Package 'TPP'

October 9, 2015

Type Package

Title Analyze thermal proteome profiling (TPP) experiments

Version 1.0.3

Author Dorothee Childs, Holger Franken, Mikhail Savitski and Wolfgang Huber

Maintainer Dorothee Childs <dorothee.childs@embl.de>

Depends R (>= 3.2.0), Biobase, ggplot2, openxlsx, plyr, gridExtra, grid

Imports VGAM, VennDiagram, reshape2, nls2, foreach, doParallel, parallel

Suggests BiocStyle, knitr, testthat

Description Analyze thermal proteome profiling (TPP) experiments with varying temperatures (TR) or compound concentrations (CCR).

License Artistic-2.0

VignetteBuilder knitr

biocViews Proteomics, MassSpectrometry

NeedsCompilation no

R topics documented:

`analyzeTPPCCR` *Analyze TPP-CCR experiment*

Description

Performs analysis of a TPP-CCR experiment by invoking routines for data import, data processing, normalization, curve fitting, and production of the result table.

Usage

```
analyzeTPPCCR(configTable, data = NULL, resultPath = NULL,
  idVar = "gene_name", fcStr = "rel_fc_", naStrs = c("NA", "n/d", "NaN",
  "<NA>"), qualColName = "qupm", normalize = TRUE,
  ggplotTheme = tppDefaultTheme(), nonZeroCols = "qssm", r2Cutoff = 0.8,
  fcCutoff = 1.5, slopeBounds = c(1, 50), plotCurves = TRUE)
```

Arguments

`configTable`	dataframe, or character object with the path to a file, that specifies important details of the TPP-CCR experiment. See Section `details` for instructions how to create this object.
`data`	single dataframe, containing fold change measurements and additional annotation columns to be imported. Can be used instead of specifying the file path in the `configTable` argument.
`resultPath`	location where to store dose-response curve plots and results table.
`idVar`	character string indicating which data column provides the unique identifiers for each protein.

fcStr	character string indicating which columns contain the actual fold change values. Those column names containing the suffix fcStr will be regarded as containing fold change values.
naStrs	character vector indicating missing values in the data table. When reading data from file, this value will be passed on to the argument na.strings in function read.delim.
qualColName	character string indicating which column can be used for additional quality criteria when deciding between different non-unique protein identifiers.
normalize	perform median normalization (default: TRUE).
ggplotTheme	ggplot theme for dose response curve plots.
nonZeroCols	character string indicating a column that will be used for filtering out zero values.
r2Cutoff	Quality criterion on dose response curve fit.
fcCutoff	Cutoff for highest compound concentration fold change.
slopeBounds	Bounds on the slope parameter for dose response curve fitting.
plotCurves	boolan value indicating whether dose response curves should be plotted. Deactivating plotting decreases runtime.

Details

Invokes the following steps:

1. Import data using the tppccrImport function.
2. Perform normalization by fold change medians (optional) using the tppccrNormalize function. To perform normalization, set argument normalize=TRUE.
3. Fit and analyse dose response curves using the tppccrCurveFit function.
4. Export results to Excel using the tppExport function.

The default settings are tailored towards the output of the python package isobarQuant, but can be customised to your own dataset by the arguments idVar, fcStr, naStrs, qualColName.

If resultPath is not specified, the location of the input file specified in configTable will be used. If the input data are not specified in configTable, no result path will be set. This means that no output files or dose response curve plots are produced and analyzeTPPCCR just returns the results as a data frame.

The function analyzeTPPCCR reports intermediate results to the command line. To suppress this, use suppressMessages.

Value

A data frame in which the fit results are stored row-wise for each protein.

References

Savitski, M. M., Reinhard, F. B., Franken, H., Werner, T., Savitski, M. F., Eberhard, D., ... & Drewes, G. (2014). Tracking cancer drugs in living cells by thermal profiling of the proteome. Science, 346(6205), 1255784.

See Also

tppDefaultTheme

Examples

```
data(hdacCCR_smallExample)
tppccrResults <- analyzeTPPCCR(configTable=hdacCCR_config_repl1,
                               data=hdacCCR_data_repl1)
```

`analyzeTPPTR` *Analyze TPP-TR experiment*

Description

Performs analysis of a TPP-TR experiment by invoking routines for data import, data processing, normalization, curve fitting, and production of the result table.

Usage

```
analyzeTPPTR(configTable, data = NULL, resultPath = NULL,
  idVar = "gene_name", fcStr = "rel_fc_", naStrs = c("NA", "n/d", "NaN",
  "<NA>"), qualColName = "qupm", normalize = TRUE,
  normReqs = tpptrDefaultNormReqs(), ggplotTheme = tppDefaultTheme(),
  nCores = "max", startPars = c(Pl = 0, a = 550, b = 10),
  maxAttempts = 500, binWidth = 300, plotCurves = TRUE)
```

Arguments

`configTable`	dataframe, or character object with the path to a file, that specifies important details of the TPP-CCR experiment. See Section `details` for instructions how to create this object.
`data`	single dataframe, or list of dataframes, containing fold change measurements and additional annotation columns to be imported. Can be used instead of specifying the file path in the `configTable` argument.
`resultPath`	location where to store melting curve plots, intermediate results, and the final results table.
`idVar`	character string indicating which data column provides the unique identifiers for each protein.
`fcStr`	character string indicating which columns contain the actual fold change values. Those column names containing the suffix `fcStr` will be regarded as containing fold change values.
`naStrs`	character vector indicating missing values in the data table. When reading data from file, this value will be passed on to the argument `na.strings` in function `read.delim`.
`qualColName`	character string indicating which column can be used for additional quality criteria when deciding between different non-unique protein identifiers.

normalize	perform normalization (default: TRUE).
normReqs	list of filtering criteria for construction of the normalization set.
ggplotTheme	ggplot theme for melting curve plots.
nCores	either a numerical value given the desired number of CPUs, or 'max' to automatically assign the maximum possible number (default).
startPars	start values for the melting curve parameters. Will be passed to function nls for curve fitting.
maxAttempts	maximal number of curve fitting attempts if model does not converge.
binWidth	bin width used for p-value computation.
plotCurves	boolan value indicating whether melting curves should be plotted. Deactivating plotting decreases runtime.

Details

Invokes the following steps:

1. Import data using the tpptrImport function.
2. Perform normalization (optional) using the tpptrNormalize function. To perform normalization, set argument normalize=TRUE. The normalization will be filtered according to the criteria specified in the normReqs argument (also see the documentation of tpptrNormalize and tpptrDefaultNormReqs for further information.
3. Fit melting curves using the tpptrCurveFit function.
4. Produce result table using the tpptrResultTable function.
5. Export results to Excel using the tppExport function.

The default settings are tailored towards the output of the python package isobarQuant, but can be customised to your own dataset by the arguments idVar, fcStr, naStrs, qualColName.

If resultPath is not specified, the location of the first input file specified in configTable will be used. If the input data are not specified in configTable, no result path will be set. This means that no output files or melting curve plots are produced and analyzeTPPTR just returns the results as a data frame.

The function analyzeTPPTR reports intermediate results to the command line. To suppress this, use suppressMessages.

The argument nCores could be either 'max' (use all available cores) or an upper limit of CPUs to be used.

The melting curve plots will be stored in a subfolder with name Melting_Curves at the location specified by resultPath.

If the melting curve fitting procedure does not converge, it will be repeatedly started from perturbed starting parameters (maximum iterations defined by argument maxAttempts).

Value

A data frame in which the fit results are stored row-wise for each protein.

References

Savitski, M. M., Reinhard, F. B., Franken, H., Werner, T., Savitski, M. F., Eberhard, D., ... & Drewes, G. (2014). Tracking cancer drugs in living cells by thermal profiling of the proteome. Science, 346(6205), 1255784.

See Also

tppDefaultTheme

Examples

```
data(hdacTR_smallExample)
tpptrResults <- analyzeTPPTR(configTable=hdacTR_config, data=hdacTR_data, nCores=1)
```

---

hdacCCR_config_repl1 *The configuration table to analyze hdacCCR_data_repl1.*

Description

The configuration table to analyze hdacCCR_data_repl1.

Details

`hdacCCR_config_repl1` is a data frame that specifies the experiment name, isobaric labels, and the administered drug concentrations at each label.

---

hdacCCR_config_repl2 *The configuration table to analyze hdacCCR_data_repl2.*

Description

The configuration table to analyze hdacCCR_data_repl2.

Details

`hdacCCR_config_repl2` is a data frame that specifies the experiment name, isobaric labels, and the administered drug concentrations at each label.

hdacCCR_data_repl1 *TPP-CCR example dataset (replicate 1)*

Description

Example subset of a Panobinostat TPP-CCR dataset (replicate 1)

Details

A subset of a dataset obtained by TPP-CCR experiments to investigate drug effects for HDAC inhibitor Panobinostat. It contains 7 HDACs as well as a random selection of 493 further proteins.

You can use this dataset to explore the TPP package functionalities without invoking the whole time consuming analysis on the big dataset.

The originial dataset is located in the folder `'example_data/CCR_example_data'` in the package's installation directory. You can find it on your system by the R command `system.file('example_data', package = 'TPP')`

hdacCCR_data_repl2 *TPP-CCR example dataset (replicate 2)*

Description

Example subset of a Panobinostat TPP-CCR dataset (replicate 2)

Details

A subset of a dataset obtained by TPP-CCR experiments to investigate drug effects for HDAC inhibitor Panobinostat. It contains 7 HDACs as well as a random selection of 493 further proteins.

You can use this dataset to explore the TPP package functionalities without invoking the whole time consuming analysis on the big dataset.

The originial dataset is located in the folder `'example_data/CCR_example_data'` in the package's installation directory. You can find it on your system by the R command `system.file('example_data', package = 'TPP')`

hdacCCR_smallExample *Example subsets of a Panobinostat TPP-CCR dataset (replicates 1 and 2) and the corresponding configuration tables to start the analysis.*

Description

Example subsets of a Panobinostat TPP-CCR dataset (replicates 1 and 2) and the corresponding configuration tables to start the analysis.

`hdacTR_config` *The configuration table to analyze hdacTR_data.*

Description

The configuration table to analyze hdacTR_data.

Details

`hdacTR_config` is a data frame that specifies the experiment name, isobaric labels, and the administered drug concentrations at each label.

`hdacTR_data` *TPP-TR example dataset.*

Description

Example subset of a dataset obtained by TPP-TR experiments to investigate possible targets for HDAC inhibitor Panobinostat.

Details

`hdacTR_data` is a list of data frames that contain measurements for HDACs as well as a random selection of 500 further proteins.

You can use this dataset to explore the `TPP` package functionalities without invoking the whole time consuming analysis on the whole dataset.

The originial dataset is located in the folder `'example_data/TR_example_data'` in the package's installation directory. You can find it on your system by the `R` command `system.file('example_data', package = 'TPP')`

`hdacTR_fittedData_smallExample`
Example of a TPP-TR dataset with fitted melting curve parameters.

Description

Example of a TPP-TR dataset with fitted melting curve parameters.

Details

Contains the data object `trDataFitted`.

hdacTR_resultsTable_smallExample *Example of a TPP-TR result table.*

Description

Example of a TPP-TR result table.

Details

Contains the data object `resultTable`.

hdacTR_smallExample *Example subset of a Panobinostat TPP-TR dataset and the corresponding configuration table to start the analysis.*

Description

Example subset of a Panobinostat TPP-TR dataset and the corresponding configuration table to start the analysis.

resultTable *Example of a TPP-TR result table.*

Description

Example of a TPP-TR result table.

Details

`resultTable` is a data frame that contains the measurements of several TPP-TR experiments, the fitted melting curve parameters, as well as p-values and the results of additional quality checks for each protein. It can be used as input for the function `tppQCPlotsCorrelateExperiments`.

TPP *Thermal proteome profiling (TPP)*

Description

TPP is a toolbox for analyzing thermal proteome profiling (TPP) experiments.

Usage

```
.onLoad(libname, pkgname)
```

Arguments

libname	a character string giving the library directory where the package defining the namespace was found. Passed to .onLoad function.
pkgname	a character string giving the name of the package. Passed to .onLoad function.

Details

In order to start a TPP-TR analysis, use function analyzeTPPTR. For a TPP-CCR analysis, use function analyzeTPPCCR. See the vignette for detailed instructions.

Value

No return value defined for this document.

References

Savitski, M. M., Reinhard, F. B., Franken, H., Werner, T., Savitski, M. F., Eberhard, D., ... & Drewes, G. (2014). Tracking cancer drugs in living cells by thermal profiling of the proteome. Science, 346(6205), 1255784.

tppccrCurveFit *Fit and analyse dose response curves*

Description

tppccrCurveFit fits logistic dose response curves to fold change measurements of a TPP-CCR experiment and returns quality information about the estimated parameters.

Usage

```
tppccrCurveFit(data, resultPath = NULL, ggplotTheme = tppDefaultTheme(),
  doPlot = TRUE, fcCutoff = 1.5, r2Cutoff = 0.8, slopeBounds = c(1, 50))
```

Arguments

data	ExpressionSet containing protein fold changes for curve fitting.
resultPath	location where to store dose-response curve plots and result table.
ggplotTheme	ggplot theme for dose response curve plots.
doPlot	boolan value indicating whether dose response curves should be plotted. Deactivating plotting decreases runtime.
fcCutoff	cutoff for highest compound concentration fold change.
r2Cutoff	quality criterion on dose response curve fit.
slopeBounds	bounds on the slope parameter for dose response curve fitting.

Details

data is an ExpressionSet object created by tppccrImport. It contains the isobaric labels and administered drug concentrations in the phenoData and user-defined protein properties in the featureData. Protein IDs are stored in the featureNames.

The dose response curve plots will be stored in a subfolder with name DoseResponse_Curves at the location specified by resultPath.

Value

A data frame in which the fit results are stored row-wise for each protein.

See Also

tppccrImport, tppDefaultTheme

Examples

```
data(hdacCCR_smallExample)
tppccrData <- tppccrImport(configTable=hdacCCR_config_repl1,
                           data=hdacCCR_data_repl1)
tppccrNorm <- tppccrNormalize(data=tppccrData)
tppccrTransformed <- tppccrTransform(data=tppccrNorm)
tppccrFitResults <- tppccrCurveFit(data=tppccrTransformed)
subset(tppccrFitResults, passed_filter)
```

---

tppccrImport *Import TPP-CCR dataset for analysis by the TPP package.*

Description

tppccrImport imports a table of protein fold changes and stores them in an ExpressionSet for use in the TPP package.

Usage

```
tppccrImport(configTable, data = NULL, idVar = "gene_name",
  fcStr = "rel_fc_", naStrs = c("NA", "n/d", "NaN", "<NA>"),
  qualColName = "qupm", nonZeroCols = "qssm")
```

Arguments

configTable	either a dataframe or the path to a spreadsheet. In both cases it specifies necessary information of the TPP-CCR experiment.
data	dataframe containing fold change measurements and additional annotation columns to be imported. Can be used instead of specifying the file path in configTable.
idVar	character string indicating which data column provides the unique identifiers for each protein.
fcStr	character string indicating which columns contain the actual fold change values. Those column names containing the suffix fcStr will be regarded as containing fold change values.
naStrs	character vector indicating missing values in the data table. When reading data from file, this value will be passed on to the argument na.strings in function read.delim.
qualColName	character string indicating which column can be used for additional quality criteria when deciding between different non-unique protein identifiers.
nonZeroCols	character string indicating a column that will be used for filtering out zero values.

Details

The imported dataset has to contain measurements obtained by a TPP-CCR experiment. Fold changes need to be pre-computed using the lowest concentration as reference.

The dataset can be specified by filename in the configTable argument, or given directly in the data argument

The default settings are adjusted to analyse data of the pyhton package isobarQuant. You can also customise them for your own dataset.

The configTable argument is a dataframe, or the path to a spreadsheet (tab-delimited text-file or xlsx format). Information about each experiment is stored row-wise. It contains the following columns:

- Path: location of the datafile. Alternatively, data can be directly handed over by the data argument.
- Experiment: unique experiment name.
- Label columns: each isobaric label names a column that contains the concentration administered for the label in the individual experiments.

During data import, proteins with NAs in the data column specified by idVar receive unique generic IDs so that they can be processed by the package.

Value

ExpressionSet object storing the measured fold changes, as well as row and column metadata. In each ExpressionSet `S`, the fold changes can be accessed by `exprs(S)`. Protein expNames can be accessed by `featureNames(S)`. TMT labels and the corresponding temperatures are returned by `S$labels` and `S$temperatures`.

See Also

`tpptrImport`, `tppccrCurveFit`

Examples

```
data(hdacCCR_smallExample)
tppccrData <- tppccrImport(configTable=hdacCCR_config_repl1, data = hdacCCR_data_repl1)
```

---

tppccrNormalize *Normalize data from TPP-CCR experiments*

Description

Normalize each fold change column by its median.

Usage

```
tppccrNormalize(data)
```

Arguments

`data`	expressionSet with measurements to be normalized

Value

ExpressionSet object storing the normalized fold changes, as well as row and column metadata. In each ExpressionSet `S`, the fold changes can be accessed by `exprs(S)`. Protein expNames can be accessed by `featureNames(S)`. TMT labels and the corresponding temperatures are returned by `S$labels` and `S$temperatures`.

Examples

```
data(hdacCCR_smallExample)
tppccrData <- tppccrImport(configTable=hdacCCR_config_repl1, data = hdacCCR_data_repl1)
tppccrNorm <- tppccrNormalize(data=tppccrData)
head(exprs(tppccrNorm))
```

tppccrTransform *Transform fold changes of TPP-CCR experiment*

Description

Transform fold changes of TPP-CCR experiment to prepare them for dose response curve fitting.

Usage

```
tppccrTransform(data, fcCutoff = 1.5)
```

Arguments

data	expressionSet object containing the data to be transformed.
fcCutoff	Cutoff for highest compound concentration fold change.

Value

ExpressionSet object storing the transformed fold changes, as well as row and column metadata. In each ExpressionSet S, the fold changes can be accessed by `exprs(S)`. Protein expNames can be accessed by `featureNames(S)`. TMT labels and the corresponding temperatures are returned by `S$labels` and `S$temperatures`.

Examples

```
data(hdacCCR_smallExample)
tppccrData <- tppccrImport(configTable=hdacCCR_config_repl1, data = hdacCCR_data_repl1)
tppccrNorm <- tppccrNormalize(data=tppccrData)
tppccrTransformed <- tppccrTransform(data=tppccrNorm)
determinedEffectTypes <- featureData(tppccrTransformed)$CompoundEffect
transformedData <- data.frame(exprs(tppccrTransformed),
                              Type=determinedEffectTypes)
```

tppDefaultTheme *Default ggplot theme for melting curve plots.*

Description

Default theme to be passed to the gplots produced by the TPP package.

Usage

```
tppDefaultTheme()
```

Details

Internally, the theme is used as an argument for the function `ggplot2::theme_set` in order specify the appearance of the melting curve plots.

The specified plot properties include bold font and increased font size for axis labels and title, as well as a 90 degree angle for y axis labels.

Value

ggplot theme with default settings for melting plot apparance.

Examples

```
# Import data:
data(hdacTR_smallExample)
tpptrData <- tpptrImport(configTable=hdacTR_config, data=hdacTR_data)
# Obtain template with default settings:
normRequirements <- tpptrDefaultNormReqs()
print(normRequirements)
# Relax filter on the 10th fold change column for
# normalization set production:
normRequirements$fcRequirements[3,3] <- 0.25
# Perform normalization:
tpptrNorm <- tpptrNormalize(data=tpptrData, normReqs=)
```

---

tppExport *Produce Excel table of TPP-TR or TPP-CCR experiment.*

Description

Produce Excel table of TPP-TR or TPP-CCR experiment.

Usage

```
tppExport(tab, file)
```

Arguments

`tab`	Table with results of the TPP analysis.
`file`	path for storing results table

Value

No value returned.

Examples

```
data(hdacTR_resultsTable_smallExample)
tppExport(resultTable, "tpptr_example_results.xlsx")
```

`tppQCPlotsCorrelateExperiments` *Plot pairwise relationships between the proteins in different TPP experiments.*

Description

Plot pairwise relationships between the proteins in different TPP experiments.

Usage

```
tppQCPlotsCorrelateExperiments(tppData, annotStr = "", path = NULL,
  ggplotTheme = tppDefaultTheme())
```

Arguments

`tppData`	List of expressionSets with data to be plotted.
`annotStr`	String with additional information to be added to the plot.
`path`	Location where to store resulting plot.
`ggplotTheme`	ggplot theme for the created plots.

Value

List of plots for each experiment.

See Also

`tppDefaultTheme`

Examples

```
data(hdacTR_smallExample)
tpptrData <- tpptrImport(configTable=hdacTR_config, data=hdacTR_data)
# Quality control (QC) plots BEFORE normalization:
tppQCPlotsCorrelateExperiments(tppData=tpptrData,
annotStr="Non-normalized Fold Changes", ggplotTheme=ggplotTheme)
# Quality control (QC) plots AFTER normalization:
tpptrNorm <- tpptrNormalize(data=tpptrData, normReqs=tpptrDefaultNormReqs())
tpptrDataNormalized <- tpptrNorm$normData
tppQCPlotsCorrelateExperiments(tppData=tpptrDataNormalized,
annotStr="Normalized Fold Changes", ggplotTheme=ggplotTheme)
```

tpptrCurveFit *Fit melting curves to all proteins in a dataset.*

Description

Fit melting curves to all proteins in a dataset.

Usage

```
tpptrCurveFit(data, resultPath = NULL, ggplotTheme = tppDefaultTheme(),
  doPlot = TRUE, startPars = c(Pl = 0, a = 550, b = 10),
  maxAttempts = 500, nCores = "max")
```

Arguments

data	list of ExpressionSets with protein fold changes for curve fitting.
resultPath	location where to store the melting curve plots.
ggplotTheme	ggplot theme for melting curve plots.
doPlot	boolan value indicating whether melting curves should be plotted, or whether just the curve parameters should be returned.
startPars	start values for the melting curve parameters. Will be passed to function nls for curve fitting.
maxAttempts	maximal number of curve fitting attempts if model does not converge.
nCores	either a numerical value given the desired number of CPUs, or 'max' to automatically assign the maximum possible number (default).

Details

If the melting curve fitting procedure does not converge, it will be repeatedly started from perturbed starting parameters (maximum iterations defined by argument maxAttempts)

Value

A list of ExpressionSets storing the data together with the melting curve parameters for each treatment condition and biological replicate. Each ExpressionSet contains the measured fold changes, as well as row and column metadata. In each ExpressionSet S, the fold changes can be accessed by exprs(S). Protein expNames can be accessed by featureNames(S). TMT labels and the corresponding temperatures are returned by S$labels and S$temperatures.

See Also

tppDefaultTheme

Examples

```
data(hdacTR_smallExample)
tpptrData <- tpptrImport(configTable=hdacTR_config, data=hdacTR_data)
tpptrNorm <- tpptrNormalize(data=tpptrData, normReqs=tpptrDefaultNormReqs())
normalizedData <- tpptrNorm$normData
hdacSubsets <- lapply(normalizedData,
                      function(d) d[grepl("HDAC", featureNames(d))])
tpptrFittedHDACs <- tpptrCurveFit(hdacSubsets, nCores=1)
```

---

tpptrDefaultNormReqs *Default filter criteria for fold change normalization*

Description

Filter criteria as described in the publication.

Usage

```
tpptrDefaultNormReqs()
```

Value

List with two entries: 'fcRequirements' describes filtering requirements on fold change columns, 'otherRequirements' contains criteria on additional metadata columns.

Examples

```
data(hdacTR_smallExample)
tpptrData <- tpptrImport(configTable=hdacTR_config, data=hdacTR_data)
tpptrNorm <- tpptrNormalize(data=tpptrData, normReqs=tpptrDefaultNormReqs())
```

---

tpptrImport *Import TPP-TR datasets for analysis by the TPP package.*

Description

`tpptrImport` imports several tables of protein fold changes and stores them in a list of ExpressionSets for use in the TPP package.

Usage

```
tpptrImport(configTable, data = NULL, idVar = "gene_name",
  fcStr = "rel_fc_", naStrs = c("NA", "n/d", "NaN"), qualColName = "qupm")
```

Arguments

configTable	either a dataframe or the path to a spreadsheet. In both cases it specifies necessary information of the TPP-CCR experiment.
data	single dataframe, or list of dataframes, containing fold change measurements and additional annotation columns to be imported. Can be used instead of specifying the file path in configTable.
idVar	character string indicating which data column provides the unique identifiers for each protein.
fcStr	character string indicating which columns contain the actual fold change values. Those column names containing the suffix fcStr will be regarded as containing fold change values.
naStrs	character vector indicating missing values in the data table. When reading data from file, this value will be passed on to the argument na.strings in function read.delim.
qualColName	character string indicating which column can be used for additional quality criteria when deciding between different non-unique protein identifiers.

Details

The imported datasets have to contain measurements obtained by TPP-TR experiments. Fold changes need to be pre-computed using the lowest temperature as reference.

An arbitrary number of datasets can be specified by filename in the Path-column of the configTable argument, or given directly as a list of dataframes in the data argument. They can differ, for example, by biological replicate or by experimental condition (for example, treatment versus vehicle). Their names are defined uniquely by the Experiment column in configTable. If this argument is not specified, generic names will be assigned. Replicates and experimental conditions can be specified by optional columns in configTable.

The default settings are adjusted to analyse data of the pyhton package isobarQuant. You can also customise them for your own dataset.

The configTable argument is a dataframe, or the path to a spreadsheet (tab-delimited text-file or xlsx format). Information about each experiment is stored row-wise. It contains the following columns:

- Path:location of each datafile. Alternatively, data can be directly handed over by the data argument.
- Experiment: unique experiment names.
- Condition: experimental conditions of each dataset.
- Replicate:experimental replicates of each dataset.
- Label columns: each isobaric label names a column that contains the temperatures administered for the label in the individual experiments.

Proteins with NAs in the data column specified by idVar receive unique generic IDs so that they can be processed by the package.

Value

A list of ExpressionSets storing the imported data for each treatment condition and biological replicate. Each ExpressionSet contains the measured fold changes, as well as row and column metadata. In each ExpressionSet `S`, the fold changes can be accessed by `exprs(S)`. Protein expNames can be accessed by `featureNames(S)`. TMT labels and the corresponding temperatures are returned by `S$labels` and `S$temperatures`.

See Also

`tppccrImport`

Examples

```
data(hdacTR_smallExample)
tpptrData <- tpptrImport(configTable=hdacTR_config, data=hdacTR_data)
```

---

tpptrNormalize *Normalize protein fold changes*

Description

Normalizes fold changes determined by TPP-TR experiments over different experimental groups.

Usage

```
tpptrNormalize(data, normReqs = tpptrDefaultNormReqs(),
  qcPlotTheme = tppDefaultTheme(), qcPlotPath = NULL, startPars = c(Pl =
  0, a = 550, b = 10), maxAttempts = 1)
```

Arguments

`data`	List of `ExpressionSets` with protein fold changes to be normalized.
`normReqs`	List of filtering criteria for construction of the normalization set.
`qcPlotTheme`	ggplot theme for the created plots
`qcPlotPath`	location where plots of the curves fitted to the normalization set medians should be stored.
`startPars`	start values for the melting curve parameters. Will be passed to function `nls` for curve fitting.
`maxAttempts`	maximal number of curve attempts to fit melting curve to fold change medians when computing normalization factors.

Details

Performs normalization of all fold changes in a given list of ExpressionSets. The normalization procedure is described in detail in Savitski et al. (2014). Whether normalization needs to be performed and what method is best suited depends on the experiment. Here we provide a reasonable solution for the data at hand.

We distinguish between filtering conditions on fold changes and on additional annotation columns. Correspondingly, `normReqs` contains two fields, `fcFilters` and `otherFilters`. Each entry contains a data frame with three columns specifying the column to be filtered, as well as upper and lower bounds. An example is given by `tpptrDefaultNormReqs`.

Value

A list of ExpressionSets storing the normalized data for each treatment condition and biological replicate. Each ExpressionSet contains the measured fold changes, as well as row and column metadata. In each ExpressionSet `S`, the fold changes can be accessed by `exprs(S)`. Protein expNames can be accessed by `featureNames(S)`. TMT labels and the corresponding temperatures are returned by `S$labels` and `S$temperatures`.

References

Savitski, M. M. and Reinhard, F. BM. and Franken, H. and Werner, T. and Savitski, M. F. and Eberhard, D. and Molina, D. M. and Jafari, R. and Dovega, R. B. and Klaeger, S. and others (2014) Tracking cancer drugs in living cells by thermal profiling of the proteome. Science 346(6205), p. 1255784.

See Also

`tpptrImport`

Examples

```
data(hdacTR_smallExample)
tpptrData <- tpptrImport(configTable=hdacTR_config, data=hdacTR_data)
tpptrNorm <- tpptrNormalize(data=tpptrData, normReqs=tpptrDefaultNormReqs())
tpptrNorm$qcPlotObj
```

---

tpptrQCPlotsMinSlopes_vs_MPdiffs

Plot minSlope vs. melting point differences

Description

`tpptrQCPlotsMinSlopes_vs_MPdiffs` plots the minSlope vs. melting point difference for each protein and highlights proteins with significant shifts.

Usage

```
tpptrQCPlotsMinSlopes_vs_MPdiffs(resultTable, expNames = NULL,
  ggplotTheme = tppDefaultTheme(), expRepl = NULL, expCond = NULL,
  path = NULL)
```

Arguments

resultTable	Table of the TPP-TR results.
expNames	Names of each experiment.
ggplotTheme	ggplot theme to be used in the plots.
expRepl	Replicate each experiment belongs to.
expCond	Condition (Treatment or Vehicle) each experiment belongs to.
path	Location where to store resulting plot.

Value

No value returned.

Examples

```
data(hdacTR_resultsTable_smallExample)
tpptrQCPlotsMinSlopes_vs_MPdiffs(resultTable=resultTable,
expNames=c("Vehicle_1", "Vehicle_2", "Panobinostat_1", "Panobinostat_2"),
expRepl=c(1,2,1,2),
expCond=c("Vehicle", "Vehicle", "Treatment", "Treatment"))
```

---

tpptrResultTable *Statistical analysis of melting curve parameters.*

Description

Summarizes the output of a TPP-TR experiment and performs statistical comparisons of conditions, if appropriate.

Usage

```
tpptrResultTable(data, binWidth = 300)
```

Arguments

data	list of ExpressionSets containing fold changes and metadata. It's featureData contains the fitted melting curve parameters
binWidth	bin width used for p-value computation

Details

If a TPP-TR experiment was performed with the conditions "Vehicle" and "Treatment", the melting points between these conditions will be compared statistically, producing p-values for each protein and replicate.

Value

A data frame in which the fit results are stored row-wise for each protein.

References

Cox, J., & Mann, M. (2008). MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification. Nature biotechnology, 26(12), 1367-1372.

Examples

```
data(hdacTR_fittedData_smallExample)
resultTable <- tpptrResultTable(trDataFitted)
subset(resultTable, fulfills_all_4_requirements)$Protein_ID
```

---

tppVenn *Venn diagrams of detected proteins per experiment.*

Description

tppVenn illustrates the overlaps between different TPP-TR/CCR experiments by a venn diagram.

Usage

```
tppVenn(data)
```

Arguments

data	list of ExpressionSets that contain the imported data per experiment (return value of function tpptrImport or tppccrImport.

Value

Venn diagram plot. Can be plotted by grid.draw.

Examples

```
data(hdacTR_smallExample)
trImported <- tpptrImport(configTable=hdacTR_config, data=hdacTR_data)
vennPlot <- tppVenn(data=trImported)
grid.draw(vennPlot)
```

`trDataFitted` *Example of a TPP-TR dataset with fitted melting curve parameters.*

Description

Example of a TPP-TR dataset with fitted melting curve parameters.

Details

`trDataFitted` is a list of ExpressionSets that contain the measurements of a TPP-TR experiment (accessible by the `exprs` command), as well as fitted melting curve parameters in the `featureData`. It can be used as input for the function `tpptrResultTable`.

Index