Package ‘artMS’

April 15, 2019

Type Package

Title Analytical R tools for Mass Spectrometry

Version 1.0.12

Date 2019-04-07

Author David Jimenez-Morales, Alexandre Rosa Campos, John Von Dollen.

Maintainer David Jimenez-Morales <biodavidjm@gmail.com>

Description artMS provides a set of tools for the analysis of proteomics label-free datasets. It takes as input the MaxQuant search result output (evidence.txt file) and performs quality control, relative quantification using MSstats, downstream analysis and integration. artMS also provides a set of functions to re-format and make it compatible with other analytical tools, including, SAINTq, SAINTexpress, Phosfate, and PHOTON.

License GPL (>= 3) + file LICENSE

URL https://github.com/biodavidjm/artMS

BugReports https://github.com/biodavidjm/artMS/issues

Depends R (>= 3.5.0)

Imports AnnotationDbi, biomaRt, bit64, circlize, cluster, ComplexHeatmap, corrplot, data.table, dplyr, factoextra, FactoMineR, getopt, ggdendro, ggplot2, gplots, ggrepel, gProfileR, graphics, grDevices, grid, limma, MSstats, openxlsx, org.Hs.eg.db, org.Mm.eg.db, PerformanceAnalytics, pheatmap, plotly, plyr, RColorBrewer, reshape2, seqinr, stats, stringr, tidyr, UpSetR, utils, VennDiagram, yaml

Suggests BiocStyle, knitr, rmarkdown, testthat

Encoding UTF-8

LazyLoad yes

LazyData true

biocViews Proteomics, DifferentialExpression, BiomedicalInformatics, SystemsBiology, MassSpectrometry, Annotation, QualityControl, GeneSetEnrichment, Clustering, Normalization, ImmunoOncology, MultipleComparison

NeedsCompilation no

VignetteBuilder knitr

RoxygenNote 6.1.1

Roxygen list(markdown = TRUE)

git_url https://git.bioconductor.org/packages/artMS

git_branch RELEASE_3_8

git_last_commit cb3b4fc

git_last_commit_date 2019-04-07

Date/Publication 2019-04-15

R topics documented:

artmsAnalysisQuantifications

Analysis of the Relative Quantifications

Description

Analysis of relative quantifications, including:

- Annotations
- Summary files in different format (xls, txt) and shapes (long, wide)
- Numerous summary plots
- Enrichment analysis using Gprofiler
- PCA of quantifications
- Clustering analysis
- Basic imputation of missing values

Usage

```
artmsAnalysisQuantifications(log2fc_file, modelqc_file, species,
  output_dir = ".", enrich = TRUE, l2fc_thres = 1.5,
  choosePvalue = c("adjpvalue", "pvalue"),
  isBackground = "nobackground", isPtm = "global", mnbr = 2,
  isFluomics = FALSE, pathogen = "nopathogen",
  plotPvaluesLog2fcDist = TRUE, plotAbundanceStats = TRUE,
  plotReproAbundance = TRUE, plotCorrConditions = TRUE,
  plotCorrQuant = TRUE, plotPCAabundance = TRUE,
  plotFinalDistributions = TRUE, plotPropImputation = TRUE,
  plotHeatmapsChanges = TRUE, plotTotalQuant = TRUE,
  plotClusteringAnalysis = TRUE, verbose = TRUE)
```

Arguments

`log2fc_file` (char) MSstats results file location

`modelqc_file` (char) MSstats modelqc file location

`species` (char) Select one species. Species currently supported for a full analysis (including enrichment analysis):

- HUMAN
- MOUSE

To find out species supported only for annotation check `?artmsIsSpeciesSupported()`

`output_dir` (char) Name for the folder to output the results from the function. Default is current directory (recommended to provide a new folder name).

`enrich` (logical) Performed enrichment analysis using GprofileR? Only available for species HUMAN and MOUSE. `TRUE` (default if "human" or "mouse" are the species) or `FALSE` `l2fc_thres` (int) log2fc cutoff for enrichment analysis (default, `l2fc_thres = 1.5`)

`choosePvalue` (char) specify whether `pvalue` or `adjpvalue` should use for the analysis. The default option is `adjpvalue` (multiple testing correction). But if the number of biological replicates for a given experiment is too low (for example n = 2), then `choosePvalue = pvalue` is recommended.

`isBackground` (char) background of gene names for enrichment analysis. `nobackground` (default) will use the total number of genes detected. Alternatively provided the file path name to the background gene list.

`isPtm` (char) Is a ptm-site quantification?

- `global` (default),
- `ptmsites` (for site specific analysis),
- `ptmph` (Jeff Johnson script output evidence file)

`mnbr` (int) minimal number of biological replicates for imputation and filtering. Default: `mnbr = 2` (Proteins must be found in one of the conditions in at least 2 of the biological replicates)

`isFluomics` (logical) Does this data belong to the FluOMICs project? `TRUE` or `FALSE` (default)

`pathogen` (char) Is there a pathogen in the dataset as well? if it does not, then use `pathogen = nopathogen` (default). Pathogens available: `tb` (Tuberculosis), `lpn` (Legionella)

`plotPvaluesLog2fcDist`
(logical) If `TRUE` (default) plots pvalues and log2fc distributions

`plotAbundanceStats`
(logical) If `TRUE` (default) plots stats graphs about abundance values

`plotReproAbundance`
(logical) If `TRUE` plots reproducibility based on normalized abundance values

`plotCorrConditions`
(logical) If `TRUE` plots correlation between the different conditions

`plotCorrQuant` (logical) if `TRUE` plots correlation between the available quantifications (comparisons)

`plotPCAabundance`
(logical) if `TRUE` performs PCA analysis of conditions using normalized abundance values

`plotFinalDistributions`
(logical) if `TRUE` plots distribution of both log2fc and pvalues

`plotPropImputation`
(logical) if `TRUE` plots proportion of overall imputation

`plotHeatmapsChanges`
(logical) if `TRUE` plots heatmaps of quantified changes (both all and significant only)

`plotTotalQuant` (logical) if `TRUE` plots barplot of total number of quantifications per comparison

`plotClusteringAnalysis`
(logical) if `TRUE` performs clustering analysis between quantified comparisons (more than 1 comparison required)

`verbose` (logical) `TRUE` (default) shows function messages

Value

(data.frame) summary of quantifications, including annotations, enrichments, etc

Examples

```
# Testing that the files cannot be empty
artmsAnalysisQuantifications(log2fc_file = NULL,
                              modelqc_file = NULL,
                              species = NULL,
                              output_dir = NULL)
```

`artmsAnnotateSpecie` *Adding a column with the species name*

Description

Adding the species name to every protein. This makes more sense if there are more than one species in the dataset, which must be specified in the `pathogen` option. Influenza is a special case that it does not need to be specified, as far as the proteins were originally annotated as `INFLUENZAGENE_STRAIN` (strains covered `H1N1`, `H3N2`, `H5N1`), as for example, `NS1_H1N1`

Usage

```
artmsAnnotateSpecie(df, pathogen = "nopathogen", species)
```

Arguments

`df`	(data.frame) with a `Protein` column (of uniprot ids)
`pathogen`	(char) Is there a pathogen in the dataset as well? if it does not, then use `pathogen = nopathogen` (default). Supported`tb` (Tuberculosis), `lpn` (Legionella)
`species`	(char) Host organism (supported for now: `human` or `mouse`)

Value

(data.frame) The same data.frame but with an extra column specifying the species

Examples

```
# Adding a new column with the main species of the data. Easy.
# But the main functionality is to add both the host-species and a pathogen,
# which is not illustrated in this example
data_with_specie <- artmsAnnotateSpecie(df = artms_data_ph_msstats_results,
                                        species = "human")
```

artmsAnnotationUniprot *Annotate table with Gene Symbol and Name based on Uniprot ID(s)*

Description

Annotate gene name and symbol based on uniprot ids. It will take the column from your data.frame specified by the `columnid` argument, search for the gene symbol, name, and entrez based on the species (`species` argument) and merge the information back to the input data.frame

Usage

```
artmsAnnotationUniprot(x, columnid, species, verbose = TRUE)
```

Arguments

`x`	(data.frame) to be annotated (or file path and name)
`columnid`	(char) The column with the uniprotkb ids
`species`	(char) The species name. Check `?artmsMapUniprot2Entrez` to find out more about supported species.
`verbose`	(logical) `TRUE` (default) shows function messages

Value

(data.frame) with two new columns: `Gene` and `Protein.name`

Examples

```
# This example adds annotations to the evidence file available in
# artMS, based on the column 'Proteins'.

evidence_anno <- artmsAnnotationUniprot(x = artms_data_ph_evidence,
                                        columnid = 'Proteins',
                                        species = 'human')
```

artmsAvgIntensityRT *Summarize average intensity and retention time per protein*

Description

Input an evidence file from MaxQuant and a file containing a list of proteins of interest (optional). The function will summarize from the evidence file and report back the average intensity, average retention time, and the average caliberated retention time. If a list of proteins is provided, then only those proteins will be summarized and returned.

Usage

```
artmsAvgIntensityRT(evidence_file, protein_file = NULL,
  output_file = FALSE, species, verbose = TRUE)
```

Arguments

`evidence_file`	(char) The filepath to the MaxQuant searched data (evidence) file (txt tab delimited file).
`protein_file`	(char) The filepath to a file or vector conatining a list of proteins of interest.
`output_file`	(char) The file name for the results (must have the extension `.txt`). If empty, then the results will be returned as an R object.
`species`	(char) The species name. Check `?artmsMapUniprot2Entrez` for supported species
`verbose`	(logical) `TRUE` (default) shows function messages

Value

An R object with the results and a file with the results (if the output_file argument is provided). It contains averages of Intensity, Retention Time, Caliberated Retention Time

Examples

```
ave_int <- artmsAvgIntensityRT(evidence_file = artms_data_ph_evidence,
                               species = "human")
```

`artmsChangeColumnName` *Change a specific column name in a given data.frame*

Description

Making easier to change a column name in any data.frame

Usage

```
artmsChangeColumnName(dataset, oldname, newname)
```

Arguments

`dataset`	(data.frame) with the column name you want to change
`oldname`	(char) the old column name
`newname`	(char) the new name for that column

Value

(data.frame) with the new specified column name

Examples

```
artms_data_ph_evidence <- artmsChangeColumnName(
                              dataset = artms_data_ph_evidence,
                              oldname = "Phospho..STY.",
                              newname = "PH_STY")
```

artmsConvertMetabolomics

Convert Markview Metabolomics file (alignment table) into a artMS compatible format

Description

artMS enables the relative quantification of untargeted polar metabolites using the alignment table generated by Markview. MarkerView is an ABSciex software that supports the files generated by Analyst software (.wiff) used to run our specific mass spectrometer (ABSciex Triple TOF 5600+). It also supports .t2d files generated by the Applied Biosystems 4700/4800 MALDI-TOF. MarkerView software is used to align mass spectrometry data from several samples for comparison. Using the import feature in the software, .wiff files (also .t2d MALDI-TOF files and tab-delimited .txt mass spectra data in mass-intensity format) are loaded for retention time alignment. Once the data files are selected, a series of windows will appear wherein peak finding, alignment, and filtering options can be entered and selected. These options include minimum spectral peak width, minimum retention time peak width, retention time and mass tolerance, and the ability to filter out peaks that do not appear in more than a user selected number of samples.

‘artmsConvertMetabolomics“ processes the markview file to enable QC analysis and relative quantification using the artMS functions

Usage

```
artmsConvertMetabolomics(input_file, out_file, id_file = NULL,
  verbose = TRUE)
```

Arguments

input_file	(char) Markview input file
out_file	(char) Output file name
id_file	(char) KEGG database
verbose	(logical) TRUE (default) shows function messages

Value

(text file) Outputs the converted output name

Examples

```
# Testing that the arguments cannot be null
artmsConvertMetabolomics(input_file = NULL,
                         out_file = NULL)
```

artmsDataPlots *Individual Normalized abundance dot plots for every protein*

Description

Protein abundance dot plots for each unique uniprot id. It can take a long time

Usage

```
artmsDataPlots(input_file, output_file, verbose = TRUE)
```

Arguments

`input_file`	(char) File path and name to the `-normalized.txt` output file from MSstats
`output_file`	(char) Output file (path) name (add the `.pdf` extension)
`verbose`	(logical) `TRUE` (default) shows function messages

Value

(pdf) file with each individual protein abundance plot for each conditions

Examples

```
artmsDataPlots(input_file = "results/ab-results-mss-normalized.txt",
               output_file = "results/ab-results-mss-normalized.pdf")
```

artmsEnrichLog2fc *Enrichment of changes in protein abundance or PTMs*

Description

Enrichment analysis of the selected proteins

Usage

```
artmsEnrichLog2fc(dataset, species, background, heatmaps = FALSE,
  output_name = "enrichment.txt", verbose = TRUE)
```

Arguments

`dataset`	(data.frame) with a `Gene` and `Comparison or Label` (with the name of the comparisons specified in the contrast file) columns
`species`	(char) Specie, only supported "human" or "mouse"
`background`	(vector) Background genes for the enrichment analysis.
`heatmaps`	(logical) if `TRUE` generates heatmaps (pdf), `FALSE` (default) otherwise.
`output_name`	(char) Name of the annotation files, which will be used as well for the heatmaps (if `heatmaps` is selected) Default `output_name = "enrichment.txt"`
`verbose`	(logical) `TRUE` (default) shows function messages

Value

(data.frame) Results from the enrichment analysis using Gprofiler and heatmaps (if selected)

Examples

```
# The data must be annotated (Protein and Gene columns)
data_annotated <- artmsAnnotationUniprot(
                      x = artms_data_ph_msstats_results,
                      columnid = "Protein",
                      species = "human")
# And then the enrichment
enrich_set <- artmsEnrichLog2fc(
                   dataset = data_annotated,
                   species = "human",
                   background = unique(data_annotated$Gene))
```

---

artmsEnrichProfiler *Enrichment analysis using GprofileR*

Description

This function simplifies the enrichment analysis performed by the excellent tool GprofileR.

Usage

```
artmsEnrichProfiler(x, categorySource = c("GO"), species,
  background = NA, verbose = TRUE)
```

Arguments

`x` (list, data.frame) List of protein ids. It can be anything: either a list of ids, or you could also send a data.frame and it will find the columns with the IDs. Is not cool? Multiple list can be also sent simultaneously, as for example running: `tmp <- split(enrichment$Gene, enrichment$cl_number, drop= TRUE)`

`categorySource` (vector) Resources providing the terms on which the enrichment will be performed. The supported resources by gprofiler are:

- GO (GO:BP, GO:MF, GO:CC): Gene Ontology (see more below)
- KEGG: Biological pathways
- REAC: Biological pathways (Reactome)
- TF: Regulatory motifs in DNA (TRANSFAC TFBS)
- MI: Regulatory motifs in DNA (miRBase microRNAs)
- CORUM: protein complexes database
- HP: Human Phenotype Ontology
- HPA: Protein databases (Human Protein Atlas)
- OMIM: Online Mendelian Inheritance in Man annotations:
- BIOGRID: BioGRID protein-protein interactions The type of annotations for Gene Ontology:
- Inferred from experiment (IDA, IPI, IMP, IGI, IEP)
- Direct assay (IDA) / Mutant phenotype (IMP]

- Genetic interaction (IGI) / Physical interaction (IPI)
- Traceable author (TAS) / Non-traceable author (NAS) / Inferred by curator (IC)
- Expression pattern (IEP) / Sequence or structural similarity (ISS) / Genomic context (IGC)
- Biological aspect of ancestor (IBA) / Rapid divergence (IRD)
- Reviewed computational analysis (RCA) / Electronic annotation (IEA)
- No biological data (ND) / Not annotated or not in background (NA)

`species` (char) Specie code: Organism names are constructed by concatenating the first letter of the name and the family name. Example: human - ’hsapiens’, mouse - ’mmusculus’. Check gProfileR to find out more about supported species.

`background` (vector) gene list to use as background for the enrichment analysis. Default: `NA`

`verbose` (logical) `TRUE` (default) shows function messages

Details

This function uses the following `gprofiler` arguments as default:

- ordered_query = FALSE
- significant = TRUE
- exclude_iea = TRUE
- underrep = FALSE
- evcodes = FALSE
- region_query = FALSE
- max_p_value = 0.05
- min_set_size = 0
- max_set_size = 0
- min_isect_size = 0
- correction_method = "analytical" #Options: "gSCS", "fdr", "bonferroni"
- hier_filtering = "none"
- domain_size = "known" # annotated or known
- numeric_ns = ""
- png_fn = NULL
- include_graph = TRUE

Value

The enrichment results as provided by gprofiler

Examples

```
# annotate the MSstats results to get the Gene name
data_annotated <- artmsAnnotationUniprot(
                                     x = artms_data_ph_msstats_results,
                                     columnid = ”Protein”,
                                     species = ”human”)

# Filter the list of genes with a log2fc > 2
```

```
filtered_data <-
unique(data_annotated$Gene[which(data_annotated$log2FC > 2)])

# And perform enrichment analysis
data_annotated_enrich <- artmsEnrichProfiler(
                                 x = filtered_data,
                                 categorySource = c('KEGG'),
                                 species = "hsapiens",
                                 background = unique(data_annotated$Gene))
```

artmsEvidenceToSaintExpress

MaxQuant evidence file to SAINTexpress format

Description

Converts the MaxQuant evidence file to the 3 required files by SAINTexpress. One can choose to either use the `spectral counts` (use `msspc`) or the `intensities` (use `msint`) for the analysis.

Usage

```
artmsEvidenceToSaintExpress(evidence_file, keys_file, ref_proteome_file,
  quant_variable = c("msspc", "msint"), output_file, verbose = TRUE)
```

Arguments

`evidence_file` (char) The evidence file path and name

`keys_file` (char) Keys file with a SAINT column specifying test (`T`) and control (`C`) conditions

`ref_proteome_file` (char) Reference proteome path file name in fasta format

`quant_variable` (char) choose either

- `msspc` (spectral counts, default) or
- `msint` (MS Intensity)

`output_file` (char) Output file name (must have extension .txt)

`verbose` (logical) `TRUE` (default) shows function messages

Value

The 3 required files by SAINTexpress:

- `interactions.txt`
- `preys.txt`
- `baits.txt`

Examples

```
# Testing that the files cannot be empty
artmsEvidenceToSaintExpress(evidence_file = NULL,
keys_file = NULL, ref_proteome_file = NULL)
```

artmsEvidenceToSAINTq *MaxQuant evidence file to SAINTq format*

Description

Converts the MaxQuant evidence file to the required files by SAINTq. The user can choose to use either peptides with `spectral counts` (use `msspc`) or the all the peptides (use `all`) for the analysis. The quantitative can be also chosen (either MS Intensity or Spectral Counts)

Usage

```
artmsEvidenceToSAINTq(evidence_file, keys_file, output_dir = ".",
  sc_option = c("all", "msspc"), fractions = FALSE,
  quant_variable = c("msint", "msspc"), verbose = TRUE)
```

Arguments

`evidence_file`	(char or data.frame) The evidence file path and name, or data.frame
`keys_file`	(char) Keys file with a SAINT column specifying test (`T`) and control (`C`) conditions
`output_dir`	(char) New directory to create and save files. Default is current directory (recommended to provide a new folder name).
`sc_option`	(char). Filter peptides with spectral counts only. Two options: • `msspc`: use only peptides with spectral_counts • `all` (default): all peptides detected (including the one resulting from the MaxQuant 'Match between run' algorithm)
`fractions`	(logical) `TRUE` for 2D proteomics (fractions). Default: `FALSE`
`quant_variable`	(char) Select the quantitative variable. Two options available: • `msint`: MS Intensity (default) • `msspc`: MS.MS.count (Spectral Counts)
`verbose`	(logical) `TRUE` (default) shows function messages

Details

After running the script, the new specified folder should contain the folling files:

- saintq-config-peptides
- saintq-config-proteins
- saintq_input_peptides.txt
- saintq_input_proteins.txt

Then `cd` into the new folder and run either of the following two options (assuming that `saintq` is installed in your linux/unix/mac os x system):

```
> saintq config-saintq-peptides
```

or

```
> saintq config-saintq-proteins
```

Value

The input files requires to run SAINTq

Examples

```
# Testing that the files cannot be empty
artmsEvidenceToSAINTq   (evidence_file = NULL,
                                    keys_file = NULL,
                                    output_dir = NULL)
```

---

artmsFilterEvidenceContaminants

Remove contaminants and empty proteins from the MaxQuant evidence file

---

Description

Remove contaminants and erronously identified 'reverse' sequences by MaxQuant, in addition to empty protein ids

Usage

```
artmsFilterEvidenceContaminants(x, verbose = TRUE)
```

Arguments

x	(data.frame) of the Evidence file
verbose	(logical) TRUE (default) shows function messages

Value

(data.frame) without REV__ and CON__ Protein ids

Examples

```
ef <- artmsFilterEvidenceContaminants(x = artms_data_ph_evidence)
```

---

artmsGeneratePhSiteExtended

Generate ph-site specific detailed file

Description

Generate extended detailed ph-site file, where every line is a ph site instead of a peptide. Therefore, if one peptide has multiple ph sites it will be breaking down in each of the sites. This file will help generate input files for tools as Phosfate or PHOTON

Usage

```
artmsGeneratePhSiteExtended(df, pathogen = "nopathogen", species,
  ptmType, output_name)
```

Arguments

`df`	(data.frame) of log2fc and imputed values
`pathogen`	(char) Is there a pathogen in the dataset as well? Available pathogens are `tb` (Tuberculosis), `lpn` (Legionella). If it is not, then use `nopathogen` (default).
`species`	(char) Main organism (supported for now: `human` or `mouse`)
`ptmType`	(char) It must be a ptm-site quantification dataset. Either: yes: `ptmsites` (for site specific analysis), or `ptmph` (Jeff's script output evidence file).
`output_name`	(char) A output file name (extension `.txt` required)

Value

(data.frame) extended version of the ph-site

Examples

```
artmsGeneratePhSiteExtended(df = dfobject,
                            species = "mouse",
                            ptmType = "ptmsites",
                            output_name = log2fc_file)
```

`artmsIsEvidenceNewVersion` *Check if a given evidencee file was generated by a new version of MaxQuant (v>1)*

Description

MaxQuant introduced changes in the column names and number of columns for the evidence file in version 1 (we think). This function check whether the evidence comes from the latest version of MaxQuant.

Usage

```
artmsIsEvidenceNewVersion(evidence_file)
```

Arguments

`evidence_file`	the evidence file name

Value

(logical) `TRUE` if it is a newer version of MaxQuant, `FALSE` otherwise

Examples

```
artmsIsEvidenceNewVersion(evidence_file = artms_data_ph_evidence)
```

artmsIsSpeciesSupported *Check if a species is supported and available*

Description

Given a species name, it checkes whether is supported, and if supported, check whether the annotation package is installed.

Usage

```
artmsIsSpeciesSupported(species, verbose = TRUE)
```

Arguments

`species` (char) The species name. Species currently supported as part of artMS:

- HUMAN
- MOUSE

And the following species can be used as well, but the user needs to install the corresponding org.db package:

- ANOPHELES (`install.packages(org.Ag.eg.db)`)
- BOVINE (`install.packages(org.Bt.eg.db)`)
- WORM (`install.packages(org.Ce.eg.db)`)
- CANINE (`install.packages(org.Cf.eg.db)`)
- FLY (`install.packages(org.Dm.eg.db)`)
- ZEBRAFISH (`install.packages(org.Dr.eg.db)`)
- CHICKEN (`install.packages(org.Gg.eg.db)`)
- RHESUS (`install.packages(org.Mmu.eg.db)`)
- CHIMP (`install.packages(org.Pt.eg.db)`)
- RAT (`install.packages(org.Rn.eg.db)`)
- YEAST (`install.packages(org.Sc.sgd.db)`)
- PIG (`install.packages(org.Ss.eg.db)`)
- XENOPUS (`install.packages(org.Xl.eg.db)`)

`verbose` (logical) `TRUE` (default) shows function messages

Value

(string) Name of the package for the given species

Examples

```
# Should return TRUE
artmsIsSpeciesSupported(species = "HUMAN")
artmsIsSpeciesSupported(species = "CHIMP")
```

artmsMapUniprot2Entrez

Map GENE SYMBOL, NAME, AND ENTREZID to a vector of Uniprot IDS

Description

Map GENE SYMBOL, NAME, AND ENTREZID to a vector of Uniprot IDS

Usage

```
artmsMapUniprot2Entrez(uniprotkb, species)
```

Arguments

`uniprotkb`	(vector) Vector of UniprotKB IDs
`species`	(char) The species name. Species currently supported as part of artMS: check `?artmsIsSpeciesSupported()` to find out the list of supported species‘

Value

(data.frame) with ENTREZID and GENENAMES mapped on UniprotKB ids

Examples

```
# Load an example with human proteins
exampleID <- c("Q6P996", "B1N8M6")
artmsMapUniprot2Entrez(uniprotkb = exampleID,
                       species = "HUMAN")
```

artmsMergeEvidenceAndKeys

Merge evidence.txt (or summary.txt) with keys.txt files

Description

Merge the evidence and keys files on the given columns

Usage

```
artmsMergeEvidenceAndKeys(x, keys, by = c("RawFile"),
  isSummary = FALSE, verbose = TRUE)
```

Arguments

`x` (data.frame or char) The evidence data, either as data.frame or the file name (and path). It also works for the summary.txt file

`keys` The keys data, either as a data.frame or file name (and path)

`by` (vector) specifying the columns use to merge the evidence and keys. Default: `by=c('RawFile')`

`isSummary` (logical) TRUE or FALSE (default)

`verbose` (logical) `TRUE` (default) shows function messages

Value

(data.frame) with the evidence and keys merged

Examples

```
evidenceKeys <- artmsMergeEvidenceAndKeys(x = artms_data_ph_evidence,
                                          keys = artms_data_ph_keys)
```

`artmsMsstatsSummary` *Summarize the MSStats results and data quantification*

Description

Converts the MSStats results file to wide format (unique Protein ID and columns are the comparisons), as well as adds BioReplicate information about

- the Number of Unique Peptides,
- Spectral Counts
- Intensities for each protein. In cases where there are multiple values for a Protein-BioReplicate pair due to minute changes in sequence, the maximum value is taken for the pair. Any pairs without a value are assigned a value of NA.

Usage

```
artmsMsstatsSummary(evidence_file, prot_group_file, keys_file,
  results_file, return_df = FALSE, verbose = TRUE)
```

Arguments

`evidence_file` (char or data.frame) The filepath to the MaxQuant searched data (evidence) file (txt tab delimited file). Only works for the newer versions of the evidence file.

`prot_group_file`
(char) The filepath to the MaxQuant `proteinGroups.txt` file (txt tab delimited file) or data.frame

`keys_file` (char) The filepath to the keys file used with MSStats (txt tab delimited file).

`results_file` (char) The filepath to the MSStats results file in t he default long format (txt tab delimited file or data.frame).

`return_df`	(data.frame) Whether or not to return the results to the R environment upon completion. This is useful if this is being used in an R pipeline and you want to feed the results directly into the next stage of analysis via an R environment/terminal. Regardless, the results will be written to file. Default = FALSE
`verbose`	(logical) `TRUE` (default) shows function messages

Value

(data.frame or txt file) with the summary

Examples

```
# Testing warning if files are not submitted
test <- artmsMsstatsSummary(evidence_file = NULL,
                      prot_group_file = NULL,
                      keys_file = NULL,
                      results_file = NULL)
```

`artmsPhosfateOutput` *Generate Phosfate Input file*

Description

It takes as input the `imputedL2fcExtended.txt` results generated by the `artmsAnalysisQuantifications()` function and generates the Phosfate input file (or data.frame) Please, notice that the only species suported by Phosfate is humans.

Usage

```
artmsPhosfateOutput(inputFile, output_dir = ".", verbose = TRUE)
```

Arguments

`inputFile`	(char) the `imputedL2fcExtended.txt` file name and location
`output_dir`	(char) Name of the folder to output results (Default: current directory. Recommended: `phosfate_input`)
`verbose`	(logical) `TRUE` (default) to show function messages

Value

Multiple output files (inputs of phosfate)

Examples

```
artmsPhosfateOutput(inputFile)
```

`artmsPhotonOutput` *Generate PHOTON Input file*

Description

It takes as input the `imputedL2fcExtended.txt` results generated by the `artmsAnalysisQuantifications()` function and generates the PHOTON input file. Please, notice that the only species suported by PHOTON is humans.

Usage

```
artmsPhotonOutput(inputFile, output_dir = ".", verbose = TRUE)
```

Arguments

`inputFile`	(char) the `imputedL2fcExtended.txt` file name and location
`output_dir`	(char) Name of the folder to output results (Default: current. Recommended: "photon_input_files" or similar)
`verbose`	(logical) `TRUE` (default) to show function messages

Value

Multiple output files (inputs of phosfate)

Examples

```
artmsPhotonOutput(inputFile)
```

`artmsPlotHeatmapQuant` *Outputs a heatmap of the MSStats results created using the log2fold changes*

Description

Heatmap of the Relative Quantifications (MSStats results)

Usage

```
artmsPlotHeatmapQuant(input_file,
  output_file = "quantifications_heatmap.pdf", species, labels = "*",
  cluster_cols = FALSE, display = "log2FC", lfc_lower = -2,
  lfc_upper = 2, whatPvalue = "adj.pvalue", FDR = 0.05,
  verbose = TRUE)
```

Arguments

`input_file` (char) MSstats `results.txt` file and location (or data.frame of resuts)

`output_file` (char) Output file name (pdf format) and location. Default:"quantifications_heatmap.pdf"

`species` (char). Specie name to be able to add the Gene name. To find out more about the supported species check `?artmsMapUniprot2Entrez`

`labels` (vector) of uniprot ids if only specific labes would like to be plotted. Default: all labels

`cluster_cols` (boolean) `True` or `False` to cluster columns. Default: FALSE

`display` Metric to be displayed. Options:

- `log2fc` (default)
- `adj.pvalue`
- `pvalue`

`lfc_lower` (int) Lower limit for the log2fc. Default: -2

`lfc_upper` (int) Upper limit for the log2fc. Default: +2

`whatPvalue` (char) `pvalue` or `adj.pvalue` (default)

`FDR` (int) Upper limit false discovery rate (or pvalue). Default: 0.05

`verbose` (logical) `TRUE` (default) shows function messages

Value

(pdf or ggplot2 object) heatmap of the MSStats results using the selected metric

Examples

```
artmsPlotHeatmapQuant(input_file = artms_data_ph_msstats_results,
                      species = "human",
                      output_file = NULL,
                      whatPvalue = "pvalue",
                      lfc_lower = -1,
                      lfc_upper = 1)
```

`artmsProtein2SiteConversion`

Converts the `Proteins` *column of the evidence file to site-specific* `Uniprot_PTM` *notation*

Description

It enables the site-specific quantification of PTMs by converting the `Proteins` column of the evidence file to an `Uniprot_PTM` or `RefSeq_PTM` notation. In this way, each of the modified peptides can be quantified independently across conditions. !!! WARNING: we have detected a version of MaxQuant (1.6.3.4) outputs a‘ "Modified sequence" column of the evidence file that has two important changes for the annotation of phosphorylation:

- Uses `p` instead of `(ph)`
- The modified residue (i.e. STY) is the residue on the left of the `p`, instead of the residue to the right of `(ph)` as usual. These changes significantly affect the algorithm used by this artMS function to extract the PTM site information. We are working on debugging the issue and use an alternative function independent of the "Mofified sequence" column. If your MaxQuant version is affected, then the column `mod_sites` of the output `-mapping.txt` will have many NA values for peptides with `p` modifications (and no NA values should be present)

Usage

```
artmsProtein2SiteConversion(evidence_file, ref_proteome_file, output_file,
  mod_type, verbose = TRUE)
```

Arguments

`evidence_file` (char) The evidence file name and location

`ref_proteome_file` (char) The reference proteome used as database to search the `evidence.txt` file with MaxQuant. It will be used to map the modified peptide to the protein sequence and find the site location. Therefore, it does not use the MaxQuant's `Phospho (STY)Sites.txt`

`output_file` (char) Output file name (`-sites-evidence.txt` recommended)

`mod_type` (char) The posttranslational modification. Options:

- `UB`: Protein Ubiquitination
- `PH`: Protein Phosphorylation
- `AC`: Protein Acetylation

`verbose` (logical) `TRUE` (default) shows function messages

Value

(file) Return a new evidence file with the `Proteins` column modified by adding the sequence site location(s) + postranslational modification(s) to the uniprot entry / refseq id.

Examples: `A34890_ph3`; `Q64890_ph24_ph456`; `Q64890_ub34_ub129_ub234`; `Q64890_ac35`.

Examples

```
# Testing warning if files are not submitted
artmsProtein2SiteConversion(evidence_file = NULL, ref_proteome_file = NULL,
output_file = NULL)
```

artmsQualityControlEvidenceBasic

Quality Control analysis of the MaxQuant evidence file

Description

Quality Control analysis of the MaxQuant evidence file

Usage

```
artmsQualityControlEvidenceBasic(evidence_file, keys_file,
  prot_exp = c("AB", "PH", "UB", "APMS"), fractions = 0,
  output_name = "qcPlots_evidence", plotINTDIST = TRUE,
  plotREPRO = TRUE, plotCORMAT = TRUE, plotINTMISC = TRUE,
  plotPTMSTATS = TRUE, printPDF = TRUE, verbose = TRUE)
```

Arguments

`evidence_file` (char or data.frame) The evidence file path and name, or data.frame

`keys_file` (char or data.frame) The keys file path and name or data.frame

`prot_exp` (char) Proteomics experiment. 4 options available:

- `APMS`: affinity purification mass spectrometry
- `AB`: protein abundance
- `PH`: protein phosphorylation
- `UB`: protein ubiquitination (aka ubiquitylation)

`fractions` (binary) Is a fractionated experiment?

- 1 yes
- 0 no (default)

`output_name` (char) prefix output name (no extension). Default: "qcPlots_evidence"

`plotINTDIST` if `TRUE` (default) plots both *Box-dot plot* and *Jitter plot* of biological replicates based on MS (raw) intensity values.

`plotREPRO` if `TRUE` (default) plots a correlation dotplot for all the combinations of biological replicates of conditions, based on MS Intensity values using features (peptide+charge)

`plotCORMAT` if `TRUE` (default) plots a

- *Correlation matrix* for all the biological replicates using MS Intensity values,
- *Clustering matrix* of the MS Intensities and correlation distribution
- *histogram* of the distribution of correlations

`plotINTMISC` if `TRUE` (default) plots several pages, including bar plots of *Total Sum of Intensities in BioReplicates*, *Total Sum of Intensities in Conditions*, *Total Peptide Counts in BioReplicates*, *Total Peptide Counts in conditions* separated by categories: `CON`: contaminants, `PROT` peptides, `REV` reversed sequences used by MaxQuant to estimate the FDR; *Box plots* of MS Intensity values per biological replicates and conditions; *bar plots* of total intensity (excluding contaminants) by bioreplicates and conditions; Barplots of *total feature counts* by bioreplicates and conditions.

`plotPTMSTATS` IF `TRUE` (default) plots stats related to the selected modification, including: *bar plot of peptide counts and intensities*, broken by `PTM/other` categories; bar plots of *total sum-up of MS intensity values* by other/PTM categories.

`printPDF` If `TRUE` (default) prints out the pdfs. Warning: plot objects are not returned due to the large number of them.

`verbose` (logical) `TRUE` (default) shows function messages

Value

Quality control files and plots

Examples

```
artmsQualityControlEvidenceBasic(evidence_file = artms_data_ph_evidence,
                                 keys_file = artms_data_ph_keys,
                                 prot_exp =  "PH",
                                 plotINTDIST = FALSE,
                                  plotREPRO = TRUE,
                                  plotCORMAT = FALSE,
                                  plotINTMISC = FALSE,
                                  plotPTMSTATS = FALSE,
                                  printPDF = FALSE,
                                  verbose = FALSE)

# But we recommend the following test:
# 1. Go to a working directory:
# setwd("/path/to/your/working/directory/")
# 2. Run the following command to print out all the pdf files
# artmsQualityControlEvidenceBasic(evidence_file = artms_data_ph_evidence,
#                                  keys_file = artms_data_ph_keys,
#                                  prot_exp =  "PH")
# 3. Check your working directory and you should find pdf files with
# all the QC plots
```

artmsQualityControlEvidenceExtended

Extended Quality Control of the MaxQuant evidence.txt file

Description

Performs quality control based on the information available in the MaxQuant `evidence.txt` file.

Usage

```
artmsQualityControlEvidenceExtended(evidence_file, keys_file,
  plotPSM = TRUE, plotIONS = TRUE, plotTYPE = TRUE,
  plotPEPTIDES = TRUE, plotPROTEINS = TRUE, plotPIO = TRUE,
  plotCS = TRUE, plotME = TRUE, plotMOCD = TRUE, plotPEPICV = TRUE,
  plotPEPDETECT = TRUE, plotPROTICV = TRUE, plotPROTDETECT = TRUE,
  plotIDoverlap = TRUE, plotIC = TRUE, plotSP = TRUE,
  printPDF = TRUE, verbose = TRUE)
```

Arguments

`evidence_file` (char or data.frame) The evidence file path and name, or data.frame

`keys_file` (char or data.frame) The keys file path and name or data.frame

`plotPSM` (logical) `TRUE` generates peptide-spectrum-matches (PSMs) statistics plot: Page 1 shows the number of PSMs confidently identified in each BioReplicate. If replicates are present, Page 2 shows the mean number of PSMs per condition with error bar showing the standard error of the mean. Note that potential contaminant proteins are plotted separately.

`plotIONS` (logical) `TRUE` generates peptide ion statistics plot: A peptide ion is defined in the context of m/z, in other words, an unique peptide sequence may give rise to multiple ions with different charge state and/or amino acid modification. Page 1 shows the number of ions confidently identified in each BioReplicate . If replicates are present, Page 2 shows the mean number of peptide ions per condition with error bar showing the standard error of the mean. Note that potential contaminant proteins are plotted separately.

`plotTYPE` (logical) `TRUE` generates identification type statistics plot: MaxQuant classifies each peptide identification into different categories (e.g., MSMS, MULTI-MSMS, MULTI-SECPEP). Page 1 shows the distribution of identification type in each BioReplicate

`plotPEPTIDES` (logical) `TRUE` generates peptide statistics plot: Page 1 shows the number of unique peptide sequences (disregard the charge state or amino acid modifications) confidently identified in each BioReplicate. If replicates are present, Page 2 shows the mean number of peptides per condition with error bar showing the standard error of the mean. Note that potential contaminant proteins are plotted separately. Pages 3 and 4 show peptide identification intersection between BioReplicates (the bars are ordered by degree or frequency, respectively), and Page 4 shows the intersections across conditions instead of BioReplicates.

`plotPROTEINS` (logical) `TRUE` generates protein statistics plot: Page 1 shows the number of protein groups confidently identified in each BioReplicate. If replicates are present, Page 2 shows the mean number of protein groups per condition with error bar showing the standard error of the mean. Note that potential contaminant proteins are plotted separately. Pages 3 and 4 show peptide identification intersection between BioReplicates (the bars are ordered by degree or frequency, respectively), and Page 4 shows the intersections across conditions instead of BioReplicates.

`plotPIO` (logical) `TRUE` generates oversampling statistics plot: Page 1 shows the proportion of all peptide ions (including peptides matched across runs) fragmented once, twice and thrice or more. Page 2 shows the proportion of peptide ions (with intensity detected) fragmented once, twice and thrice or more. Page 3 shows the proportion of peptide ions (with intensity detected and MS/MS identification) fragmented once, twice and thrice or more

`plotCS` (logical) `TRUE` generates charge state plot: Page 1 shows the charge state distribution of PSMs confidently identified in each BioReplicate.

`plotME` (logical) `TRUE` generates precursor mass error plot: Page 1 shows the distribution of precursor error for all PSMs confidently identified in each BioReplicate.

`plotMOCD` (logical) `TRUE` generates precursor mass-over-charge plot: Page 1 shows the distribution of precursor mass-over-charge for all PSMs confidently identified in each BioReplicate.

`plotPEPICV` (logical) `TRUE` generates peptide intensity coefficient of variance (CV) plot: The CV is calculated for each feature (peptide ion) identified in more than one replicate. Page 1 shows the distribution of CV's for each condition, while Page 2 shows the distribution of CV's within 4 bins of intensity (i.e., 4 quantiles of average intensity).

`plotPEPDETECT` (logical) `TRUE` generates peptide detection frequency plot: Page 1 summarizes the frequency that each peptide is detected across BioReplicates of each condition, showing the percentage of peptides detected once, twice, thrice, and so on (for whatever number of replicates each condition has).

`plotPROTICV` (logical) `TRUE` generates protein intensity coefficient of variance (CV) plot: The CV is calculated for each protein (after summing the peptide intensities) identified in more than one replicate. Page 1 shows the distribution of CV's for each condition, while Page 2 shows the distribution of CV's within 4 bins of intensity (i.e., 4 quantiles of average intensity).

`plotPROTDETECT` (logical) `TRUE` generates protein detection frequency plot: Page 1 summarizes the frequency that each protein group is detected across BioReplicates of each condition, showing the percentage of proteins detected once, twice, thrice, and so on (for whatever number of replicates each condition has). Page 2 shows the feature (peptide ion) intensity distribution within each BioReplicate (potential contaminant proteins are plot separately). Page 3 shows the density of feature intensity for different feature types (i.e., MULTI-MSMS, MULTI-SECPEP).

plotIDoverlap	(logical) TRUE generates pairwise identification heatmap overlap: Pages 1 and 2 show pairwise peptide and protein overlap between any 2 BioReplicates, respectively.
plotIC	(logical) TRUE generates pairwise intensity correlation: Page 1 and 3 show pairwise peptide and protein intensity correlation and scatter plot between any 2 BioReplicates, respectively. Page 2 and 4 show principal component analysis at the intensity level for both peptide and proteins, respectively.
plotSP	(logical) TRUE generates sample quality metrics: Page 1 shows missing cleavage distribution of all peptides confidently identified in each BioReplicate. Page 2 shows the fraction of peptides with at least one methionine oxidized in each BioReplicate.
printPDF	If TRUE (default) prints out the pdfs. Warning: plot objects are not returned due to the large number of them.
verbose	(logical) TRUE (default) shows function messages

Details

all the plots are generated by default

Value

A number of QC plots based on the evidence file

Examples

```
# Testing warning if files are not submitted
test <- artmsQualityControlEvidenceExtended(evidence_file = NULL,
keys_file = NULL)
```

---

artmsQualityControlMetabolomics

Quality Control analysis of the evidence-like metabolomics dataset

---

Description

Quality Control analysis of the evidence-like metabolomics dataset

Usage

```
artmsQualityControlMetabolomics(evidence_file, keys_file,
  met_exp = c("MV"), output_name = "qcPlots_metab",
  plotINTDIST = TRUE, plotREPRO = TRUE, plotCORMAT = TRUE,
  plotINTMISC = TRUE, printPDF = TRUE, verbose = TRUE)
```

Arguments

`evidence_file` (char or data.frame) The evidence file path and name, or data.frame

`keys_file` (char or data.frame) The keys file path and name or data.frame

`met_exp` (char) Proteomics experiment. Only one option available (so far):

- `MV`: Markview output

`output_name` (char) prefix output name (no extension). Default: "qcPlots_metab"

`plotINTDIST` if `TRUE` (default) plots both *Box-dot plot* and *Jitter plot* of biological replicates based on MS (raw) intensity values.

`plotREPRO` if `TRUE` (default) plots a correlation dotplot for all the combinations of biological replicates of conditions, based on MS Intensity values using features (mz_rt+charge)

`plotCORMAT` if `TRUE` (default) generates up to 3 pdf files for technical replicates, biological replicates, and conditions. Each pdf file contains:

- *Correlation matrix* for all the biological replicates using MS Intensity values,
- *Clustering matrix* of the MS Intensities and correlation distribution
- *histogram* of the distribution of correlations

`plotINTMISC` if `TRUE` (default) plots several pages, including bar plots of *Total Sum of Intensities in BioReplicates*, *Total Sum of Intensities in Conditions*, *Total Feature Counts in BioReplicates*, *Total Feature Counts in conditions* separated by categories (INT: has a intensity value NOINT: no intensity value) *Box plots* of MS Intensity values per biological replicates and conditions; *bar plots* of total intensity by bioreplicates and conditions; Barplots of *total feature counts* by bioreplicates and conditions.

`printPDF` If `TRUE` (default) prints out the pdfs. Warning: plot objects are not returned due to the large number of them.

`verbose` (logical) `TRUE` (default) shows function messages

Value

Quality control files and plots for metabolomics

Examples

```
# Testing that input arguments cannot be null
artmsQualityControlMetabolomics(evidence_file = NULL,
                keys_file = NULL,
                met_exp = "MV")
```

`artmsQualityControlSummaryExtended`
Quality Control of the MaxQuant summary.txt file

Description

Performs quality control based on the information available in the MaxQuant summary.txt file.

Usage

```
artmsQualityControlSummaryExtended(summary_file, keys_file,
  isFractions = FALSE, plotMS1SCANS = TRUE, plotMS2 = TRUE,
  plotMSMS = TRUE, plotISOTOPE = TRUE, verbose = TRUE)
```

Arguments

`summary_file`	(char or data.frame) The evidence file path and name, or data.frame
`keys_file`	(char or data.frame) The keys file path and name or data.frame
`isFractions`	(logical) `TRUE` if it is a 2D experiment (fractions). Default: `FALSE`
`plotMS1SCANS`	(logical) `TRUE` generates MS1 scan counts plot: Page 1 shows the number of MS1 scans in each BioReplicate. If replicates are present, Page 2 shows the mean number of MS1 scans per condition with error bar showing the standard error of the mean. If isFractions `TRUE`, each fraction is a stack on the individual bar graphs.
`plotMS2`	(logical) `TRUE` generates MS2 scan counts plot: Page 1 shows the number of MSs scans in each BioReplicate. If replicates are present, Page 2 shows the mean number of MS1 scans per condition with error bar showing the standard error of the mean. If isFractions `TRUE`, each fraction is a stack on the individual bar graphs.
`plotMSMS`	(logical) `TRUE` generates MS2 identification rate (Page 1 shows the fraction of MS2 scans confidently identified in each BioReplicate. If replicates are present, Page 2 shows the mean rate of MS2 scans confidently identified per condition with error bar showing the standard error of the mean. If isFractions `TRUE`, each fraction is a stack on the individual bar graphs.
`plotISOTOPE`	(logical) `TRUE` generates Isotope Pattern counts plot: Page 1 shows the number of Isotope Patterns with charge greater than 1 in each BioReplicate. If replicates are present, Page 2 shows the mean number of Isotope Patterns with charge greater than 1 per condition with error bar showing the standard error of the mean. If isFractions `TRUE`, each fraction is a stack on the individual bar graphs.
`verbose`	(logical) `TRUE` (default) shows function messages

Value

A number of plots from the summary file

Examples

```
# Testing warning if files are not submitted
test <- artmsQualityControlSummaryExtended(summary_file = NULL,
keys_file = NULL)
```

`artmsQuantification` *Relative quantification using MSstats*

Description

Relative quantification using MSstats including:

- plots
- quantifications (log2fc, pvalues, etc)
- normalized abundance values

Usage

```
artmsQuantification(yaml_config_file, verbose = TRUE)
```

Arguments

`yaml_config_file`
(char) The yaml file name and location

`verbose` (logical) `TRUE` (default) shows function messages

Value

The relative quantification of the conditions and comparisons specified in the keys/contrast file resulting from running MSstats, in addition to quality control plots (if selected)

Examples

```

artmsQuantification("artms-ab-config.yaml")
```

`artmsReplicatePlots` *Create the replicate plots based on the pairings from the replicate plot file*

Description

Outputs a replicate plots based on a user provied file containing the replicates to be compared. Values are based on the log2 value of the maximum intensities per modified sequence. The "replicate file" should describe which replicates of which conditions should be compared against each other. Each row represents a replicate plot to be created. The file should be structured using the following format and column names:

```
**condition1**|**rep1_1**|**rep1_2**|**condition2**|**rep2_1**|**rep2_2**
-----|-----|-----|-----|-----|-----
Cal33|Cal33-1|Cal33-2|HSC6|HSC6-1|HSC6-2
Cal33|Cal33-3|Cal33-4|HSC6|HSC6-3|HSC6-4
```

Usage

```
artmsReplicatePlots(input_file, keys_file, replicate_file, out_file,
  prot_exp = c("AB", "PH", "UB", "APMS"), verbose = TRUE)
```

Arguments

`input_file`	(char) MaxQuant evidence file and location
`keys_file`	(char) Keys file with the experimental details
`replicate_file`	(char) Replicate file. Check Vignette for examples
`out_file`	(char) Output .TEXT file with the intensity values of every feature
`prot_exp`	(char) Proteomics experiment. 4 options available: • `AB`: (default) protein abundance • `APMS`: affinity purification mass spectrometry • `PH`: protein phosphorylation • `UB`: protein ubiquitination (aka ubiquitylation)
`verbose`	(logical) `TRUE` (default) shows function messages

Value

The output file of the summary of features and intensity values

Examples

```
# First, let's make the "replicate file" (in a data.frame)
x_names <- c("condition1", "rep1_1", "rep1_2", "condition2", "rep2_1",
"rep2_2")
x_values <- c("Cal33", "Cal33-1", "Cal33-4", "HSC6", "HSC6-2", "HSC6-3")
replica_info <- data.frame(t(x_values))
colnames(replica_info) <- x_names

# Now let's make the plots.
artmsReplicatePlots(input_file = artms_data_ph_evidence,
                     keys_file = artms_data_ph_keys,
                     replicate_file = replica_info,
                     out_file = NULL,
                     prot_exp = "PH")

# Remember that if you want to see the txt results and pdf file, just
# change out_file = NULL" to out_file = 'output_file.pdf'
```

`artmsResultsWide` *Reshape the MSstats results file from long to wide format*

Description

Converts the normal MSStats results.txt file into "wide" format where each row represents a unique protein's results, and each column represents the comparison made by MSStats. The fold change and p-value of each comparison will be its own column.

Usage

```
artmsResultsWide(results_msstats, output_file = NULL,
  select_pvalues = c("adjpvalue", "pvalue"), species, verbose = TRUE)
```

Arguments

results_msstats
: (char) Input file name and location (MSstats `results.txt` file)

output_file
: (char) Output file name and location (e.g. `results-wide.txt`). If `NULL` (default) returns an R object (data.frame)

select_pvalues
: (char) Either
  - `pvalue` or
  - `adjpvalue` (default)

species
: (char) Specie name for annotation purposes. Check `?artmsMapUniprot2Entrez` to find out more about the supported species (e.g `species = "human"`)

verbose
: (logical) `TRUE` (default) shows function messages

Value

(output file tab delimited) reshaped file with unique protein ids and as many columns log2fc and adj.pvalues as comparisons available

Examples

```
ph_results_wide <- artmsResultsWide(
                         results_msstats = artms_data_ph_msstats_results,
                         output_file = NULL,
                         species = "human")
```

`artmsSILACtoLong` *Convert the SILAC evidence file to MSstats format*

Description

Converting the evidence file from a SILAC search to a format compatible with MSstats. It basically modifies the Raw.files adding the Heavy and Light label

Usage

```
artmsSILACtoLong(evidence_file, output, verbose = TRUE)
```

Arguments

evidence_file
: (char) Text filepath to the evidence file

output
: (char) Text filepath of the output name

verbose
: (logical) `TRUE` (default) shows function messages

Value

(data.frame) with SILAC data processed for MSstats (and output file)

Examples

```
evidence2silac <- artmsSILACtoLong(evidence_file = "silac.evicence.txt",
                                   output = "silac-evidence.txt")
```

artmsSpectralCounts *Outputs the spectral counts from the MaxQuant evidence file.*

Description

Outputs the spectral counts from the MaxQuant evidence file.

Usage

```
artmsSpectralCounts(evidence_file, keys_file, output_file = NULL,
  verbose = TRUE)
```

Arguments

evidence_file	(char) Maxquant evidence file or data object
keys_file	(char) Keys file with the experimental design or data object
output_file	(char) Output file name (add .txt extension). If NULL (default) it returns a data.frame object
verbose	(logical) TRUE (default) shows function messages

Value

A txt file with biological replicates, protein id, and spectral count columns

Examples

```
summary_spectral_counts <- artmsSpectralCounts(
                                  evidence_file = artms_data_ph_evidence,
                                  keys_file = artms_data_ph_keys)
```

artmsVolcanoPlot *Volcano plot (log2fc / pvalues)*

Description

It generates a scatter-plot used to quickly identify changes

Usage

```
artmsVolcanoPlot(mss_results, output_name = "volcano_plot.pdf",
  lfc_upper = 1, lfc_lower = -1, whatPvalue = "adj.pvalue",
  FDR = 0.05, PDF = TRUE, decimal_threshold = 16, verbose = TRUE)
```

Arguments

mss_results (data.frame or file) Selected MSstats results

output_name (char) Name for the output file (don't forget the `.pdf` extension)

lfc_upper (numeric) log2fc upper threshold (positive value)

lfc_lower (numeric) log2fc lower threshold (negative value)

whatPvalue (char) `pvalue` or `adj.pvalue` (default)

FDR (numeric) False Discovery Rate threshold

PDF (logical) Option to generate pdf format. Default: `T`

decimal_threshold (numeric) Decimal threshold for the pvalue. Default: 16 (10^-16)

verbose (logical) `TRUE` (default) shows function messages

Value

(pdf) of a volcano plot

Examples

```
artmsVolcanoPlot(mss_results = artms_data_ph_msstats_results,
                 whatPvalue = "pvalue",
                 PDF = FALSE)
```

---

artmsWriteConfigYamlFile

Write out a template file of the artMS configuration file (yaml)

Description

Creates a template file of the artMS configuration file, which is required to run `artmsQuantification`. Check `?artms_config` and the vignettes to find out more about the details of the structure of the file and how to fill it up

Usage

```
artmsWriteConfigYamlFile(config_file_name = "artms_config_file.yaml",
  verbose = TRUE)
```

Arguments

config_file_name (char) The name for the configuration file. It must have a `.yaml` extension. If `NULL`, it returns the config as a yaml object

verbose (logical) `TRUE` (default) shows function messages

Value

A file (or yaml data object) of the artMS configuration file

Examples

```
config_empty <- artmsWriteConfigYamlFile(config_file_name = NULL)
```

artms_config *artMS configuration template*

Description

The configuration file in `yaml` format contains the configuration details required to run `artmsQuantification()`, which includes quality control functions

Usage

```
artms_config
```

Format

The configuration (`yaml`) file contains the following sections:

files
- `evidence` : /path/to/the/evidence.txt
- `keys` : /path/to/the/keys.txt
- `contrasts` : /path/to/the/contrast.txt
- `output` : /path/to/the/output/results/results.txt

qc
- basic: 1 # 1 = yes; 0 = no
- extended: 1 # 1 = yes; 0 = no

data
- enabled : 1 # 1 = yes; 0 = no
- fractions:
  - enabled : 0 # 1 for protein fractionation
- silac:
  - enabled : 0 # 1 for SILAC experiments
- filters:
  - enabled : 1
- contaminants : 1
- protein_groups : remove #remove, keep
- modifications : ab # PH, UB, AB, APMS
- sample_plots : 1 # correlation plots

msstats
- enabled : 1
- msstats_input : # blank if not previous msstats input file is available
- profilePlots : none # before, after, before-after, none
- normalization_method : equalizeMedians # globalStandards (include a reference protein(s)), equalizeMedians, quantile, 0
- normalization_reference : #should be a value in the Protein column
- summaryMethod : TMP # "TMP"(default) means Tukey's median polish, which is robust estimation method. "linear" uses linear mixed model. "logOfSum" conducts log2 (sum of intensities) per run.
- censoredInt : NA # Missing values are censored or at random. 'NA' (default) assumes that all 'NA's in 'Intensity' column are censored. '0' uses zero intensities as censored intensity. In this case, NA intensities are missing at random. The output from Skyline should use '0'. Null assumes that all NA intensites are randomly missing.

- cutoffCensored : minFeature # Cutoff value for censoring. only with censoredInt='NA' or '0'. Default is 'minFeature', which uses minimum value for each feature.'minFeatureNRun' uses the smallest between minimum value of corresponding feature and minimum value of corresponding run. 'minRun' uses minumum value for each run.
- MBimpute : 1 # only for summaryMethod="TMP" and censoredInt='NA' or '0'. TRUE (default) imputes 'NA' or '0' (depending on censoredInt option) by Accelated failure model. FALSE uses the values assigned by cutoffCensored.
- feature_subset: all # all|highQuality : highQuality seems to be buggy right now

output_extras
- output_extras :
  - enabled : 1 # if 0, it wont do anything in this section
- annotate :
  - enabled: 1 # 1|0 whether to annotate the proteins in the results or not
- species : HUMAN # Supported species: HUMAN, MOUSE, ANOPHELES, ARABIDOPSIS, BOVINE, WORM, CANINE, FLY, ZEBRAFISH, ECOLI_STRAIN_K12, ECOLI_STRAIN_SAKAI, CHICKEN, RHESUS, MALARIA, CHIMP, RAT, YEAST, PIG, XENOPUS
- plots:
  - volcano: 1
  - heatmap: 1
  - LFC : -1.5 1.5 # Range of minimal log2fc
  - FDR : 0.05
  - heatmap_cluster_cols : 0
  - heatmap_display : log2FC # log2FC or pvalue

`artms_data_corum_mito_database` *CORUM Protein Complexes database use for complex enrichment analysis*

Description

The list of protein complexes has been enriched with mitochondria proteins from mouse, as described in this paper:

2018 - Ruchi Masand, Esther Paulo, Dongmei Wu , Yangmeng Wang, Danielle L. Swaney, David Jimenez-Morales, Nevan J. Krogan, and Biao Wang Proteome Imbalance of Mitochondrial Electron Transport Chain in Brown Adipocytes Leads to Metabolic Benefits. Cell Metab. 2018 Mar 06; 27(3):616-629.e4

Usage

```
artms_data_corum_mito_database
```

Format

Tab delimited file.

To find out more about the format and columns available at CORUM, please visit this link

Details

LAST CORUM DOWNLOAD DATE: 2017-08-01

`artms_data_pathogen_LPN`

LPN PATHOGEN: Legionella pneumophila subsp. pneumophila (strain Philadelphia 1 / ATCC 33152 / DSM 7513) UNIPROT IDS

Description

LPN PATHOGEN: Legionella pneumophila subsp. pneumophila (strain Philadelphia 1 / ATCC 33152 / DSM 7513) UNIPROT IDS

Usage

```
artms_data_pathogen_LPN
```

Format

A data.frame of Entry IDs

`artms_data_pathogen_TB`

TB PATHOGEN: Mycobacterium tuberculosis (strain ATCC 35801 / TMC 107 / Erdman) UNIPROTS IDS

Description

TB PATHOGEN: Mycobacterium tuberculosis (strain ATCC 35801 / TMC 107 / Erdman) UNIPROTS IDS

Usage

```
artms_data_pathogen_TB
```

Format

A data.frame of Entry IDs

`artms_data_ph_evidence` *Evidence file example*

Description

Evidence file from a PH experiment consisting of two head and neck cancer cell lines ("Conditions" `"Cal33"` and `"HSC6"`).

Unfortunately, the number of lines was reduced to 1/8 due to bioconductor limitations on data size, which means that this data is not very representative of a real evidence file. However, both the full evidence.txt and keys.txt file are available at: http://kroganlab.ucsf.edu/artms/ph/evidence.txt http://kroganlab.ucsf.edu/artms/ph/keys.txt

Usage

```
artms_data_ph_evidence
```

Format

A data frame with all the columns available in an evidence file generated with MaxQuant version 1.6.2.3

`artms_data_ph_keys` *Keys File Example*

Description

the `artMS` keys file provides the details of the experimental design for any given proteomics experiment.

This particular example belongs to a PH experiment consisting of two head and neck cancer cell lines ("Conditions" `"Cal33"` and `"HSC6"`), with 2 biological replicates each (in this reduced version)

Usage

```
artms_data_ph_keys
```

Format

Tab delimited file with the following columns:

Raw.file Raw file processed. Each one should be a unique biological (or technical) replicate

IsotopeLabelType Type of labeling. `L` is used for label free experiments

Condition Label for conditions. VERY IMPORTANT: Only alpha-numeric characters and `underscore (_)` are allowed

BioReplicate Label for the Biological replicates. VERY IMPORTANT: Use the same labeling for bioreplicate as the Condition, but adding a `dash (-)` corresponding to the number of biological replicate. For example, for `Condition "Cal"`, use `Cal-1`, `Cal-2`, `Cal-3`, etc for the bioreplicates

Run The MS run number

`artms_data_ph_msstats_results` *MSstats results file example*

Description

Relative quantification results obtained running MSstats on a PH datasets (global analysis). Changes in protein phosphorylation were quantified between two conditions

Usage

```
artms_data_ph_msstats_results
```

Format

A data frame resulting from running the lastest version of MSstats

`artms_data_randomDF` *Random data set*

Description

Dataset randomly generated for testing purposes

Usage

```
artms_data_randomDF
```

Format

A data frame with 100 rows and 10 variables:

Dataset generated using this code

```
data.frame(replicate(10,sample(0:1,100,rep=TRUE)))
```

Index