

Package ‘IntLIM’

October 12, 2022

Type Package

Title Integration of Omics Data Using Linear Modeling

Version 2.0.2

Description This workflow takes analyte levels from two different types of analytes (e.g. gene expression and metabolite abundance), meta-information on each analyte type, and sample outcome and metadata to identify analyte pairs that are significantly associated with a continuous or discrete outcome (e.g. drug response or tumor type). The following references describe the methods in this package: (1) Jalal K. Siddiqui, et al. (2018) <[doi:10.1186/s12859-018-2085-6](https://doi.org/10.1186/s12859-018-2085-6)>, (2) Andrew Patt, et al. (2019) <[doi:10.1007/978-1-4939-9027-6_23](https://doi.org/10.1007/978-1-4939-9027-6_23)>.

License GPL-2

Depends R (>= 3.2.0)

Imports ComplexHeatmap, DT, ggplot2, graphics, grDevices, heatmaply, highcharter, htmltools, KernSmooth, margins, methods, MASS, RColorBrewer, reshape2, rmarkdown, shiny, shinydashboard, shinyFiles, shinyjs, stats, testthat, utils

Suggests knitr,

Encoding UTF-8

NeedsCompilation no

Collate 'AllClasses.R' 'internalfunctions.R' '01_ReadData.R' '02_FilterData.R' '03_RunIntLim.R' '04_ProcessResults.R' 'IntLimResults_extendedfunctions.R' 'PermuteIntLIM.R' 'crossvalfunctions.R' 'plottingfunctions.R' 'runShiny.R' 'statsfunctions.R' 'writingfunctions.R'

VignetteBuilder knitr

RoxygenNote 7.1.2

Author Jalal Siddiqui [aut],
Shunchao Wang [aut],
Rohith Vanam [aut],
Elizabeth Baskin [aut],
Tara Eicher [aut, cre],
Kyle Spencer [aut],
Ewy Mathe [aut]

Maintainer Tara Eicher <tara.eicher@nih.gov>

Repository CRAN

Date/Publication 2022-08-22 15:10:04 UTC

R topics documented:

BuildDataAndLines	3
CreateCrossValFolds	4
DistPvalues	4
DistRSquared	5
FilterData	5
FilterDataFolds	6
getQuantileForInteractionCoefficient	7
getStatsAllLM	8
getstatsOneLM	9
HistogramPairs	9
InteractionCoefficientGraph	10
IntLimData-class	10
IntLimResults-class	11
MarginalEffectsGraph	12
MarginalEffectsGraphDataframe	12
multi.which	13
OutputData	14
OutputResults	14
PermutationCountSummary	15
PermutationPairSummary	15
PermuteIntLIM	16
PlotDistributions	17
PlotFoldOverlapUpSet	18
PlotPair	18
PlotPairFlat	19
PlotPCA	20
ProcessResults	21
ProcessResultsAllFolds	22
ProcessResultsContinuous	23
pvalCoefVolcano	23
PValueBoxPlots	24
ReadData	25
RemovePlusInCovars	26
RunCrossValidation	26
RunIntLim	28
RunIntLimAllFolds	29
runIntLIMApp	30
RunLM	30
ShowStats	31

Index

32

BuildDataAndLines	<i>A helper function for the PlotPair functions (i.e. the highcharter one and the flat, base-R one).</i>
-------------------	--

Description

A helper function for the PlotPair functions (i.e. the highcharter one and the flat, base-R one).

Usage

```
BuildDataAndLines(  
  inputData,  
  inputResults,  
  outcome,  
  independentVariable,  
  independentAnalyteOfInterest,  
  outcomeAnalyteOfInterest,  
  palette = "Set1",  
  stype  
)
```

Arguments

inputData	IntLimObject output of ReadData() or FilterData()
inputResults	Data frame with model results (output of ProcessResults())
outcome	'1' or '2' must be set as outcome/independent variable
independentVariable	'1' or '2' must be set as outcome/independent variable
independentAnalyteOfInterest	independent analyte in pair
outcomeAnalyteOfInterest	outcome analyte in pair
palette	choose an RColorBrewer palette ("Set1", "Set2", "Set3", "Pastel1", "Pastel2", "Paired", etc.) or submit a vector of colors
stype	Phenotype or outcome variable

`CreateCrossValFolds` *Creates multiple cross-validation folds from the data. Format is a list of `IntLIMData` training and testing pairs. The "training" slot contains all data except that in the given fold, and the "testing" contains all data in the fold.*

Description

Creates multiple cross-validation folds from the data. Format is a list of `IntLIMData` training and testing pairs. The "training" slot contains all data except that in the given fold, and the "testing" contains all data in the fold.

Usage

```
CreateCrossValFolds(inputData, folds)
```

Arguments

<code>inputData</code>	<code>IntLimData</code> object (output of <code>ReadData()</code>) with analyte levels and associated meta-data
<code>folds</code>	number of folds to create

Value

A set of `IntLimData` training and testing sets, of the following format: `list(list("train" = IntLimData, "test" = IntLimData), ... list("train" = IntLimData, "test" = IntLimData))`

List of `IntLimModel` objects with model results

`DistPvalues` *Visualize the distribution of unadjusted p-values from linear models*

Description

Visualize the distribution of unadjusted p-values from linear models

Usage

```
DistPvalues(IntLimResults, breaks = 100, adjusted = TRUE)
```

Arguments

<code>IntLimResults</code>	output of <code>RunIntLim()</code>
<code>breaks</code>	the number of breaks to use in histogram (see <code>hist()</code> documentation for more details)
<code>adjusted</code>	Whether or not to plot adjusted p-values. If <code>TRUE</code> (default), adjusted p-values are plotted. If <code>FALSE</code> , unadjusted p-values are plotted.

Value

No return value, called for side effects

DistRSquared	<i>Visualize the distribution of unadjusted p-values from linear models</i>
--------------	---

Description

Visualize the distribution of unadjusted p-values from linear models

Usage

```
DistRSquared(IntLimResults, breaks = 100)
```

Arguments

IntLimResults	output of RunIntLim()
breaks	the number of breaks to use in histogram (see hist() documentation for more details)

Value

No return value, called for side effects

FilterData	<i>Filter input data by abundance values and number of missing values.</i>
------------	--

Description

Filter data by abundance (with user-input percentile cutoff) of missing values (with user-input percent cutoff). Missing values are commonly found in metabolomics data.

Usage

```
FilterData(  
  inputData,  
  analyteType1perc = 0,  
  analyteType2perc = 0,  
  analyteMiss = 0,  
  suppressWarnings = FALSE,  
  cov.cutoff = 0  
)
```

Arguments

inputData	IntLimData object (output of ReadData()) with analyte levels and associated meta-data
analyteType1perc	percentile cutoff (0-1) for filtering analyte type 1 (e.g. remove analytes with mean values < 'analyteType1perc' percentile) (default: 0)
analyteType2perc	percentile cutoff (0-1) for filtering analyte type 2 (default: no filtering of analytes) (default:0)
analyteMiss	missing value percent cutoff (0-1) for filtering both analyte types (analytes with > 80% missing values will be removed) (default:0)
suppressWarnings	whether or not to print warnings. If TRUE, warnings will not be printed.
cov.cutoff	percentile cutoff (0-1) for the covariances of the analytes (default: 0.30)

Value

filtData IntLimData object with input data after filtering

FilterDataFolds	<i>Filter input data by abundance values (analyte data) and number of missing values.</i>
-----------------	---

Description

Filter data by abundance (with user-input percentile cutoff) of missing values (with user-input percent cutoff). Missing values are commonly found in metabolomics data.

Usage

```
FilterDataFolds(
  inputDataFolds,
  analyteType1perc = 0,
  analyteType2perc = 0,
  analyteMiss = 0,
  cov.cutoff = 0,
  suppressWarnings = FALSE
)
```

Arguments

inputDataFolds	List of IntLimData objects (output of ReadData()) with analyte levels and associated meta-data
analyteType1perc	percentile cutoff (0-1) for filtering analyte type 1 (e.g. remove analytes with mean values < 'analyteType1perc' percentile) (default: 0)

`analyteType2perc` percentile cutoff (0-1) for filtering analyte type 2 (default: no filtering of analytes) (default:0)

`analyteMiss` missing value percent cutoff (0-1) for filtering analytes (analytes with > 80% missing values will be removed) (default:0)

`cov.cutoff` percentile cutoff (0-1) for the covariances of the analytes (default: 0.30)

`suppressWarnings` whether to suppress warnings

Value

`filtData` `IntLimData` object with input data after filtering

`getQuantileForInteractionCoefficient`

Function that gets numeric cutoffs from percentile

Description

Function that gets numeric cutoffs from percentile

Usage

`getQuantileForInteractionCoefficient(tofilter, interactionCoeffPercentile)`

Arguments

`tofilter` dataframe for percentile filtering

`interactionCoeffPercentile` percentile cutoff for interaction coefficient (default bottom 10 percent (high negative coefficients) and top 10 percent (high positive coefficients))

Value

vector with numeric cutoffs

getStatsAllLM

Function that runs Linear Models for all analytes

Description

Function that runs Linear Models for all analytes

Usage

```
getStatsAllLM(
  outcome,
  independentVariable,
  type1,
  type2,
  type,
  covar,
  covarMatrix,
  continuous,
  save.covar.pvals,
  remove.tri = FALSE,
  suppressWarnings = FALSE
)
```

Arguments

outcome	'1' or '2' must be set as outcome/independent variable
independentVariable	'1' or '2' must be set as outcome/independent variable
type1	Analyte type 1 dataset
type2	Analyte type 2 dataset
type	vector of sample type (by default, it will be used in the interaction term). Only 2 categories are currently supported.
covar	vector of additional vectors to consider
covarMatrix	covariate matrix
continuous	indicate whether data is discrete (FALSE) or continuous (TRUE)
save.covar.pvals	boolean to indicate whether or not to save the p-values of all covariates, which can be analyzed later but will also lengthen computation time.
remove.tri	boolean to indicate whether or not to remove the 1-1 or 2-2 pair with the highest p-value across two duplicate models (e.g. m1~m2 and m2~m1)
suppressWarnings	whether or not to suppress warnings

Value

list of matrices (interaction.pvalues, interaction.adj.pvalues, interaction.coefficients)

getstatsOneLM	<i>Function that runs linear models for analyte vs. all analytes of the other type</i>
---------------	--

Description

Function that runs linear models for analyte vs. all analytes of the other type

Usage

```
getstatsOneLM(form, clindata, arraydata, analytename, suppressWarnings = FALSE)
```

Arguments

form	LM formulat (typically m~g+t+g:t)
clindata	data frame with 1st column: expression of one analyte; 2nd column sample type (e.g. cancer/non-cancer)
arraydata	matrix of analyte values
analytename	name of independent analyte in the model
suppressWarnings	whether or not to suppress warnings

HistogramPairs	<i>histogram of analyte pairs depending upon independent or outcome analyte</i>
----------------	---

Description

histogram of analyte pairs depending upon independent or outcome analyte

Usage

```
HistogramPairs(inputResults, type = "outcome", breaks = 50)
```

Arguments

inputResults	Data frame with model results (output of ProcessResults())
type	'independent' or 'outcome'. 'outcome' set as default
breaks	Number of breaks selected for histogram

Value

No return value, called for side effects

InteractionCoefficientGraph

Graphs a scatterplot of pairs vs. the interaction coefficient for the pair

Description

Graphs a scatterplot of pairs vs. the interaction coefficient for the pair

Usage

```
InteractionCoefficientGraph(
  inputResults,
  interactionCoeffPercentile = 0.1,
  percentageToPlot = 0.01,
  independent.var.type = 1,
  outcome = 2
)
```

Arguments

`inputResults` Data frame with model results (output of `ProcessResults()`)

`interactionCoeffPercentile` percentile cutoff for interaction coefficient (default bottom 10 percent (high negative coefficients) and top 10 percent (high positive coefficients))

`percentageToPlot` percentage of points to plot (the points will be randomly selected) – plotting all points will likely overwhelm plotting function.

`independent.var.type` type of analyte used as the independent variable ("1" or "2")

`outcome` type of analyte used as the outcome/dependent variable ("1" or "2")

Value

a scatterplot

IntLimData-class

IntLimData class

Description

IntLimData class

Slots

analyteType1 A matrix of abundance, expression, or other levels for a specific type of analyte (e.g. protein abundance, metabolite abundance, or gene expression)
 analyteType2 A second matrix of abundance, expression, or other levels for a specific type of analyte (e.g. protein abundance, metabolite abundance, or gene expression)
 analyteType1MetaData A data frame of metadata for analyte type 1.
 analyteType2MetaData A data frame of metadata for analyte type 2.
 sampleMetaData A data frame of covariate values from the patient data.

IntLimResults-class *IntLimResults class*

Description

IntLimResults class

Slots

interaction.pvalues matrix of interaction p-values
 interaction.adj.pvalues matrix of adjusted interaction pvalues
 interaction.coefficients matrix of interaction coefficients
 covariate.coefficients data frame of coefficients for each covariate
 covariate.pvalues data frame of p-values for each covariate
 model.rsquared matrix of r-squared values
 corr matrix of correlations in group 1 and 2
 filt.results data frame of filtered results
 warnings a message of whether analytes have 0 standard deviation
 stype column name that represents sample type (by default, it will be used in the interaction term).
 Only 2 categories are currently supported.
 outcome outcome is either '1' or '2'
 independent.var.type independent variable type (either '1 or '2')
 covar describing additional variables and the class they form
 continuous "1" if outcome is continuous, "0" if not

`MarginalEffectsGraph` *Creates a dataframe of the marginal effect of phenotype*

Description

Creates a dataframe of the marginal effect of phenotype

Usage

```
MarginalEffectsGraph(dataframe, title, ylab, xlab)
```

Arguments

<code>dataframe</code>	from <code>MarginalEffectsGraphDataframe</code>
<code>title</code>	for graph
<code>ylab</code>	outcome analyte in pair
<code>xlab</code>	independent analyte in pair

Value

values used for graphing

`MarginalEffectsGraphDataframe`
Creates a dataframe of the marginal effect of phenotype

Description

Creates a dataframe of the marginal effect of phenotype

Usage

```
MarginalEffectsGraphDataframe(  
  inputResults,  
  inputData,  
  independentAnalyteOfInterest,  
  outcomeAnalyteOfInterest,  
  continuous,  
  outcome,  
  independentVariable  
)
```

Arguments

inputResults IntLimResults object with model results (output of RunIntLim())

inputData Named list (output of FilterData()) with analyte levels and associated meta-data

independentAnalyteOfInterest
 independent analyte in pair

outcomeAnalyteOfInterest
 outcome analyte in pair

continuous whether or not the outcome is continuous (TRUE or FALSE)

outcome '1' or '2' must be set as outcome/independent variable

independentVariable
 '1' or '2' must be set as outcome/independent variable

Value

dataframe for further analysis

multi.which	<i>A which for multidimensional arrays. Mark van der Loo 16.09.2011</i>
-------------	---

Description

A which for multidimensional arrays. Mark van der Loo 16.09.2011

Usage

```
multi.which(A)
```

Arguments

A Boolean function defined over a matrix

Value

vector with numeric cutoffs

OutputData	<i>Output data into individual CSV files. All data will be zipped into one file with all data.</i>
------------	--

Description

Output data into individual CSV files. All data will be zipped into one file with all data.

Usage

```
OutputData(inputData, filename = "")
```

Arguments

inputData	data output from ReadData() or FilterData() function
filename	name of file to be output (default: 'tempdir/output.zip')

Value

the filename of the CSV file with results named with cohort

OutputResults	<i>Output results into a zipped CSV file. Results include gene and metabolite pairs, along with model interaction p-values, and correlations in each group being evaluated.</i>
---------------	---

Description

Output results into a zipped CSV file. Results include gene and metabolite pairs, along with model interaction p-values, and correlations in each group being evaluated.

Usage

```
OutputResults(inputResults, filename = "")
```

Arguments

inputResults	IntLimResults object with model results (output of ProcessResults())
filename	name of file to be output (default: 'tempdir/results.csv')

Value

the filename of the CSV file with results named with cohort

 PermutationCountSummary

Return the number of significant analytes and the number of permutations in which each analyte is significant. If plot = TRUE, show a box plot of number of significant analytes over permutations, overlaid with the number of significant analytes in the original data.

Description

Return the number of significant analytes and the number of permutations in which each analyte is significant. If plot = TRUE, show a box plot of number of significant analytes over permutations, overlaid with the number of significant analytes in the original data.

Usage

```
PermutationCountSummary(inputResults, permResults, plot)
```

Arguments

inputResults	Data frame with model results (output of ProcessResults())
permResults	An object of type PermutationResults (output of PermulateIntLIM())
plot	Whether or not to show the boxplot. Default is TRUE.

Value

A data frame that includes, for each permutation, the number of significant pairs and the number of unique analytes of each analyte type within those pairs

PermutationPairSummary

Return the number of significant analytes / pairs per permutation and the number of permutations in which each analyte is significant. If plot = TRUE, show a box plot of number of significant analytes over permutations, overlaid with the number of significant analytes in the original data.

Description

Return the number of significant analytes / pairs per permutation and the number of permutations in which each analyte is significant. If plot = TRUE, show a box plot of number of significant analytes over permutations, overlaid with the number of significant analytes in the original data.

Usage

```
PermutationPairSummary(inputResults, permResults, plot)
```

Arguments

inputResults	Data frame with model results (output of ProcessResults())
permResults	An object of type PermutationResults (output of PermuteIntLIM())
plot	Whether or not to show the boxplot. Default is TRUE.

Value

A data frame that includes each significant pair from the unpermuted data and the number of times that pair was significant in the permuted data.

PermuteIntLIM	<i>Run permutations of the IntLIM code to search for random cross-omic associations in dataset</i>
---------------	--

Description

This function allows users to test different permutations of the metadata with their analytes to ensure that any pairs being deemed significant by IntIM are not being suggested due to random chance, as is sometimes a problem in correlative associations.

Usage

```
PermuteIntLIM(
  data,
  stype = "",
  outcome = 1,
  independent.var.type = 1,
  covar = c(),
  save.covar.pvals = FALSE,
  continuous = FALSE,
  pvalcutoff = 0.05,
  interactionCoeffPercentile = 0,
  rsquaredCutoff = 0,
  num.permutations = 1,
  seed = 1
)
```

Arguments

data	IntLimData object (output of ReadData()) with analyte levels and associated sample meta-data
stype	column name that represents sample type (by default, it will be used in the interaction term). Only 2 categories are currently supported.
outcome	'1' or '2' must be set as outcome/independent variable (default is '1')
independent.var.type	'1' or '2' must be set as independent variable (default is '1')

covar	Additional variables from the phenotypic data that be integrated into linear model
save.covar.pvals	boolean to indicate whether or not to save the p-values of all covariates, which can be analyzed later but will also lengthen computation time. The default is FALSE.
continuous	boolean to indicate whether the data is continuous or discrete
pvalcutoff	FDR adjusted p-value cutoff for number of significant multi-omic pairs (default = 0.20)
interactionCoeffPercentile	Interaction coefficient cutoff for the IntLIM linear model (default = 0.10)
rsquaredCutoff	Cutoff for the R-squared values for the models as a quality control (default = 0.50)
num.permutations	Number of permutations to be ran (default = 1)
seed	set.seed paramter allowing for custom random number generation seeds

Value

List object with 1st slot populated with dataframe containing the R² values of the models, and number of significant pairs before and after p-value adjustment. The 2nd slot in the list contains a string vector of the IDs of the significant pairs.

PlotDistributions *Get some stats after reading in data*

Description

Get some stats after reading in data

Usage

```
PlotDistributions(inputData, viewer = TRUE, palette = "Set1")
```

Arguments

inputData	IntLimObject output of ReadData()
viewer	whether the plot should be displayed in the RStudio viewer (TRUE) or in Shiny/Knitr (FALSE)
palette	choose an RColorBrewer palette ("Set1", "Set2", "Set3", "Pastel1", "Pastel2", "Paired", etc.) or submit a vector of colors

Value

a highcharter object

`PlotFoldOverlapUpSet` *Makes an UpSet plot showing the filtered pairs of analytes found in each fold. This plot should only be made for cross-validation data.*

Description

Makes an UpSet plot showing the filtered pairs of analytes found in each fold. This plot should only be made for cross-validation data.

Usage

```
PlotFoldOverlapUpSet(inputResults)
```

Arguments

`inputResults` List of outputs of `ProcessResultsAllFolds()`, each of which is a list of `IntLIM-Results`.

Value

an UpSet plot

`PlotPair` *scatter plot of pairs (based on user selection)*

Description

scatter plot of pairs (based on user selection)

Usage

```
PlotPair(
  inputData,
  inputResults,
  outcome,
  independentVariable,
  independentAnalyteOfInterest,
  outcomeAnalyteOfInterest,
  palette = "Set1",
  viewer = TRUE
)
```

Arguments

inputData	IntLimObject output of ReadData() or FilterData()
inputResults	Data frame with model results (output of ProcessResults())
outcome	'1' or '2' must be set as outcome/independent variable
independentVariable	'1' or '2' must be set as outcome/independent variable
independentAnalyteOfInterest	independent analyte in pair
outcomeAnalyteOfInterest	outcome analyte in pair
palette	choose an RColorBrewer palette ("Set1", "Set2", "Set3", "Pastel1", "Pastel2", "Paired", etc.) or submit a vector of colors
viewer	whether the plot should be displayed in the RStudio viewer (TRUE) or in Shiny/Knitr (FALSE)

Value

No return value, called for side effects

PlotPairFlat	<i>scatter plot of pairs (based on user selection). This version does not use highcharter and instead plots a base R plot.</i>
--------------	--

Description

scatter plot of pairs (based on user selection). This version does not use highcharter and instead plots a base R plot.

Usage

```
PlotPairFlat(
  inputData,
  inputResults,
  outcome,
  independentVariable,
  independentAnalyteOfInterest,
  outcomeAnalyteOfInterest,
  palette = "Set1"
)
```

Arguments

inputData	IntLimObject output of ReadData() or FilterData()
inputResults	Data frame with model results (output of ProcessResults())
outcome	'1' or '2' must be set as outcome/independent variable
independentVariable	'1' or '2' must be set as outcome/independent variable
independentAnalyteOfInterest	independent analyte in pair
outcomeAnalyteOfInterest	outcome analyte in pair
palette	choose an RColorBrewer palette ("Set1", "Set2", "Set3", "Pastel1", "Pastel2", "Paired", etc.) or submit a vector of colors

Value

No return value, called for side effects

 PlotPCA

PCA plots of data for QC

Description

PCA plots of data for QC

Usage

```
PlotPCA(inputData, viewer = TRUE, stype = "", palette = "Set1")
```

Arguments

inputData	IntLimObject output of ReadData()
viewer	whether the plot should be displayed in the RStudio viewer (TRUE) or in Shiny/Knitr (FALSE)
stype	category to color-code by (can be more than two categories)
palette	choose an RColorBrewer palette ("Set1", "Set2", "Set3", "Pastel1", "Pastel2", "Paired", etc.) or submit a vector of colors

Value

a highcharter object

ProcessResults	<i>Retrieve significant pairs, based on adjusted p-values. For each pair that is statistically significant, calculate the correlation within group1 (e.g. cancer) and the correlation within group2 (e.g. non-cancer). Users can then remove pairs with a difference in correlations between groups 1 and 2 less than a user-defined threshold.</i>
----------------	---

Description

Retrieve significant pairs, based on adjusted p-values. For each pair that is statistically significant, calculate the correlation within group1 (e.g. cancer) and the correlation within group2 (e.g. non-cancer). Users can then remove pairs with a difference in correlations between groups 1 and 2 less than a user-defined threshold.

Usage

```
ProcessResults(
  inputResults,
  inputData,
  pvalcutoff = 0.05,
  interactionCoeffPercentile = 0,
  rsquaredCutoff = 0
)
```

Arguments

inputResults	IntLimResults object with model results (output of RunIntLim())
inputData	MultiDataSet object (output of ReadData()) with analyte levels and associated meta-data
pvalcutoff	cutoff of FDR-adjusted p-value for filtering (default 0.05)
interactionCoeffPercentile	percentile cutoff for interaction coefficient (default bottom 10 percent (high negative coefficients) and top 10 percent (high positive coefficients))
rsquaredCutoff	cutoff for lowest r-squared value

Value

IntResults object with model results (now includes correlations)

ProcessResultsAllFolds

Retrieve significant pairs, based on adjusted p-values, interaction coefficient percentile, and r-squared values. This is a wrapper for ProcessResults.

Description

Retrieve significant pairs, based on adjusted p-values, interaction coefficient percentile, and r-squared values. This is a wrapper for ProcessResults.

Usage

```
ProcessResultsAllFolds(
  inputResults,
  inputData,
  pvalcutoff = 0.05,
  interactionCoeffPercentile = 0.5,
  rsquaredCutoff = 0,
  treecuts = 0
)
```

Arguments

<code>inputResults</code>	List of IntLimResults object with model results (output of RunIntLimAllFolds())
<code>inputData</code>	List of MultiDataSet objects (output of CreateCrossValFolds()) with analyte levels and associated meta-data
<code>pvalcutoff</code>	cutoff of FDR-adjusted p-value for filtering (default 0.05)
<code>interactionCoeffPercentile</code>	percentile cutoff for interaction coefficient (default bottom 10 percent (high negative coefficients) and top 10 percent (high positive coefficients))
<code>rsquaredCutoff</code>	cutoff for lowest r-squared value
<code>treecuts</code>	user-selected number of clusters (of pairs) to cut the tree into

Value

List of IntResults object with model results (now includes correlations)

 ProcessResultsContinuous

Retrieve significant pairs (aka filter out nonsignificant pairs) based on value of analyte:type interaction coefficient from linear model

Description

Retrieve significant pairs (aka filter out nonsignificant pairs) based on value of analyte:type interaction coefficient from linear model

Usage

```
ProcessResultsContinuous(
  inputResults,
  interactionCoeffPercentile = 0.1,
  pvalCutoff = 0.05,
  rsquaredCutoff = 0
)
```

Arguments

`inputResults` IntLimResults object with model results: output of RunIntLim

`interactionCoeffPercentile` percentile cutoff for interaction coefficient default bottom 10 percent (high negative coefficients) and top 10 percent (high positive coefficients)

`pvalCutoff` cutoff of FDR-adjusted p-value for filtering (default 0.05)

`rsquaredCutoff` cutoff of R-squared value for filtering (default 0, no filtering)

Value

A data frame with the following columns for each pair of analytes: "Analyte1", "Analyte2", "interaction_coeff", "Pval", "FDRadjPval", and "rsquared". Optionally, coefficients for each covariate may also be included.

`pvalCoefVolcano` *'volcano' plot (difference in correlations vs p-values) of all pairs*

Description

'volcano' plot (difference in correlations vs p-values) of all pairs

Usage

```
pvalCoefVolcano(
  inputResults,
  inputData,
  nrpoints = 10000,
  pvalcutoff = 0.05,
  coefPercentileCutoff = 0.9
)
```

Arguments

<code>inputResults</code>	Data frame with model results (output of <code>ProcessResults()</code>)
<code>inputData</code>	Named list (output of <code>FilterData()</code>) with analyte levels and associated meta-data
<code>nrpoints</code>	number of points to be plotted in lowest density areas (see 'smoothScatter' documentation for more detail)
<code>pvalcutoff</code>	cutoff of FDR-adjusted p-value for filtering (default 0.05)
<code>coefPercentileCutoff</code>	cutoff of interaction coefficient percentile.

Value

a `smoothScatter` plot

<code>PValueBoxPlots</code>	<i>Visualize the distribution of unadjusted p-values for all covariates from linear models using a bar chart.</i>
-----------------------------	---

Description

Visualize the distribution of unadjusted p-values for all covariates from linear models using a bar chart.

Usage

```
PValueBoxPlots(IntLimResults)
```

Arguments

<code>IntLimResults</code>	output of <code>RunIntLim()</code>
----------------------------	------------------------------------

Value

No return value, called for side effects

ReadData	<i>Read in CSV file</i>
----------	-------------------------

Description

Read in CSV file

Usage

```
ReadData(  
  inputFile,  
  analyteType1id = "id",  
  analyteType2id = "id",  
  logAnalyteType1 = FALSE,  
  logAnalyteType2 = FALSE,  
  class.feats = list(),  
  suppressWarnings = FALSE  
)
```

Arguments

<code>inputFile</code>	input file in CSV format (see Description)
<code>analyteType1id</code>	name of column from Analyte Type 1 meta data to be used as id (required if an Analyte Type 1 meta data file is present, must match Analyte Type 1 data)
<code>analyteType2id</code>	name of column from Analyte Type 2 meta data to be used as id (required if an Analyte Type 2 meta data file is present, must match Analyte Type 2 data)
<code>logAnalyteType1</code>	whether or not to log values for Analyte Type 1(T/F)
<code>logAnalyteType2</code>	whether or not to log values for Analyte Type 2(T/F)
<code>class.feats</code>	class ("factor" or "numeric") for each covariate. The following format is required: <code>list(covar1="numeric", covar2="factor")</code>
<code>suppressWarnings</code>	whether or not to suppress warnings

Value

IntLimData object with input data

RemovePlusInCovars	<i>RemovePlusInCovars</i>
--------------------	---------------------------

Description

RemovePlusInCovars

Usage

```
RemovePlusInCovars(covar = c(), sampleDataColnames)
```

Arguments

covar vector of additional vectors to consider
 sampleDataColnames vector of column names, which is a superset of the covar vector.

Value

list containing two elements: new covariates and new column names.

RunCrossValidation	<i>Runs the cross-validation end-to-end using the following steps: 1. Create multiple cross-validation folds from the data. 2. Filter each fold using the filtering criteria applied to the entire dataset. 3. Run IntLIM for all folds. 4. Process the results for all folds.</i>
--------------------	--

Description

Runs the cross-validation end-to-end using the following steps: 1. Create multiple cross-validation folds from the data. 2. Filter each fold using the filtering criteria applied to the entire dataset. 3. Run IntLIM for all folds. 4. Process the results for all folds.

Usage

```
RunCrossValidation(
  inputData,
  folds,
  analyteType1perc = 0,
  analyteType2perc = 0,
  analyteMiss = 0,
  cov.cutoff = 0,
  stype = "",
  outcome = c(1),
  covar = c(),
  continuous = FALSE,
```

```

save.covar.pvals = FALSE,
independent.var.type = c(1),
remove.duplicates = FALSE,
pvalcutoff = 0.05,
interactionCoeffPercentile = 0,
rsquaredCutoff = 0,
treecuts = 0,
suppressWarnings = FALSE
)

```

Arguments

inputData	IntLimData object (output of ReadData()) with analyte levels and associated meta-data
fold	number of folds to create
analyteType1perc	percentile cutoff (0-1) for filtering analyte type 1 (e.g. remove analytes with mean values < 'analyteType1perc' percentile) (default: 0)
analyteType2perc	percentile cutoff (0-1) for filtering analyte type 2 (default: no filtering of analytes) (default:0)
analyteMiss	missing value percent cutoff (0-1) for filtering analytes (analytes with > 80% missing values will be removed) (default:0)
cov.cutoff	percentile cutoff (0-1) for the covariances of the analytes (default: 0.30)
stype	column name that represents sample type (by default, it will be used in the interaction term). Only 2 categories are currently supported.
outcome	list of outcomes to run. '1' or '2' must be set as outcome/independent variable (default is '1')
covar	Additional variables from the phenotypic data that be integrated into linear model
continuous	boolean to indicate whether the data is continuous or discrete
save.covar.pvals	boolean to indicate whether or not to save the p-values of all covariates, which can be analyzed later but will also lengthen computation time. The default is FALSE.
independent.var.type	list of independent variable types to run. '1' or '2' must be set as independent variable (default is '1')
remove.duplicates	boolean to indicate whether or not to remove the pair with the highest p-value across two duplicate models (e.g. m1~m2 and m2~m1)
pvalcutoff	cutoff of FDR-adjusted p-value for filtering (default 0.05)
interactionCoeffPercentile	percentile cutoff for interaction coefficient
rsquaredCutoff	cutoff for lowest r-squared value
treecuts	user-selected number of clusters (of pairs) to cut the tree into
suppressWarnings	whether to suppress warnings

Value

List of IntResults object with model results (now includes correlations)

RunIntLim

Run linear models and retrieve relevant statistics

Description

Run linear models and retrieve relevant statistics

Usage

```
RunIntLim(
  inputData,
  stype = "",
  outcome = 1,
  covar = c(),
  continuous = FALSE,
  save.covar.pvals = FALSE,
  independent.var.type = 1,
  remove.duplicates = FALSE,
  suppressWarnings = FALSE
)
```

Arguments

inputData	Named list (output of FilterData()) with analyte abundances, and associated meta-data
stype	column name that represents sample type (by default, it will be used in the interaction term). Only 2 categories are currently supported.
outcome	'1' or '2' must be set as outcome/independent variable (default is '1')
covar	Additional variables from the phenotypic data that be integrated into linear model
continuous	boolean to indicate whether the data is continuous or discrete
save.covar.pvals	boolean to indicate whether or not to save the p-values of all covariates, which can be analyzed later but will also lengthen computation time. The default is FALSE.
independent.var.type	'1' or '2' must be set as independent variable (default is '1')
remove.duplicates	boolean to indicate whether or not to remove the pair with the highest p-value across two duplicate models (e.g. m1~m2 and m2~m1)
suppressWarnings	whether or not to print warnings. If TRUE, do not print.

Value

IntLimResults object with model results

RunIntLimAllFolds *Run linear models for all data folds. This is a wrapper to RunIntLim.*

Description

Run linear models for all data folds. This is a wrapper to RunIntLim.

Usage

```
RunIntLimAllFolds(
  inputData,
  stype = "",
  outcome = 1,
  covar = c(),
  continuous = FALSE,
  save.covar.pvals = FALSE,
  independent.var.type = 1,
  remove.duplicates = FALSE,
  suppressWarnings = FALSE
)
```

Arguments

inputData	IntLimData object (output of ReadData()) with analyte levels and associated meta-data
stype	column name that represents sample type (by default, it will be used in the interaction term). Only 2 categories are currently supported.
outcome	'1' or '2' must be set as outcome/independent variable (default is '1')
covar	Additional variables from the phenotypic data that be integrated into linear model
continuous	boolean to indicate whether the data is continuous or discrete
save.covar.pvals	boolean to indicate whether or not to save the p-values of all covariates, which can be analyzed later but will also lengthen computation time. The default is FALSE.
independent.var.type	'1' or '2' must be set as independent variable (default is '1')
remove.duplicates	boolean to indicate whether or not to remove the pair with the highest p-value across two duplicate models (e.g. m1~m2 and m2~m1)
suppressWarnings	whether to suppress warnings

Value

List of IntLimModel objects with model results

runIntLIMApp	<i>run shiny app</i>
--------------	----------------------

Description

run shiny app

Usage

```
runIntLIMApp(port = "127.0.0.1")
```

Arguments

port set port

Value

No return value, called for side effects

RunLM	<i>Function that runs linear models and returns interaction p-values.</i>
-------	---

Description

Function that runs linear models and returns interaction p-values.

Usage

```
RunLM(
  uncommon,
  outcome = 1,
  independentVariable = 2,
  type = "",
  covar = c(),
  continuous = FALSE,
  save.covar.pvals = FALSE,
  keep.highest.pval = FALSE,
  suppressWarnings = FALSE
)
```

Arguments

<code>incommon</code>	Named list (output of <code>FilterData()</code>) with analyte levels, and associated meta-data
<code>outcome</code>	'1' or '2' must be set as outcome/independent variable (default is '1')
<code>independentVariable</code>	'1' or '2' must be set as outcome/independent variable
<code>type</code>	vector of sample type (by default, it will be used in the interaction term). Only 2 categories are currently supported.
<code>covar</code>	vector of additional vectors to consider
<code>continuous</code>	boolean to indicate whether the data is continuous or discrete
<code>save.covar.pvals</code>	boolean to indicate whether or not to save the p-values of all covariates, which can be analyzed later but will also lengthen computation time. (rather than interaction terms).
<code>keep.highest.pval</code>	boolean to indicate whether or not to remove the pair with the highest p-value across two duplicate models (e.g. <code>m1~m2</code> and <code>m2~m1</code>)
<code>suppressWarnings</code>	whether or not to suppress warnings.

 ShowStats

Get some stats after reading in data

Description

Get some stats after reading in data

Usage

```
ShowStats(IntLimObject)
```

Arguments

`IntLimObject` output of `ReadData()`

Value

data.frame with some # of samples, features, etc.

Index

BuildDataAndLines, [3](#)
CreateCrossValFolds, [4](#)
DistPvalues, [4](#)
DistRSquared, [5](#)
FilterData, [5](#)
FilterDataFolds, [6](#)
getQuantileForInteractionCoefficient,
[7](#)
getStatsAllLM, [8](#)
getstatsOneLM, [9](#)
HistogramPairs, [9](#)
InteractionCoefficientGraph, [10](#)
IntLimData-class, [10](#)
IntLimResults-class, [11](#)
MarginalEffectsGraph, [12](#)
MarginalEffectsGraphDataframe, [12](#)
multi.which, [13](#)
OutputData, [14](#)
OutputResults, [14](#)
PermutationCountSummary, [15](#)
PermutationPairSummary, [15](#)
PermuteIntLIM, [16](#)
PlotDistributions, [17](#)
PlotFoldOverlapUpSet, [18](#)
PlotPair, [18](#)
PlotPairFlat, [19](#)
PlotPCA, [20](#)
ProcessResults, [21](#)
ProcessResultsAllFolds, [22](#)
ProcessResultsContinuous, [23](#)
pvalCoefVolcano, [23](#)
PValueBoxPlots, [24](#)
ReadData, [25](#)
RemovePlusInCovars, [26](#)
RunCrossValidation, [26](#)
RunIntLim, [28](#)
RunIntLimAllFolds, [29](#)
runIntLIMApp, [30](#)
RunLM, [30](#)
ShowStats, [31](#)