Package 'DGCA'

October 12, 2022

Title Differential Gene Correlation Analysis

Description Performs differential correlation analysis on input matrices, with multiple conditions specified by a design matrix. Contains functions to filter, process, save, visualize, and interpret differential correlations of identifier-pairs across the entire identifier space, or with respect to a particular set of identifiers (e.g., one). Also contains several functions to perform differential correlation analysis on clusters (i.e., modules) or genes. Finally, it contains functions to generate empirical p-values for the hypothesis tests and adjust them for multiple comparisons. Although the package was built with gene expression data in mind, it is applicable to other types of genomics data as well, in addition to being potentially applicable to data from other fields entirely. It is described more fully in the manuscript introducing it, freely available at <doi:10.1186/s12918-016-0349-1>.

Version 1.0.2

Author Bin Zhang [aut],
Andrew McKenzie [aut, cre]

Maintainer Andrew McKenzie <amckenz@gmail.com>

Depends R (>= 3.2)

Imports WGCNA, matrixStats, methods

Suggests knitr, impute, gplots, fdrtool, testthat, ggplot2, plotrix, GOstats, HGNChelper, org.Hs.eg.db, AnnotationDbi, abind, MEGENA, Matrix, doMC, igraph, cowplot, stats, utils

VignetteBuilder knitr

RoxygenNote 5.0.1

License GPL-3

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-09 10:30:16 UTC

R topics documented:

adjustPVals *Adjusts a numeric vector of p-values.*

Description

Wraps around the R base implementation of p.adjust, as well as the methods used in the fdr tool package.

Usage

```
adjustPVals(pVals, adjust = "none", plotFdr = FALSE, verbose = FALSE)
```

Arguments

pVals	Numeric vector of p-values to adjust.
adjust	Allows for resulting p-values to be corrected for multiple hypothesis tests. Optional and some non-default choices require the "fdrtool" package. Default = "none", which means that no p-value adjustment is performed. Other options include methods in ?p.adjust (i.e., "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"), and methods in ?fdrtool (i.e., "fndr", "pct0", "locfdr").
plotFdr	Allows for plotting of fdrtool p-value adjustment result, if this is chosen. Requires fdrtool package. Default = FALSE.
verbose	If TRUE, the function prints out the general method used for multiple correction analysis. Default = FALSE.

Value

A numeric vector of p-values that have been adjusted according to the specified method.

Examples

```
pvals = runif(100, 0, 1)
adj_pvals_bh = adjustPVals(pvals, adjust = "BH")
adj_pvals_hommel = adjustPVals(pvals, adjust = "hommel")
```

ages_darmanis *Brain sample ages vector.*

Description

A vector specifying the ages of the brain cell types from the single-cell RNA-seq study. The total data set can be downloaded by following the links in the original paper.

Usage

```
ages_darmanis
```

Format

An object of class `numeric` of length 158.

References

Darmanis S, Sloan SA, Zhang Y, et al. A survey of human brain transcriptome diversity at the single cell level. Proc Natl Acad Sci USA. 2015;112(23):7285-90.

`bigEmpPVals` *Use speed-optimized sorting to calculate p-values observed and simulated null test statistic using a reference pool distribution.*

Description

A reimplementation of qvalue::empPvals designed to work faster and require less memory in the average case. Unlike qvalue::empPvals, *requires* the use of a reference pool distribution rather than having this as an option. Another difference of this function that the original is that it handles ties for test statistics equal to 0, to handle cases where test statistics are thresholded and may be zero more commonly than expected by chance (i.e., very rarely).

Usage

```
bigEmpPVals(stat, stat0, increasing = TRUE)
```

Arguments

`stat`	The vector of test statistics for which p-values should be returned.
`stat0`	The vector or matrix of simulated null test statistics (if matrix, will be coerced to a vector).
`increasing`	Logical indicating whether the test statistics because more extreme as they increase in value from 0. Negative numbers are not allowed as inputs (i.e., the test statistic must be monotonic).

Value

A vector of p-values adjusted for the null statistics calculated, to be used as an input to the qvalue function.

Author(s)

John Storey, Andrew McKenzie

References

Please see ?qvalue::empPVals for more; from which this function was adapted.

Examples

```
test_stat = rnorm(100, 1, 1)
test_stat0 = rnorm(1000, 0, 1)
emp_pvals = bigEmpPVals(test_stat, test_stat0)
```

---

corMats-class *An S4 class to store correlation matrices and associated info.*

Description

An S4 class to store correlation matrices and associated info.

Slots

`corMatList` List of correlation, number of sample, and correlation significance matrices in each condition.

`design` Design matrix inputted into the function.

`options` Character vector of options used in the function.

---

darmanis *Single-cell gene expression data from different brain cell types.*

Description

This data set has been filtered to include the genes that are in the 95th percentile or above in both oligodendrocytes and neurons. The total data set can be downloaded by following the links in the original paper.

Usage

```
darmanis
```

Format

An object of class `data.frame` with 572 rows and 158 columns.

References

Darmanis S, Sloan SA, Zhang Y, et al. A survey of human brain transcriptome diversity at the single cell level. Proc Natl Acad Sci USA. 2015;112(23):7285-90.

dCorAvg *Get average empirical differential correlations.*

Description

Finds the average (median or mean) of differential correlations for either one gene versus all others or for all gene pairs in the input matrix of differences in z-scores between conditions, with significance calculated via a comparison with the permutation samples.

Usage

```
dCorAvg(zDiff, zDiffPerm, dCorAvgType, oneSidedPVal = FALSE,
  secondMat = FALSE, dCorAvgMethod = "median")
```

Arguments

zDiff	Matrix containing the actual difference of z-scores for each gene-gene pair.
zDiffPerm	Matrix containing the differences of z-scores for each gene-gene pair in simulated data.
dCorAvgType	Character vector specifying the type of average differential correlation calculation that should be performed. Types = c("gene_average", "total_average"). gene_average calculates whether each genes' differential correlation with all others is more than expected via permutation samples (and subsequent empirical FDR adjustment, in the case of > 1 gene), while total_average calculates whether the total average differential correlation is higher than expected via permutation samples. If splitSet is specified, then only genes in the splitSet have their average gene differential correlation calculated if gene_average is chosen.
oneSidedPVal	If the dCorAvgType test is total_average, this option specifies whether a one-sided p-value should be reported, as opposed to a two-sided p-value. That is, if the average difference of z-scores is greater than zero, test whether the permutation average difference of z-scores are less than that average to get the p-value, and vice versa for the case that the average difference of z-scores is less than 0. Otherwise, test whether the absolute value of the average difference in z-scores is greater than the absolute values of the permutation average difference in z-scores. Default = FALSE.
secondMat	Logical indicator of whether there is a second matrix in the comparison or not.
dCorAvgMethod	Character vector specifying the method for calculating the "average" differential correlation calculation that should be used. Options = "median", "mean".

Value

A list containing the average difference(s) in z-score and the empirical p-value of that statistic calculated using the permutation samples.

dCorClass *Classify differential correlations.*

Description

Classifies identifiers (e.g., genes) into one of the different categories pairwise-differential correlation classes. These categories are one of the Cartesian products of "Up Correlation", "No Correlation", and "Down Correlation" in each of the conditions, as well as a category for "no significant differential correlation".

Usage

```
dCorClass(corsA, pvalsA, corsB, pvalsB, dCorPVals, sigThresh = 1,
  corSigThresh = 0.05, convertClasses = FALSE)
```

Arguments

corsA	Numeric vector of correlations between gene pairs in condition A.
pvalsA	Numeric vector of the significance of correlation calls between gene pairs in condition A.
corsB	Numeric vector of correlations between gene pairs in condition B.
pvalsB	Numeric vector of the significance of correlation calls between gene pairs in condition B.
dCorPVals	Numeric vector of the differential correlation p-value calls.
sigThresh	If classify = TRUE, this numeric value specifies the p-value threshold at which a differential correlation p-value is deemed significant for differential correlation class calculation. Default = 1, as investigators may use different cutoff thresholds; however, this can be lowered to establish significant classes as desired.
corSigThresh	Threshold at which the correlation p-values must be below in order to be called "significant". Default = 0.05.
convertClasses	Logical indicating whether the returned classes should be in numeric (factor) format or character format indicating the "actual" class.

Value

A numeric vector of classes derived from each of the input vectors.

Examples

```
rho1 = runif(100, -1, 1); rho2 = runif(100, -1, 1)
pvalsA = runif(100, 0, 1); pvalsB = runif(100, 0, 1); dcor_pvals = runif(100, 0, 1)
cor_classes = dCorClass(rho1, pvalsA, rho2, pvalsB, dcor_pvals)
cor_classes = dCorClass(rho1, pvalsA, rho2, pvalsB, dcor_pvals, convertClasses = TRUE)
```

dCorMats *Finds differential correlations between matrices.*

Description

Takes two corresponding correlation and nsamp matrices and returns matrices for the scaled difference in correlation as well as the p-value of that difference.

Usage

```
dCorMats(matA, nmatA, matB, nmatB, corr_cutoff = 0.99, corrType = "pearson",
  secondMat = FALSE, signType = "none")
```

Arguments

matA	Correlation matrix with numeric entries.
nmatA	Number of samples (nsamp) matrix with numeric entries, corresponding to the number of samples used for each of the correlations calculated in matA.
matB	Correlation matrix with numeric entries.
nmatB	Number of samples (nsamp) matrix with numeric entries, corresponding to the number of samples used for each of the correlations calculated in matB.
corr_cutoff	Cutoff specifying correlation values beyond which will be truncated to this value, to reduce the effect of outlier correlation values when using small sample sizes. Note that this does NOT affect the reported underlying correlation values, but does affect the z-score difference of correlation calculation.
corrType	The correlation type of the analysis, limited to "pearson" or "spearman".
secondMat	Logical indicator of whether there is a second matrix in the comparison or not. If no, then computations will only be performed the upper triangle of the input matrices.
signType	Coerce all correlation coefficients to be either positive (via "positive"), negative (via "negative"), or none (via "none"). This could be used if you think that going from a positive to a negative correlation is unlikely to occur biologically and is more likely to be due to noise, and you want to ignore these effects. Note that this does NOT affect the reported underlying correlation values, but does affect the z-score difference of correlation calculation. Default = "none", for no coercing.

Value

A list of two differential correlation matrices: one for the difference in z-scores and one for the corresponding p-values.

dCorrs *Differential correlation between two conditions.*

Description

Z-transforms correlation coefficients (Pearson or Spearman) and then calculates the difference in z-scores between the two conditions divided by the square root of the standard errors (which is inversely proportion to the sample sizes used to calculate the correlations).

Usage

```
dCorrs(rho1, n1, rho2, n2, corrType = "pearson")
```

Arguments

rho1	Numeric vector of correlation coefficients in condition 1.
n1	Numeric vector of the number of samples used in the correlation calculations in condition 1.
rho2	Numeric vector of correlation coefficients in condition 2.
n2	Numeric vector of the number of samples used in the correlation calculations in condition 2.
corrType	The correlation type of the analysis, limited to "pearson" or "spearman".

Value

Numeric vector with scaled difference in z-scores of correlations between the two conditions.

References

Tests For Rank Correlation Coefficients, I. http://biomet.oxfordjournals.org/content/44/3-4/470.full.pdf+html

Examples

```
rho1 = runif(100, -1, 1); rho2 = runif(100, -1, 1)
n1 = rep(100, 100); n2 = rep(110, 100)
dcorrs_res = dCorrs(rho1, n1, rho2, n2)
```

dcPair-class

S4 class for pairwise differential correlation matrices and associated info.

Description

S4 class for pairwise differential correlation matrices and associated info.

Slots

`corA` Correlation matrix for identifiers in condition A.

`corPvalA` Matrix of correlation significances in condition A.

`corB` Correlation matrix for identifiers in condition B.

`corPvalB` Matrix of correlation significances in condition B.

`ZDiff` Matrix of differences in z-values between conditions.

`PValDiff` Matrix of p-values for differences in z-values between conditions.

`options` Character vector of options used in the function.

dcTopPairs

Creates a data frame for the top differentially correlated gene pairs in your data set.

Description

Reads in a dcPair object and outputs a table of all gene pairs (or just the top n pairs), sorted by their unadjusted differential correlation p-value.

Usage

```
dcTopPairs(dcObject, nPairs, adjust = "none", plotFdr = FALSE,
  classify = TRUE, sigThresh = 1, corSigThresh = 0.05,
  zScorePerm = NULL, verbose = FALSE, compare = NULL, secondMat = FALSE)
```

Arguments

`dcObject`	The dcPair class object which you'd like to convert into a table.
`nPairs`	The number of gene pairs to display in the resulting table.
`adjust`	Allows for resulting p-values to be corrected for multiple hypothesis tests, optional. Some non-default choices require the "fdrtool" package or the "qvalue". Default = "none", which means that no p-value adjustment is performed. Other options include "perm" to use permutation samples, methods in ?p.adjust (i.e., "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"), and methods in ?fdrtool (i.e., "fndr", "pct0", "locfdr").

plotFdr	Allows for plotting of p-value adjustment result, if this is chosen. Requires fdrtool or qvalue package. Default = FALSE.
classify	Binary value specifying whether the correlation values in each condition and differential correlation scores should be used to classifying the resulting identifiers into groups. Default = TRUE
sigThresh	If classify = TRUE, this numeric value specifies the p-value threshold at which a differential correlation p-value is deemed significant for differential correlation class calculation. Default = 1, as investigators may use different cutoff thresholds; however, this can be lowered to establish significant classes as desired.
corSigThresh	If classify = TRUE, this numeric value specifies the p-value threshold at which a correlation p-value is deemed significant. Default = 0.05.
zScorePerm	A matrix of values with z-scores from permutation tests to be used to generate empirical p-values. Default = NULL.
verbose	Whether summaries of the operations should be reported.
compare	Vector of two character strings, each corresponding to one group name in the design matrix, that should be compared.
secondMat	Logical indicator of whether there is a second matrix in the comparison or not.

Value

A table containing columns for each name in the considered gene pair (the order of which is arbitrary), correlation values in each condition, differences in z-score of the correlation, and p-values for that z-score difference.

---

ddcorAll *Calls the DGCA pairwise pipeline.*

Description

Runs the full discovery of differential correlation (ddcor) section for comparing pairwise correlations across conditions in the Differential Gene Correlation Analysis (DGCA) package.

Usage

```
ddcorAll(inputMat, design, compare, inputMatB = NULL, splitSet = NULL,
  impute = FALSE, corrType = "pearson", nPairs = "all",
  sortBy = "zScoreDiff", adjust = "perm", nPerms = 10, classify = TRUE,
  sigThresh = 1, corSigThresh = 0.05, heatmapPlot = FALSE,
  color_palette = NULL, verbose = FALSE, plotFdr = FALSE,
  corr_cutoff = 0.99, signType = "none", getDCorAvg = FALSE,
  dCorAvgType = "gene_average", dCorAvgMethod = "median",
  oneSidedPVal = FALSE, customize_heatmap = FALSE, heatmapClassic = FALSE,
  corPower = 2, ...)
```

Arguments

inputMat	The matrix (or data.frame) of values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers whose correlations and differential correlations you are interested in analyzing, while the columns should correspond to the rows of the design matrix and should be separable into your groups.
design	A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see vignettes for more information.
compare	Vector of two character strings, each corresponding to one group name in the design matrix, that should be compared.
inputMatB	Optional, secondary input matrix that allows you to calculate correlation and differential correlation for the rows between inputMat and imputMatB. Default = NULL.
splitSet	Optional character vector that splits the first matrix into two matrices and calculates differential correlation across these matrices. Common use case is when you want the differential correlation of a small set of identifiers (e.g., one), compared with all of the other identifiers in the matrix in each condition. Cannot be used when a second matrix is inputted – setting both of arguments to non-NULL values will result in an error.
impute	A binary variable specifying whether values should be imputed if there are missing values. Note that the imputation is performed in the full input matrix (i.e., prior to subsetting) and uses k-nearest neighbors.
corrType	The correlation type of the analysis, limited to "pearson" or "spearman". Default = "pearson".
nPairs	Either a number, specifying the number of top differentially correlated identifier pairs to display in the resulting table, or a the string "all" specifying that all of the pairs should be returned. If splitSet is specified, this is reset to the number of non-splitSet identifiers in the input matrix, and therefore will not be evaluated.
sortBy	Character string specifying the way by which you'd like to sort the resulting table.
adjust	Allows for resulting p-values to be corrected for multiple hypothesis tests, optional. Some non-default choices require the "fdrtool" package or the "qvalue". Default = "none", which means that no p-value adjustment is performed. Other options include "perm" to use permutation samples, methods in ?p.adjust (i.e., "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"), and methods in ?fdrtool (i.e., "fndr", "pct0", "locfdr").
nPerms	Number of permutations to generate. If NULL, permutation testing will not be performed. Default = "10".
classify	Binary value specifying whether the correlation values in each condition and differential correlation scores should be used to classifying the resulting identifiers into groups. Default = TRUE

`sigThresh` If classify = TRUE, this numeric value specifies the p-value threshold at which a differential correlation p-value is deemed significant for differential correlation class calculation. Default = 1, as investigators may use different cutoff thresholds; however, this can be lowered to establish significant classes as desired.

`corSigThresh` If classify = TRUE, this numeric value specifies the p-value threshold at which a correlation p-value is deemed significant. Default = 0.05.

`heatmapPlot` Option indicating whether a heatmap of the differential correlations between the two conditions should be plotted. Default = TRUE.

`color_palette` Color palette for plotting the heatmap. If not specified, the heatmap defaults to a red-green color-blind palette with bluish green indicating negative correlations and vermillion indicating positive correlations. Default = NULL

`verbose` Option indicating whether the program should give more frequent updates about its operations. Default = FALSE.

`plotFdr` Allows for plotting of fdrtool p-value adjustment result OR empirical FDR q-value adjustment technique, if either of these are chosen. Requires fdrtool package OR qvalue package. Default = FALSE.

`corr_cutoff` Cutoff specifying correlation values beyond which will be truncated to this value, to reduce the effect of outlier correlation values when using small sample sizes. Note that this does NOT affect the underlying correlation values, but does affect the z-score difference of correlation calculation in the dcTopPairs table. Default = 0.99

`signType` Coerce all correlation coefficients to be either positive (via "positive"), negative (via "negative"), or none (via "none") prior to calculating differential correlation. This could be used if, e.g., you think that going from a positive to a negative correlation is unlikely to occur biologically and is more likely to be due to noise, and you want to ignore these effects. Note that this does NOT affect the reported underlying correlation values, but does affect the z-score difference of correlation calculation. Default = "none", for no coercing.

`getDCorAvg` Logical, specifying whether the average difference in correlation between groups should be calculated. Default = FALSE

`dCorAvgType` Character vector specifying the type of average differential correlation calculation that should be performed. Only evaluated if dCorAge is TRUE. Types = c("gene_average", "total_average", "both"). gene_average calculates whether each genes' differential correlation with all others is more than expected via permutation samples (and empirical FDR adjustment, in the case of > 1 gene), while total_average calculates whether the total average differential correlation is higher than expected via permutation samples. "both" performs both of these. If splitSet is specified, then only genes in the splitSet have their average gene differential correlation calculated if gene_average is chosen.

`dCorAvgMethod` Character vector specifying the method for calculating the "average" differential correlation calculation that should be used. Options = "median", "mean".

`oneSidedPVal` If the dCorAvgType test is total_average, this option specifies whether a one-sided p-value should be reported, as opposed to a two-sided p-value. That is, if the average difference of z-scores is greater than zero, test whether the permutation average difference of z-scores are less than that average to get the p-value,

and vice versa for the case that the average difference of z-scores is less than 0. Otherwise, test whether the absolute value of the average difference in z-scores is greater than the absolute values of the permutation average difference in z-scores. Default = FALSE.

`customize_heatmap`
: Option to remove some default options in the heatmap plot, to allow users to add custom options.

`heatmapClassic`
: Option to make the heatmap more granular (e.g., not showing the individual gene symbols) and more of a "classic" type of heatmap. Overrides most other heatmap options.

`corPower`
: The power to raise the correlations to before plotting the classic heatmap. Larger correlation powers emphasize larger correlation values relatively more compared to smaller correlation values.

`...`
: Additional plotting arguments if heatmapPlot = TRUE.

Value

Typically, the returned object is a data frame of the table of differential correlations between conditions. In the case that dCorAvg is calculated, the returned object is instead a list containing that table as well as the object summarizing the difference in average correlation for the specified portion of the data set.

Examples

```
data(darmanis); data(design_mat); darmanis_subset = darmanis[1:30, ]
ddcor_res = ddcorAll(inputMat = darmanis_subset, design = design_mat,
compare = c("oligodendrocyte", "neuron"))
```

---

ddcorFindSignificant *Find groups of differentially correlated gene symbols.*

Description

Takes a table of differentially correlated genes with respect to one gene in the Gene2 column and returns the a list of vectors with unique, non-NA gene symbols for genes in each of the differentially correlated classes.

Usage

```
ddcorFindSignificant(ddcor_res, pval_gene_thresh = 0.05, adjusted = FALSE,
  classes = FALSE, geneNameCol = c("Gene1", "Gene2"),
  unique_genes = FALSE, regcor = FALSE)
```

Arguments

ddcor_res	The table of differential correlations outputted from ddcor. Expected to have pValDiff or pValDiff_adj columns as well as zScoreDiff, Gene1, +/- Classes columns.
pval_gene_thresh	p-value threshold to call a gene as having significant differential correlation or not. Default = 0.05
adjusted	Logical indicating whether adjusted p-values from the differential correlation table (i.e., column "pValDiff_adj", when adjusted = TRUE) or unadjusted p-values (i.e., column "pValDiff", when adjusted = FALSE) should be used to subset the table into significant and non-significant portions. Default = FALSE
classes	Logical indicator specifying whether individual differential correlation gene classes should be extracted from the table or not. If not, only the zScoreDiff column is used to specify positively or negatively differentially correlated genes between the two conditions. Default = FALSE
geneNameCol	Character vector specifying the name of the columns that are used to extract the gene symbols. Note that the default is c("Gene1", "Gene2"), but this only makes sense in the context of a full DGCA experiment. In the case of a splitSet, you may want to use "Gene1" to avoid counting the splitSet names in all of the categories.
unique_genes	Logical, if TRUE indicates that unique gene symbols from each category compared to the other groups should be chosen prior to GO enrichment analysis.
regcor	Logical specifying whether the ddcorGO analysis should be performed on the results of a regcor data analysis. Note that the classes option is not available in this case.

Value

A list of significantly differentially correlated genes.

---

ddcorGO *Gene ontology of differential correlation-classified genes.*

Description

Extracts a data frame of the top enriched gene sets in gene ontology databases using the hypergeometric test for gene synmols that are members of gene pairs in each of the classes specified in the differentially correlated gene pairs input table. Default parameter settings are to take in a result table with HGNC symbols and convert them to Ensembl symbols for gene ontology testing.

Usage

```
ddcorGO(ddcor_res, universe, pval_gene_thresh = 0.05, classes = FALSE,
  geneNameCol = c("Gene1", "Gene2"), pval_GO_cutoff = 1,
  HGNC_clean = TRUE, HGNC_switch = TRUE, gene_ontology = "all",
  adjusted = FALSE, annotation = "org.Hs.eg.db", conditional = FALSE,
  calculateVariance = FALSE, unique_genes = FALSE, regcor = FALSE,
  ddcor_find_significant = TRUE, ddcorGO_res = NULL)
```

Arguments

ddcor_res	The table of differential correlations outputted from ddcor. Expected to have pValDiff or pValDiff_adj columns as well as zScoreDiff, Gene1, +/- Classes columns.
universe	Character vector of gene symbols which should be used as the background in the hypergeomtric test. If using this in the context of a DGCA experiment, this gene list most likely should be the gene set post-filtering, but prior to differential correlation analysis.
pval_gene_thresh	p-value threshold to call a gene as having significant differential correlation or not.
classes	Logical indicator specifying whether individual differential correlation gene classes should be extracted from the table or not. If not, only the zScoreDiff column is used to specify positively or negatively differentially correlated genes between the two conditions.
geneNameCol	Character vector specifying the name of the columns that are used to extract the gene symbols. Note that the default is c("Gene1", "Gene2"), but this only makes sense in the context of a full DGCA experiment. In the case of a splitSet, you may want to use "Gene1" to avoid counting the splitSet names in all of the categories.
pval_GO_cutoff	Cutoff for the unadjusted p-values of gene ontology terms in the enrichment tests that should be displayed in the resulting table.
HGNC_clean	Logical indicating whether the input gene symbols should be switched to clean HGNC symbols using the checkGeneSymbols function from the R package HGNChelper. Only applies if HGNC symbols are inputted.
HGNC_switch	Logical indicating whether or not the input gene symbols need to be switched from HGNC to Ensembl, the latter of which is required for GOstats enrichment test. Note that this is done by selecting the first Enembl symbol that maps to a particular HGNC symbol, which is not always unique. If you need more precision on the conversion, you should do this outside of the function and insert the Ensembl list to the function.
gene_ontology	A string specifying the branch of GO that should be used for enrichment analysis. One of "BP" (Biological Process), "MF" (Molecular Function), "CC" (Cellular Component), or "all". If "all" is chosen, then this function finds the enrichment for all of the terms and combines them into one table. Default = "all"

`adjusted` Logical indicating whether adjusted p-values from the differential correlation table (i.e., column "pValDiff_adj", when adjusted = TRUE) or unadjusted p-values (i.e., column "pValDiff", when adjusted = FALSE) should be used to subset the table into significant and non-significant portions.

`annotation` The library indicating the GO annotation database from which the Go terms should be mapped to gene symbols. Default = "org.Hs.eg.db", which is the table for Homo sapiens. Other common choices include "org.Mm.eg.db", "org.Rn.eg.db". The corresponding annotation library needs to be installed.

`conditional` Logical specifying whether the GO analysis should be done conditionally to take into account the hierarchical structure of the GO database in making sense of the gene set enrichments.

`calculateVariance`
Optionally, find the variance of the odds ratio for each enrichment test. In particular, this finds the standard error of the log odds ratio, which converges to a normal distribution much more quickly than the non-log OR.

`unique_genes` Logical, if TRUE indicates that unique gene symbols within gene pairs from each category compared to the other groups should be chosen prior to GO enrichment analysis.

`regcor` Logical specifying whether the ddcorGO analysis should be performed on the results of a regcor data analysis. Note that the classes option is not available in this case.

`ddcor_find_significant`
Logical specifying whether this enrichment analysis should be performed on the result of a ddcor analysis. If FALSE, then a ddcorGO_res object, which is a named list of gene vectors, must be defined instead.

`ddcorGO_res` Optional named list of gene vectors to find the enrichment of if ddcor_find_signficiant is FALSE.

Value

A list of data frames corresponding to the gene ontology enrichment analysis results for the extracted gene sets from each of the differential correlation classes.

References

Agresti A: Categorical Data Analysis. 2012:70-77.

`ddMEGENA` *Integration function to use MEGENA to perform network analyses of DGCA results.*

Description

Takes a table of results from a DGCA analysis and inputs it into the MEGENA package pipeline.

Usage

```
ddMEGENA(ddcor_res, adjusted = TRUE, pval_gene_thresh = 0.05,
  evalCompactness = TRUE, nPerm = 100, hubPVal = 0.05,
  modulePVal = 0.05, minModSize = 10, maxModSize = 1000,
  saveOutput = FALSE, parallelize = FALSE, nCores = 4, ...)
```

Arguments

ddcor_res	The table of differential correlations outputted from ddcor. Expected to have pValDiff or pValDiff_adj columns as well as zScoreDiff, Gene1, +/- Classes columns.
adjusted	Logical indicating whether adjusted p-values from the differential correlation table (i.e., column "pValDiff_adj", when adjusted = TRUE) or unadjusted p-values (i.e., column "pValDiff", when adjusted = FALSE) should be used to subset the table into significant and non-significant portions.
pval_gene_thresh	p-value threshold to call a gene as having significant differential correlation or not.
evalCompactness	Logical indicating whether or not the resulting modules should be filtered for compactness. For inputs with relatively small numbers of significant gene pairs, this may not be desirable. Note that if this option is not chosen, all of the modules will be returned, but some of the module-specific results will not be available for all of these modules.
nPerm	The number of permutations to use in evaluating module hubs and module compactness in do.MEGENA.
hubPVal	The p-value threshold used to classify a gene as a hub within a module.
modulePVal	The p-value threshold used to include or disclude modules following module compactness evaluation in do.MEGENA.
minModSize	The minimum module size.
maxModSize	The minimum module size.
saveOutput	Whether the output of MEGENA should be saved in the current directory. Default = FALSE.
parallelize	Logical indicating whether or not multiple cores should be utilized as a form of parallel processing. Requires the doMC package.
nCores	If parallelize is TRUE, the number of cores to use in the processing. Ignored if parallelize is FALSE.
...	Additional arguments to do.MEGENA from the MEGENA R package.

Value

A list containing a the planar filter network, the data frame of identified differentially correlated modules, as well as various other objects including module-specific hub genes, depending on the parameters chosen.

ddplot *Create a heatmap showing the correlations in two conditions.*

Description

This function orders the differences in correlations between conditions by the median strength of correlation differences for each gene and plots a heatmap of the correlations in each condition (lower = condition A, upper = condition B) using the heatmap.2 function from the gplots package.

Usage

```
ddplot(dcObject = NULL, corMatA = NULL, corMatB = NULL, zDiff = NULL,
  flip = TRUE, color_palette = NULL, customize_heatmap = FALSE,
  heatmapClassic = FALSE, corPower = 2, ...)
```

Arguments

dcObject	A differential correlation object from which correlation and differential correlation matrices will be extracted. Optional; can also input the correlation matrices and differential correlation matrix individually.
corMatA	Optional, correlation matrix from condition A. Will be plotted in the lower left triangle.
corMatB	Optional, correlation matrix from condition B. Will be plotted in the upper right triangle.
zDiff	Optional, difference measure of correlations between conditions A and B.
flip	Switch the ordering of z-differences to be inverse. Default = TRUE
color_palette	Colors for plotting the heatmap. If not specified, defaults to a color-blind palette where blue corresponds to a negative correlation and orange/red corresponds to a positive one.
customize_heatmap	Option to remove some default options in the heatmap plot, to allow users to add custom options.
heatmapClassic	Option to make the heatmap more granular (e.g., not showing the individual gene symbols) and more of a "classic" type of heatmap. Overrides most other heatmap options.
corPower	The power to raise the correlations to before plotting the classic heatmap. Larger correlation powers emphasize larger correlation values relatively more compared to smaller correlation values.
...	Additional plotting arguments to the heatmap.2 function.

Value

The sorted difference in z-score matrix in both conditions, which you can use to create your own plot if you'd prefer.

design_mat — *Design matrix of cell type specifications of the single-cell RNA-seq samples.*

Description

Cell type specifications were performed by the authors. The total data set can be downloaded by following the links in the original paper.

Usage

```
design_mat
```

Format

An object of class `matrix` with 158 rows and 2 columns.

References

Darmanis S, Sloan SA, Zhang Y, et al. A survey of human brain transcriptome diversity at the single cell level. Proc Natl Acad Sci USA. 2015;112(23):7285-90.

DGCA — *DGCA: An R package for Differential Gene Correlation Analysis*

Description

The DGCA package provides three major classes of functions: 1) Functions to calculate correlations, correlation significances, and number of samples in each correlation calculation based on an input matrix and a design matrix. 2) Functions to calculate differential correlations between regions, which in the current package can only be pairwise (i.e., one condition vs another). 3) Functions to extract, sort information about the differential correlation calculations in a convenient format. The first two functions comprise the discovery of differential correlation (ddcor) portion of the package, which is why the names of the functions and object names often begin with ddcor. Note that DGCA makes use of the SAF = getOption("stringsAsFactors"); on.exit(options(stringsAsFactors = SAF)); options(stringsAsFactors = FALSE) design pattern many times in order to avoid errors related to stringsAsFactors in porting code to new environments. This should not affect the stringsAsFactors options in your environment; however, you may want to be aware of this.

extractModuleGO *Extract results from the module GO analysis*

Description

Turns the list of lists from the moduleGO function into a more comprehensible data frame. Note that if a GO term enrichment does not exist for that module, it is set as NA.

Usage

```
extractModuleGO(moduleGO_list, labels = NULL)
```

Arguments

moduleGO_list	The list of list of data frames for each module to be turned into a data frame.
labels	Optional, a list of module names. Optional; if not inputted, these will be extracted from the list of module GO enrichments.

Value

A data frame summarizing the GO term enrichments from each group, with columns on the ordered by the minimum p-value for OR term enrichment in any group.

filterGenes *Filter rows out of a matrix.*

Description

Filter out rows in an input matrix that are not above a certain percentile with respect to a central tendency and/or dispersion measure. To be used, e.g, prior to differential correlation testing with the function ddcorall.

Usage

```
filterGenes(inputMat, filterTypes = "central", keepRows = NULL,
  filterCentralType = "median", filterDispersionType = "dispersion_index",
  filterCentralPercentile = 0.25, filterDispersionPercentile = 0.25,
  sequential = FALSE, allGroups = FALSE, design = NULL)
```

Arguments

inputMat	The matrix (or data.frame) of all numeric values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers that you are interested in protecting from the filter, if any.
filterTypes	Vector containing up to two character strings, specifying the methods that should be used for filtering genes. Options include "central" and "dispersion" for filtering based on the measures of central tendency and dispersion, respectively. To use both, set this to c("central", "dispersion").
keepRows	Optional character vector, specifying rownames (i.e., symbols) that should not be filtered out of the matrix even if they are found to be below the quantile specified for either the central tendency or dispersion, as applicable.
filterCentralType	Method to be used for filtering for the central tendency of the input matrix. Options = "mean" (for arithmetic mean) and "median".
filterDispersionType	Method to be used for filtering for the dispersion of the input matrix. Options = "dispersion_index", "cv" (for coefficient of variation), and "variance".
filterCentralPercentile	If central tendency filtering is used, the quantile of the central tendency below which rows will be filtered out.
filterDispersionPercentile	If dispersion filtering is used, the quantile of the dispersion measure below which rows will be filtered out.
sequential	If both central tendency and dispersion measures and used for filtering the input matrix, then sequential is a logical flag indicating whether the central tendency filtering steps should be performed prior to the dispersion filtering step (and quantile cutoff specification; if sequential = TRUE), or independently (if sequential = FALSE).
allGroups	Logical for whether genes need to pass the filter in all of the groups specified in the design matrix.
design	A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see vignettes for more information.

Value

A filtered matrix.

Examples

```
data(darmanis); data(design_mat); darmanis_subset = darmanis[1:30, ]
filtered_mat = filterGenes(inputMat = darmanis_subset, filterTypes = "central")
filtered_mat_both = filterGenes(inputMat = darmanis_subset,
 filterTypes = c("central", "dispersion"), filterCentralType = "mean",
 filterDispersionPercentile = 0.1)
```

```
filtered_mat_all_groups = filterGenes(inputMat = darmanis_subset,
 design = design_mat, filterTypes = "dispersion", allGroups = TRUE)
```

`findGOTermEnrichment` *Find GO enrichment for a gene vector (using GOstats).*

Description

Given a gene character vector and a universe character vector, which can be either Ensembl or HGNC symbols, find the over-representation enrichment of the gene list relative to the universe in a gene ontology category using the hypergeometric test and the GOstats R package.

Usage

```
findGOTermEnrichment(gene_vector, universe, pval_GO_cutoff = 1,
  HGNC_switch = TRUE, HGNC_clean = TRUE, gene_ontology = "all",
  conditional = TRUE, annotation = "org.Hs.eg.db", cleanNames = FALSE)
```

Arguments

`gene_vector`	Character vector gene symbols of interest.
`universe`	Character vector of gene symbols which should be used as the background in the hypergeomtric test. If using this in the context of a ddcor experiment, this gene list most likely should be the gene set post-filtering, but prior to differential correlation analysis.
`pval_GO_cutoff`	Cutoff for the p-values of gene ontology terms in the enrichment tests that should be displayed in the resulting table.
`HGNC_switch`	Logical indicating whether or not the input gene symbols need to be switched from HGNC to Ensembl, the latter of which is required for GOstats enrichment test. Note that this is done by selecting the first Enembl symbol that maps to a particular HGNC symbol, which is not always unique. If you need more precision, you should do this outside of the function and insert the Ensembl list to the function. Only applies if cleanNames is TRUE.
`HGNC_clean`	Logical indicating whether the input gene symbols should be switched to clean HGNC symbols using the checkGeneSymbols function from the R package HGNChelper. Only applies if HGNC symbols are inputted and cleanNames is TRUE.
`gene_ontology`	A string specifying the branch of GO that should be used for enrichment analysis. One of "BP" (Biological Process), "MF" (Molecular Function), "CC" (Cellular Component), or "all". If "all" is chosen, then this function finds the enrichment for all of the terms and combines them into one table. Default = "all"
`conditional`	Logical specifying whether the GO analysis should be done conditionally to take into account the hierarchical structure of the GO database in making sense of the gene set enrichments. Default = TRUE.

annotation — The library indicating the GO annotation database from which the Go terms should be mapped to gene symbols. Default = "org.Hs.eg.db", which is the table for Homo sapiens. Other common choices include "org.Mm.eg.db", "org.Rn.eg.db". The corresponding annotation library needs to be installed.

cleanNames — Logical, indicating whether the gene names for the universe and gene vector should be cleaned prior to enrichment analysis.

Value

A data frame with the term enrichments of the GO enrichment analysis given the input gene set and universe.

---

getCors *Compute matrices necessary for differential correlation calculation.*

Description

As a first step in the standard DGCA workflow, this function reads in a design matrix and an input matrix, splits the input matrix by the named groups defined in the design matrix, and outputs a list of matrices (correlation matrix, correlation significance matrix, and number of samples matrix) to be used by downstream parts of the analysis.

Usage

```
getCors(inputMat, design, inputMatB = NULL, impute = FALSE,
  corrType = "pearson")
```

Arguments

inputMat — The matrix (or data.frame) of values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers whose correlations and differential correlations you are interested in analyzing, while the columns should correspond to the rows of the design matrix and should be separable into your groups.

design — A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see vignettes for more information.

inputMatB — Optional, secondary input matrix that allows you to calculate correlation and differential correlation for the rows between inputMat and imputMatB.

impute — A binary variable specifying whether values should be imputed if there are missing values. Note that the imputation is performed in the full input matrix (i.e., prior to subsetting) and uses k-nearest neighbors.

corrType — The correlation type of the analysis, limited to "pearson" or "spearman". Default = "pearson".

Value

A corMats S4 class object, containing a list of matrices from each group, the design matrix, and a character vector of options.

Examples

```
data(darmanis); data(design_mat); darmanis_subset = darmanis[1:30, ]
cors_res = getCors(inputMat = darmanis_subset, design = design_mat)
```

`getDCorPerm` *Get permuted groupwise correlations and pairwise differential correlations.*

Description

Takes input and methods and randomly permutes the data to do getCor as well as group-specific pairwiseDCor.

Usage

```
getDCorPerm(inputMat, design, compare, inputMatB = NULL, impute = FALSE,
  nPerms = 10, corrType = "pearson", corr_cutoff = 0.99,
  signType = "none")
```

Arguments

`inputMat`	The matrix (or data.frame) of values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers whose correlations and differential correlations you are interested in analyzing, while the columns should correspond to the rows of the design matrix and should be separable into your groups.
`design`	A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see vignettes for more information.
`compare`	Vector of two character strings, each corresponding to one name in the list of correlation matrices that should be compared.
`inputMatB`	Optional, secondary input matrix that allows you to calculate correlation and differential correlation for the rows between inputMat and imputMatB.
`impute`	A binary variable specifying whether values should be imputed if there are missing values. Note that the imputation is performed in the full input matrix (i.e., prior to subsetting) and uses k-nearest neighbors.
`nPerms`	Number of permutations to generate.
`corrType`	The correlation type of the analysis, limited to "pearson" or "spearman".

corr_cutoff	Cutoff specifying correlation values beyond which will be truncated to this value, to reduce the effect of outlier correlation values when using small sample sizes. Default = 0.99
signType	Coerce all correlation coefficients to be either positive (via "positive"), negative (via "negative"), or none (via "none"). This could be used if you think that going from a positive to a negative correlation is unlikely to occur biologically and is more likely to be due to noise, and you want to ignore these effects. Note that this does NOT affect the reported underlying correlation values, but does affect the z-score difference of correlation calculation. Default = "none", for no coercing.

Value

An array of permuted differences in z-scores calculated between conditions, with the third dimension corresponding to the number of permutations performed.

`getDCors` *Get groupwise correlations and pairwise differential correlations.*

Description

Takes input and methods to perform getCor as well as group-specific pairwiseDCor.

Usage

```
getDCors(inputMat, design, compare, inputMatB = NULL, impute = FALSE,
  corrType = "pearson", corr_cutoff = 0.99, signType = "none")
```

Arguments

inputMat	The matrix (or data.frame) of values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers whose correlations and differential correlations you are interested in analyzing, while the columns should correspond to the rows of the design matrix and should be separable into your groups.
design	A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see ?model.matrix for more information.
compare	Vector of two character strings, each corresponding to one name in the list of correlation matrices that should be compared.
inputMatB	Optional, secondary input matrix that allows you to calculate correlation and differential correlation for the rows between inputMat and imputMatB.
impute	A binary variable specifying whether values should be imputed if there are missing values. Note that the imputation is performed in the full input matrix (i.e., prior to subsetting) and uses k-nearest neighbors.

corrType	The correlation type of the analysis, limited to "pearson" or "spearman".
corr_cutoff	Cutoff specifying correlation values beyond which will be truncated to this value, to reduce the effect of outlier correlation values when using small sample sizes.
signType	Coerce all correlation coefficients to be either positive (via "positive"), negative (via "negative"), or none (via "none"). This could be used if you think that going from a positive to a negative correlation is unlikely to occur biologically and is more likely to be due to noise, and you want to ignore these effects. Note that this does NOT affect the reported underlying correlation values, but does affect the z-score difference of correlation calculation. Default = "none", for no coercing.

Value

A dcPair class object, containing the difference in z-scores for each comparison, the p-values of that differences, and the original correlation matrices and significances for subsequent classification steps. data(darmanis); data(design_mat); darmanis_subset = darmanis[1:30,] dcors_res = getDCors(inputMat = darmanis_subset, design = design_mat, compare = c("oligodendrocyte", "neuron"))

---

getGroupsFromDesign *Split input matrix(es) based on the design matrix.*

Description

This function splits the input matrix(es) based on a design matrix, into a named list of subsetted matrices. If the design matrix has no names, this function will create names for the resulting list of matrices.

Usage

```
getGroupsFromDesign(inputMat, design, inputMatB = NULL, secondMat = FALSE)
```

Arguments

inputMat	Input (e.g., expression) matrix which will be subsetted.
design	Standard design matrix, must specify at least two conditions. For more info, see ?model.matrix
inputMatB	Optional input (e.g., expression) matrix which will be subsetted in the same way.
secondMat	Logical value indicating whether there is a second input matrix to be subsetted.

Value

A list whose first element is a list of subsetted matrices and whose second element is a list of group names.

Examples

```
data(darmanis); data(design_mat); darmanis_subset = darmanis[1:30, ]
groups_from_design = getGroupsFromDesign(inputMat = darmanis_subset, design = design_mat)
str(groups_from_design)
```

---

makeDesign *Create a design matrix from a character vector.*

Description

This function wraps around model.matrix to create a design matrix based on the different levels in model.matrix. Note that the order of the column names of the design matrix will be in lexicographic order according to the locale in use. For more, see ?Comparison

Usage

```
makeDesign(x)
```

Arguments

x	Character vector to be used to create the design matrix.

Value

A design matrix as the same type as returned using ?model.matrix.

Examples

```
n_oligo_samples = 38; n_neuron_samples = 120
cell_type = c(rep("oligodendrocyte", n_oligo_samples), rep("neuron", n_neuron_samples))
design_mat = makeDesign(cell_type)
```

---

matCorr *Calculate a correlation matrix.*

Description

This function takes one or two input matrices and calculates a correlation matrix from it using the speed-optimized correlation function from WGCNA.

Usage

```
matCorr(matA, corrType, use = "pairwise.complete.obs", matB = NULL,
  secondMat = FALSE)
```

Arguments

matA	Input data matrix with numeric entries.
corrType	The type of correlation to be performed. Either "pearson" or "spearman".
use	The "use" method for performing the correlation calculation. See ?cor for more information. Default = "pairwise.complete.obs" (which is one of the speed-optimized versions; see ?WGCNA::cor for more).
matB	Optional input data matrix with which the comparison with matA will be made.
secondMat	Logical indicator of whether there is a second matrix in the comparison or not.

Value

A correlation matrix. data(darmanis); darmanis_subset = darmanis[1:30,] matcor_res = matCorr(matA = darmanis_subset, corrType = "pearson")

---

matCorSig *Calculate correlation matrix p-values.*

Description

Calculate two-sided p-values from a pairwise correlations matrix and a corresponding "number of samples" matrix.

Usage

```
matCorSig(corrs, nsamp, secondMat = FALSE)
```

Arguments

corrs	Computed correlation matrix.
nsamp	Computed number of samples used per call in the correlation matrix.
secondMat	Logical indicator of whether there is a second matrix in the comparison or not.

Value

A matrix of p-values.

References

HMisc R package https://cran.r-project.org/web/packages/Hmisc/index.html

matNSamp *Find the number of non-missing values.*

Description

This function calculates the pairwise number of non-missing values in a matrix.

Usage

```
matNSamp(matA, impute = FALSE, matB = NULL, secondMat = FALSE)
```

Arguments

matA	Input data matrix with numeric entries.
impute	Binary value; if true, indicates that imputation was performed previously, and so checking for NAs is not necessary.
matB	Optional input data matrix with which the comparison with matA will be made.
secondMat	Logical indicator of whether there is a second matrix in the comparison or not.

Value

A number of samples (nsamp) matrix. data(darmanis); darmanis_subset = as.matrix(darmanis[1:30,]) nsamp_res = matNSamp(darmanis_subset) darmanis_subset[1, 1] = NA nsamp_res_na = matNSamp(darmanis_subset)

moduleDC *Calculate modular differential connectivity (MDC)*

Description

Takes modules of genes (possibly overlapping) and calculates the change in correlation among those genes between two conditions. Also reports the genes with the strongest gain in connectivity (i.e., average difference in z-score of > 0) and the strongest loss of correlation between conditions for each module, if any pass the significance measure specified.

Usage

```
moduleDC(inputMat, design, compare, genes, labels, corr_cutoff = 0.99,
  signType = "none", corrType = "pearson", nPerms = 50,
  oneSidedPVal = FALSE, gene_avg_signif = 0.05, number_DC_genes = 3,
  dCorAvgMethod = "median")
```

Arguments

inputMat	The matrix (or data.frame) of values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers whose correlations and differential correlations you are interested in analyzing, while the columns should correspond to the rows of the design matrix and should be separable into your groups.
design	A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see vignettes for more information.
compare	Vector of two character strings, each corresponding to one group name in the design matrix, that should be compared.
genes	A character vector specifying gene symbols, present as rows in the inputMat, corresponding to each module label in the labels argument.
labels	A character vector specifying module label names, one for each gene symbol in the genes argument, with overlap allowed (i.e., each gene can be in more than one module).
corr_cutoff	Cutoff specifying correlation values beyond which will be truncated to this value, to reduce the effect of outlier correlation values when using small sample sizes. Note that this does NOT affect the underlying correlation values, but does affect the z-score difference of correlation calculation in the dcTopPairs table. Default = 0.99
signType	Coerce all correlation coefficients to be either positive (via "positive"), negative (via "negative"), or none (via "none") prior to calculating differential correlation. This could be used if, e.g., you think that going from a positive to a negative correlation is unlikely to occur biologically and is more likely to be due to noise, and you want to ignore these effects. Note that this does NOT affect the reported underlying correlation values, but does affect the z-score difference of correlation calculation. Default = "none", for no coercing.
corrType	The correlation type of the analysis, limited to "pearson" or "spearman". Default = "pearson".
nPerms	Number of permutations to generate in order to calculate the significance of the result.
oneSidedPVal	If the dCorAvgType test is total_average, this option specifies whether a one-sided p-value should be reported, as opposed to a two-sided p-value. That is, if the average difference of z-scores is greater than zero, test whether the permutation average difference of z-scores are less than that average to get the p-value, and vice versa for the case that the average difference of z-scores is less than 0. Otherwise, test whether the absolute value of the average difference in z-scores is greater than the absolute values of the permutation average difference in z-scores. Default = FALSE.
gene_avg_signif	The gene average differential correlation significance (adjusted for MHTC) required in order for the a gene to be reported as having a gain or loss in connectivity.

number_DC_genes
: The number of top differentially correlated genes with more correlation in each condition in each module to return in the data frame.

dCorAvgMethod
: Character vector specifying the method for calculating the "average" differential correlation calculation that should be used. Options = "median", "mean".

Value

A data frame with the module labels, the average change in difference in z-score between conditions (i.e., one measure of the modular average differential connectivity, or MeDC), and the empirical p-value for the significance of the change in correlation.

Examples

```
data(darmanis)
module_genes = list(mod1 = rownames(darmanis)[1:100],
 mod2 = rownames(darmanis)[90:190], mod3 = rownames(darmanis)[190:290])
modules = stack(module_genes)
modules$ind = as.character(modules$ind)
moduleDC_res = moduleDC(inputMat = darmanis, design = design_mat,
 compare = c("oligodendrocyte", "neuron"), genes = modules$values,
 labels = modules$ind)
```

---

moduleGO *Perform module GO-trait correlation*

Description

Takes input vectors of gene symbols, labels of corresponding modules, and a universe gene set and leverages the GOstats package to perform GO enrichment analysis.

Usage

```
moduleGO(genes, labels, universe, HGNC_clean = TRUE, HGNC_switch = TRUE,
  gene_ontology = "all", pval_GO_cutoff = 1, annotation = "org.Hs.eg.db",
  conditional = FALSE, calculateVariance = FALSE)
```

Arguments

genes
: A character vector specifying gene symbols, present as rows in the inputMat, corresponding to each module label in the labels argument.

labels
: A character vector specifying module label names, one for each gene symbol in the genes argument, with overlap allowed (i.e., each gene can be in more than one module).

universe
: Character vector of gene symbols which should be used as the background in the hypergeomtric test. If using this in the context of a DGCA experiment, this gene list most likely should be the gene set post-filtering, but prior to differential correlation analysis.

HGNC_clean — Logical indicating whether the input gene symbols should be switched to clean HGNC symbols using the checkGeneSymbols function from the R package HGNChelper. Only applies if HGNC symbols are inputted.

HGNC_switch — Logical indicating whether or not the input gene symbols need to be switched from HGNC to Ensembl, the latter of which is required for GOstats enrichment test. Note that this is done by selecting the first Ensembl symbol that maps to a particular HGNC symbol, which is not always unique. If you need more precision on the conversion, you should do this outside of the function and insert the Ensembl list to the function.

gene_ontology — A string specifying the branch of GO that should be used for enrichment analysis. One of "BP" (Biological Process), "MF" (Molecular Function), "CC" (Cellular Component), or "all". If "all" is chosen, then this function finds the enrichment for all of the terms and combines them into one table. Default = "all"

pval_GO_cutoff — Cutoff for the unadjusted p-values of gene ontology terms in the enrichment tests that should be displayed in the resulting table.

annotation — The library indicating the GO annotation database from which the Go terms should be mapped to gene symbols. Default = "org.Hs.eg.db", which is the table for Homo sapiens. Other common choices include "org.Mm.eg.db", "org.Rn.eg.db". The corresponding annotation library needs to be installed.

conditional — Logical specifying whether the GO analysis should be done conditionally to take into account the hierarchical structure of the GO database in making sense of the gene set enrichments.

calculateVariance — Optionally, find the variance of the odds ratio for each enrichment test. In particular, this finds the standard error of the log odds ratio, which converges to a normal distribution much more quickly than the non-log OR.

Value

A list of lists of df's, one corresponding to each module, containing GO enrichment information for each module in each of the GO categories selected.

---

pairwiseDCor *Calculate pairwise differential correlations.*

Description

Find the differential correlation between two conditions.

Usage

```
pairwiseDCor(corMatsObj, compare, corr_cutoff = 0.99, corrType = "pearson",
  secondMat = FALSE, signType = "none")
```

Arguments

corMatsObj	A class object containing a named list of lists of matrices, one list per condition, with each list containing a correlation matrix, a correlation significance p-values matrix, and a "number of samples used to calculate the correlation" matrix.
compare	Vector of two character strings, each corresponding to one name in the list of correlation matrices that should be compared.
corr_cutoff	Cutoff specifying correlation values beyond which will be truncated to this value, to reduce the effect of outlier correlation values when using small sample sizes.
corrType	The correlation type of the analysis, limited to "pearson" or "spearman".
secondMat	Logical indicator of whether there is a second matrix in the comparison or not.
signType	Coerce all correlation coefficients to be either positive (via "positive"), negative (via "negative"), or none (via "none"). This could be used if you think that going from a positive to a negative correlation is unlikely to occur biologically and is more likely to be due to noise, and you want to ignore these effects. Note that this does NOT affect the reported underlying correlation values, but does affect the z-score difference of correlation calculation. Default = "none", for no coercing.

Value

A dcPair class object, containing the difference in z scores for each comparison, the p-values of that differences, and the original correlation matrices and significances for subsequent classification steps.

---

permQValue *Calculate q-values from DGCA class objects based on permutation-based empirical null statistics.*

Description

First, estimate empirical p-values based on a comparison of the actual and permuted test statistics. Next, estimate the proportion of true null hypotheses using the qvalue package as well as qvalues from the empirical p-values, using this value. If the estimated pi0 <= 0, then sequentially recalculates using increasingly conservative set of lambda values, until lambda = 0.5.

Usage

```
permQValue(dcObject, permObject, secondMat, testSlot, verbose = FALSE,
  plotFdr = FALSE)
```

Arguments

dcObject	The original S4 class object containing the test statistics to be extracted.
permObject	The array of matrices containing the null test statistics.
secondMat	Logical, indicating whether a second matrix was used in the construction of this dcObject and permObject. If FALSE, the upper.tri of both are extracted to avoid double counting test statistics.
testSlot	The slot of the dcObject to be removed for use as the actual test statistic.
verbose	Whether summaries of the q-value operations should be reported.
plotFdr	Allows for plotting of fdrtool p-value adjustment result OR empirical FDR q-value adjustment technique, if either of these are chosen. Requires fdrtool package OR qvalue package. Default = FALSE.

Value

A list containing a vectof of empirical p-values and a vector of q-values, both of the same length as the original actual test statistics.

---

plotCors *Plot gene pair correlations in multiple conditions.*

Description

Takes the original input matrix, a design matrix, and two gene symbols to plot the corelation in the conditions specified.

Usage

```
plotCors(inputMat, design, compare, corrType = "pearson", geneA, geneB,
  oneRow = FALSE, smooth = TRUE, log = FALSE, ylab = NULL,
  xlab = NULL)
```

Arguments

inputMat	The matrix (or data.frame) of values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers whose correlations and differential correlations you are interested in analyzing, while the columns should correspond to the rows of the design matrix and should be separable into your groups.
design	A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see vignettes for more information.
compare	Vector of two character strings, each corresponding to one group name in the design matrix, that should be compared.

corrType	The correlation type of the analysis, limited to "pearson" or "spearman". Default = "pearson".
geneA	The first gene symbol.
geneB	The second gene symbol.
oneRow	Coerce all of the conditions to be plotted on the same row (as opposed to wrapping to multiple rows; relevant if there are >3 conditions).
smooth	Whether to perform lm-based smoothing of the trend in each condition and add this to the plot.
log	Logical, indicating whether the data should be log2-transformed prior to plotting (after adding a small constant of 0.5 to avoid problems with the log transform).
ylab	Override the y-axis label to one of your choice.
xlab	Override the x-axis label to one of your choice.

Value

A ggplot2 object that can be plotted, further modified, and/or saved.

---

plotGOOneGroup *Plot results from a hypergeometric enrichment test for one condition.*

Description

Uses ggplot2 to create a horizontal bar plot of the p-values (or odds-ratios) from enrichment tests of GO terms derived from differentially correlated gene sets (or any gene sets inputted into upstream functions). Note that the first column of each data frame is removed to allow for row binding, and otherwise the column names should match.

Usage

```
plotGOOneGroup(dfList, nTerms = 5, minSize = 50, maxSize = 500,
  dataCol = "Pvalue", namesCol = "Term", labelsCol = "Ontology",
  legendTitle = "GO Type", adjustPVals = FALSE)
```

Arguments

dfList	A named list of data frames corresponding to different GO term enrichments.
nTerms	The number of most-enriched terms to plot from each GO term type.
minSize	The number of genes above which a gene set should be removed from analysis (e.g., because it is so small as to be overly specific and untrustworthy).
maxSize	The number of genes above which a gene set should be removed from analysis (e.g., because it is so big as to be overly generic and relatively uninteresting).
dataCol	Column of the input matrix to be plotted in the bar plot. If "Pvalue", it will be -log10 transformed prior to plotting. If not "Pvalue", the x-axis label should be changed manually following the function call.

namesCol	The column specifying the name of the GO terms to be plotted.
labelsCol	The column specifying the fill labels of the GO terms to be plotted.
legendTitle	The title for the legend in the resulting plot.
adjustPVals	Logical, indicating whether or not p-values should be adjusted by the Benjamini-Hochberg method.

Value

An ggplot2 object that can be called in order to plot it, saved, or modified.

`plotGOTwoGroups` *Plot results from a hypergeometric enrichment test to compare two conditions.*

Description

Uses plotrix to create a pyramid plot of the odds-ratios from enrichment tests of GO terms derived from differentially correlated gene sets (or any gene sets inputted into upstream functions) in two conditions. Note that the first column of each data frame is removed to allow for row binding, and otherwise the column names should match.

Usage

```
plotGOTwoGroups(dfList1, dfList2, nTerms = 5, minSize = 40,
  maxSize = 1000, labelsCol = "Ontology", adjustPVals = TRUE,
  plotrix_gap = 20, GOTermTypes = c("BP", "CC", "MF"), pValCutoff = 0.05,
  filterSignificant = FALSE, filterSigThresh = 0.05,
  labels = c("Corr Class 1", "GO Term Name", "Corr Class 2"),
  fill_zero_cats = FALSE)
```

Arguments

dfList1	A named list of data frames corresponding to different GO term enrichments from the first condition or differential correlation class.
dfList2	A named list of data frames corresponding to different GO term enrichments from the second condition or differential correlation class.
nTerms	The number of most difference-in-enrichment (i.e., difference adjusted p-value for the difference in log OR) terms to plot from each GO term type.
minSize	The number of genes above which a gene set should be removed from analysis (e.g., because it is so small as to be overly specific and untrustworthy).
maxSize	The number of genes above which a gene set should be removed from analysis (e.g., because it is so big as to be overly generic and relatively uninteresting).
labelsCol	The column specifying the fill labels of the GO terms to be plotted.
adjustPVals	Logical, indicating whether or not p-values for the difference in log odds between conditions should be adjusted by the Benjamini-Hochberg method.

plotrix_gap	Parameter specifying the size of the gap between the two sides of the ORs.
GOTermTypes	Character vector for the GO term types to be plotted.
pValCutoff	p-Value cutoff to specify how significant each term enrichment must be in at least one of the groups to be considered for the comparison between the conditions. If no cutoff is desired, set to 1.
filterSignificant	Logical, indicating whether or not the p-value for the difference in ORs between groups should be required to be less than a particular threshold prior to downstream analysis.
filterSigThresh	Number indicating the threshold of the p-value for the difference in ORs between groups required if filterSignificant is TRUE.
labels	Character vector specifying labels for the pyramid plot; first entry = left label, second entry = middle label, third entry = right label.
fill_zero_cats	Logical, indicating whether counts of zero in some groups should be included in the comparisons. Adds 0.1 to the ORs and in the denominator of the count for the calculation of the SE in the cases where there are zero counts identified, as a convservative measure to prevent finding infinite differences between groups.

Value

A data frame with the values used to create the plot.

---

plotModuleGO *Plot extracted results from module-based GO enrichment analysis using ggplot2.*

Description

Takes a data frame of enrichment results in multiple modules and plots the results. Note that if a GO term enrichment does not exist for that module, it is set as 0 for an OR or 1 for a p-value.

Usage

```
plotModuleGO(df, nTerms = 5, termVector = NULL, modules = NULL,
  heatmapColor = NULL, plotOR = FALSE, axis_text_col = "black",
  axis_x_text_angle = 45, text_size = 10, guide_title = NULL,
  coord_flip = FALSE, adjust = TRUE)
```

Arguments

df	The data frame of term enrichments to be plotted.
nTerms	The number of terms for each module whose GO terms with the minimum enrichment p-values for that group should be plotted.
termVector	Optional character vector of GO term strings to plot, overriding other options.

modules	Optional, a list of module names to plot. Optional; if not inputted, all of the module names in the data frame will be used.
heatmapColor	Optional specification of the heatmap colors. If not specified, ?heat.colors will be used.
plotOR	Logical, indicating whether odds ratios should be plotted on the heatmap, instead of -log10 p-values (the default).
axis_text_col	Color of axis text.
axis_x_text_angle	Angle of x-axis text.
text_size	Text size of axes and legend in plot.
guide_title	Optionally, specify the title of legend.
coord_flip	Whether the coordinates should be flipped.
adjust	If p-values are plotted, whether or not the enrichment p-values from each module should be adjusted by the Benjamini-Hochberg method.

Value

A data frame summarizing the GO term enrichments from each group, with columns on the ordered by the minimum p-value for OR term enrichment in any group.

---

plotVals *Creates a dotplot of the overall values for an individual gene in multiple conditions.*

Description

Takes the original input matrix, a design matrix, and one gene symbols (row name of the original matrix) to plot its values in the conditions specified, using a dotplot, +/- a summary bar. Will remove NAs prior to plotting.

Usage

```
plotVals(inputMat, design, compare, gene, log = FALSE, ylab = NULL,
  xlab = NULL, add_summary_bar = TRUE, summary_bar = "mean",
  summary_width = 0.75, dotplot_width = 0.5, dotplot_size = NULL,
  dotplot_binwidth = NULL)
```

Arguments

inputMat	The matrix (or data.frame) of values (e.g., gene expression values from an RNA-seq or microarray study) that you are interested in analyzing. The rownames of this matrix should correspond to the identifiers whose values you are interested in analyzing, while the columns should correspond to the rows of the design matrix and should be separable into your groups.

design	A standard model.matrix created design matrix. Rows correspond to samples and colnames refer to the names of the conditions that you are interested in analyzing. Only 0's or 1's are allowed in the design matrix. Please see vignettes for more information.
compare	Vector of two character strings, each corresponding to one group name in the design matrix, that should be compared.
gene	The gene symbol (row identifier).
log	Logical, indicating whether the data should be log2-transformed prior to plotting (after adding a small constant of 0.5 to avoid problems with the log transform and stabilize the variance with respect to the mean).
ylab	Override the y-axis label to one of your choice.
xlab	Override the x-axis label to one of your choice.
add_summary_bar	Logical indicating whether to include a summary bar for each group.
summary_bar	If summary bar included, type of bar to use to calculate. Options = "mean", "median"
summary_width	Horizontal width of summary crossbar included.
dotplot_width	The width of the dots in the dotplot. See ?geom_dotplot for more information.
dotplot_size	The diameter of the dots in the dotplot. Affects the chart relative to the dotplot binwidth. If NULL, will be calculated automatically.
dotplot_binwidth	The binwidth size for the dots in the dotplot. If NULL, will be calculated automatically.

Value

A ggplot2 object that can be plotted, further modified, and/or saved.

`switchGenesToHGCN` *Switches a gene vector to cleaned HGNC symbols.*

Description

Where possible, switches a character vector of gene names to cleaned and updated HGNC symbols.

Usage

```
switchGenesToHGCN(gene_list)
```

Arguments

gene_list	Character vector of gene names.

Value

Character vector of cleaned gene names.

topDCGenes *Ranks genes by their total number of differentially correlated gene pairs.*

Description

Returns list of lists for the top differentially correlated gene pairs in each direction and/or class.

Usage

```
topDCGenes(ddcor_res, adjusted = FALSE, pval_gene_thresh = 0.05,
  geneNameCol = c("Gene1", "Gene2"), nGenes = "all", classes = TRUE)
```

Arguments

ddcor_res	The table of differential correlations outputted from ddcor. Expected to have pValDiff or pValDiff_adj columns as well as zScoreDiff, Gene1, +/- Classes columns.
adjusted	Logical indicating whether adjusted p-values from the differential correlation table (i.e., column "pValDiff_adj", when adjusted = TRUE) or unadjusted p-values (i.e., column "pValDiff", when adjusted = FALSE) should be used to subset the table into significant and non-significant portions.
pval_gene_thresh	p-value threshold to call a gene as having significant differential correlation or not.
geneNameCol	Character vector specifying the name of the columns that are used to extract the gene symbols. Note that the default is c("Gene1", "Gene2"), but this only makes sense in the context of a full DGCA experiment. In the case of a splitSet, you may want to use "Gene1" to avoid counting the splitSet names in all of the categories.
nGenes	Number of genes to display in the resulting table. Default = "all", but also can be restricted to a particular number.
classes	Gets the number of differentially correlated gene pairs associated with each of the differential correlation classes.

Value

A data frame with corresponding lists of genes most associated with each of the directions and/or correlation classes.

Examples

```
data(darmanis); data(design_mat); darmanis_subset = darmanis[1:30, ]
ddcor_res = ddcorAll(inputMat = darmanis_subset, design = design_mat,
 compare = c("oligodendrocyte", "neuron"))
top_genes = topDCGenes(ddcor_res)
```

Index