) of p-values for the differential expression analysis (top) and the differential variation analysis (bottom). The p-values are not adjusted for multiple testing. The dotted orange line represents the significance level for a Bonferroni correction for multiple testing (-log10(0.05/G) where G is the number of genes.)




To save the selected results and plots, click on the Save Report tab and then click on the Save button. The information from the Summary tab will also be included in the report, which can be viewed and edited using Microsoft Word. A sample report for this data set (without filtering, but with a transformation to achieve equal medians across samples) is provided here as a pdf file. The report is annotated and reorganized to improve the presentation (e.g., some plots were resized in order to fit side by side.)

## Data from other platforms

The three available fitting methods of DVX, i.e., limma, N3, and L2N, may also be used with count data (e.g., RNA-seq read counts) with proper data transformation. Popular transformations include counts per million (CPM) and log2-counts per million (log-CPM). See, e.g., C. W. Law et al. (2014) and C. W. Law et al. (2016). Note that these types of transformations need to be done outside of DVX, and the transformed data can be loaded into DVX as an ExpressionSet.

## Citations


Andrew J. Bass, John D. Storey with contributions from, Alan Dabney, and David Robinson. 2015. *Qvalue: Q-Value Estimation for False Discovery Rate Control*. http://github.com/jdstorey/qvalue.

Bar, Haim Y., and Elizabeth D. Schifano. 2018. "Differential Variation and Expression Analysis." University of Connecticut, Department of Statistics.

Bar, Haim Y., James G. Booth, and Martin T. Wells. 2014. "A Bivariate Model for Simultaneous Testing in Bioinformatics Data." *Journal of the American Statistical Association* 109 (506): 537–47.

Benjamini, Yoav, and Yosef Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society. Series B (Methodological)* 57 (1). Blackwell Publishing for the Royal Statistical Society: 289–300. doi:10.2307/2346101.

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. 2017. *Shiny: Web Application Framework for R*. https://CRAN.R-project.org/package=shiny.

Davis, Sean, and Paul Meltzer. 2007. "GEOquery: A Bridge Between the Gene Expression Omnibus (Geo) and Bioconductor." *Bioinformatics* 14: 1846–7.

Huber, W., Carey, V. J., Gentleman, R., Anders, et al. 2015. "Orchestrating High-Throughput Genomic Analysis with Bioconductor." *Nature Methods* 12 (2): 115–21. http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

Law, C. W., M. Alhamdoosh, S. Su, G. K. Smyth, and M.E. Ritchie. 2016. "RNA-seq analysisis easy as 1-2-3 with limma,Glimma and edgeR." *F1000Research* 5 (1408).

Law, C. W., Y. Chen, W. Shi, and G. K. Smyth. 2014. "voom: Precision weights unlock linear model analysis tools for RNA-seq read counts." *Genome Biol.* 15 (2): R29.

Lu, Tao, Liviu Aron, Joseph Zullo, Ying Pan, Haeyoung Kim, Yiwen Chen, Tun-Hsiang Yang, et al. 2014. "REST and Stress Resistance in Ageing and Alzheimer's Disease." *Nature* 507 (7493): 448–54. http://dx.doi.org/10.1038/nature13163.

R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Ritchie, Matthew E, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. 2015. "limma Powers Differential Expression Analyses for RNA-Sequencing and Microarray Studies." *Nucleic Acids Research* 43 (7): e47.

RStudio Team. 2015. *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc. http://www.rstudio.com/.

Schaffer, Michael E. 2013. *Rtf: Rich Text Format (Rtf) Output*. https://CRAN.R-project.org/package=rtf.

Smyth, Gordon K. 2004. “Linear Models for Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments.” *Statistical Applications in Genetics and Molecular Biology* 3 (1).

Storey, John D. 2002. “A Direct Approach to False Discovery Rates.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64 (3). Blackwell Publishers: 479–98. doi:10.1111/1467-9868.00346.

Xie, Yihui. 2016. *DT: A Wrapper of the Javascript Library 'Datatables'*. https://CRAN.R-project.org/package=DT.