# New Functions in packages DGE.Tools2, DGEobj and JRTutil packages:

Since Dec 2018

Domino Env: Genomics 2018-10 rev 3

DGE.Tools2 v0.9.59

DGEobj v0.9.27

JRTutil v0.9.17

For any of these functions, the ?functionname help is the best source of documentation. I try to put clear examples in the help. If you have trouble using any of these tools from reading the help, let me know and I will update the help accordingly.

## **DGE.Tools2 Package**

**Function lowIntFilter (0.9.56)**

Allows you to apply any combination of count, zFPKM or FPK filters to a DGEobj

Currently Xpress data can fail with zFPKM due to -Inf values. Doesn't happen with Omicsoft data. The issue is genes with a length shorter than the library size (~250nt).

Versions of this function prior to v0.9.57 are problematic with data derived from Xpress.

**Function obsPlot2 (0.9.56)**

Observation Plot is a grouped intensity boxplot and is intended for “gene(s) of the day” queries to produce intensity boxplots of 1-few dozen genes. The original obsPlot requires building a genes x samples dataframe of intensity data.

obsPlot2 is a tidy-ized version of the original obsPlot. The new version takes a tidy dataframe as input (see F. tidyIntensity to prepare the input from an intensity matrix)

**Function tidyIntensity (0.9.56)**

Use to prep data for obsPlot2. Takes intensity data from an intensity matrix in your favorite units (e.g. counts, log2cpm, log2TPM, etc.) with row (geneid) and colnames (SampleNames) and produces a tidy dataframe suitable for use with obsPlot2.

In our standard workflow, this comes out keyed on Ensembl gene IDs. It is easy to left\_join any additional columns you want to use as labels or graphic attributes (color, file, shape) (e.g. genesymbols and a replicate group column).

#create an Ensembl to GeneSymbol map

EnsToGeneSym <- getItem(DgeObj, "geneData") %>%

rownames\_to\_column (var=”EnsgID”) %>%

select(EnsgID, GeneSymbol=GeneName)

MyTidyData <- left\_join(MyTidyData, EnsToGeneSym)

**Function logRatioPlot (0.9.56)**

Similar to obsPlot2, but intended to plot logratio data with 95% confidence intervals (from topTable output) as barplots or points (with or without lines). For “gene of the day” queries, most biologists expect an intensity boxplot. However, the logRatio plot is often preferable to showing intensity boxplots because intensity boxplots show premodeling data and logratio data is post modeling and thus corrected for any nuisance factors (e.g. batch, demographics etc.)

**Function tidyContrasts (0.9.56)**

Takes a list of topTable dataframes and produces a tidy dataframe suitable for logRatioPlot

#Get a list of topTable dataframes from a DGEobj

contrastList <- getType(dgeObj, "topTable”)

tidyContrastData <- tidyContrasts(contrastList)

**Function MDS\_var\_explained**

MultiDimensional Scaling (as implemented in limma::plotMDS) yields results similar to PCA but uses an intensity distance-based metric that makes the axis dimensions more interpretable. The MDS\_var\_explained function takes an MDS object (i.e. the result object from limma::plotMDS which is also returned by functions DGE.Tools2::ggplotMDS) and presents 2 plots: variance for each dimension and cumulative variance explained.

Use this on the results of ggplotMDS (which is a ggplot wrapper for plotMDS).

**Function mapDGEobj (0.9.58)**

Extracts Parent-Child relationships from the workflow captured in a DGEobj and produce a class iGraph object suitable for a graph illustration of the workflow.

**Function QCplots (0.9.61)**

Enables plotting of a selection of QC data from Omicsoft QC data files or any file with QC Metrics in rows and Samples in columns.

**Function voomWorkflow**

Combines runEdgeRNorm and runVoom steps into one-step that includes calcNormFactors, voom, lmFit and eBayes. Always uses VoomWithQualityWeights and always uses eBayes robust=TRUE. Supports duplicateCorrelation.

## **DGEobj Package**

**Function adatToDGEobj**

Support for Somalogic .adat files added. Read Somalogic .adat files into a DGEobj datastructure. Uses F. initSOMAobj

**Function initSOMAobj**

Input 3 dataframes; intensities, design, proteinData.

Create a SOMAobj (class DGEobj)

**Function textToDGEobj**

Can read Omicsoft format files directly from the cloud if s3fs (mac/linux) or CloudBerry Drive (Windows) is used to mount s3 locally. Using this function in conjunction with cloud mounts obviates the need to copy files from S3 to local storage and should be the best practice going forward. I tested this on a PC with a mapped S3 drive using CloudBerry Drive.

**Function resetDGEobj**

Restore a filtered DGEobj to its original dimensions. Every DGEobj carries an unfiltered copy of the original counts + gene annotation + sample annotation. Calling resetDGEobj on a heavily filtered DGEobj returns the original unfiltered data that can be used for a new analysis, perhaps with different filtering criteria.

## **JRTutil Package**

## 

**Function getCloudFiles (0.9.23)**

Allows retrieval of multiple cloud files from an S3 bucket folder, optionally gzipping the saved files. Useful for retrieving Omicsoft output files from S3 to the local system. I prefer this method because is forces explicit documentation of the full path to the datafile.

However, I report a method below for reading directly from the cloud source that may render this function more or less obsolete.

**Function is.mounted (0.9.28)**

Test a mount (e.g. /stash) to confirm it is active (works on Win/Linux/Mac)

**Function tableStats (0.9.16)**

Calculate summary statistics (SD, mean, median, var, cv, max, min) on an input numeric matrix or dataframe.

Operates on columns or rows.

**Function detect\_os (0.9.28)**

Returns one of windows, linux, mac or unknown

**Function getStashPath (0.9.28)**

Return a valid stash path on Win/Mac/Linux. Use this to build platform-independent paths to data files in stash.

**Function inspect (0.9.28)**

Check a dataframe, matrix or vector for class, length, Dim, zero count, Inf, NegInf, NA and NANs

**Function missingDataHeatmap (0.9.33)**

Intended to show a visual of missing clinical data in a design file. It produces a two color heatmap

To show cells with or without data.

**Function checkDGEobj (0.9.33)**

Checks a DGEobj or RDS file containing a DGEobj and reports any missing data or attributes that would prevent this DGEobj from loading in GECO.

**BuildOmicsoftDGEobj (0.9.34)** (Gets data from Omicsoft S3 bucket)

If you use s3fs or Cloudberry Drive to mount the S3 bucket bmsrd-ngd-arrayserver to a local folder, you can retrieve data and build a DGEobj by simply referencing the project by name and specifying the mount point for the S3 bucket on your system. With a list of DGEobj names, you can build a selection of projects with a single lapply command.

**getRDSobjFromStash (0.9.34)** (Get data from RDS files on stash)

Retrieve an RDS object from a path on stash. The default path is the path of the DGEobj library. Thus, simply specifying the name of the file returns the corresponding DGEobj. The default path is “data/nonclin/DGEobj\_library” and contains the same set of DGEobjs that are loaded into the GECO shiny app (https://report.pri.bms.com/geco). With a list of DGEobj RDS filenames, you can import a selection of projects with a single lapply command. This will work for any .RDS file.

**Winsorize (0.9.34) (winsorize\_mean, winsorize\_sd and winsorize\_var)**

If you’re thinking about using a trimmed mean try the winsorize\_mean. The adaptive winsorize approach implemented here doesn’t trim anything if the data is normally distributed. If there are extreme outliers, the adaptive trimming is invoked.

## **Other useful tools and tips**

## [**DGE.Tools2PlotGallery.pdf**](https://biogit.pri.bms.com/thompj27/DGE.Tools2/blob/master/vignettes/DGE.ToolsPlotGallery.pdf) is a vignette that has recently been updated to show examples of each plot type supported by DGE.Tools2.

**Conflicted Package:**

Free yourself from worrying about package load order. Never again spend time debugging a problem because you thought you were executing a function from package X when the function by the same name in package Y was loaded last and takes precedence., After loading the conflicted library, any function conflicts (functions with same name in multiple packages) throw an error and list all the packages the function belongs to. This user is forced to use double colon references whenever two or more packages use the same function name. By forcing you to use double colon references, you are freed from worrying about the load order of your library commands.

**Platform-independent stash paths**

#getStashPath returns a platform-specific path and tests to make sure it’s actually mounted

stashPath <- JRTutil::getStashPath()

path <- "data/nonclin/someotherfolder"

file <- "myfilename.txt"

platformIndependentFilePath <- file.path(stashPath, path, file)

If paths are constructed as above, the code should work on all three platforms (linux, mac, windows).

**Read directly from an s3 bucket into R:**

Solutions involve establishing a local mount point to an s3 bucket.

See s3fs on Mac/Linux (https://github.com/s3fs-fuse/s3fs-fuse)

See CloudBerry Drive on Windows ($40)