Package ‘LSPFP’

January 20, 2025

Type Package

Title Lysate and Secretome Peptide Feature Plotter

Version 1.0.3

Date 2020-05-11

Author Rafael Dellen, with contributions of Fabian Kruse

Maintainer Gereon Poschmann <gereon.poschmann@hhu.de>

Description Creates plots of peptides from shotgun proteomics analysis of secretome and lysate samples. These plots contain associated protein features and scores for potential secretion and truncation.

License GPL-3

Depends seqinr, RCurl, data.table, bit64

Imports R.utils, graphics, grDevices, stats, utils

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-13 23:40:23 UTC

RoxygenNote 6.1.1

Contents

LSPFP-package 2
printSelectedPeptides 3
wrapperLSPFP 4

Index **9**

`LSPFP-package` *Lysate and Secretome Peptide Feature Plotter*

Description

Creates plots of peptides from shotgun proteomics analysis of secretome and lysate samples. These plots contain associated protein features and scores for potential secretion and truncation.

Details

The DESCRIPTION file:

Package:	LSPFP
Type:	Package
Title:	Lysate and Secretome Peptide Feature Plotter
Version:	1.0.3
Date:	2020-05-11
Author:	Rafael Dellen, with contributions of Fabian Kruse
Maintainer:	Gereon Poschmann <gereon.poschmann@hhu.de>
Description:	Creates plots of peptides from shotgun proteomics analysis of secretome and lysate samples. These plot
License:	GPL-3
Depends:	seqinr, RCurl, data.table, bit64
Imports:	R.utils, graphics, grDevices, stats, utils
NeedsCompilation:	no
Packaged:	2016-05-19 10:46:48 UTC; dellen
Repository:	CRAN
Date/Publication:	2016-05-19 16:35:29
RoxygenNote:	6.1.1

Index of help topics:

```
LSPFP-package           Lysate and Secretome Peptide Feature Plotter
printSelectedPeptides   Only print peptides
wrapperLSPFP            wrapperLSPFP
```

The package offers two functions. Please have a look at wrapperLSPFP and printSelectedPeptides for details.

Author(s)

Rafael Dellen, with contributions of Fabian Kruse

Maintainer: Gereon Poschmann <gereon.poschmann@hhu.de>

Examples

```
#Please have a look at wrapperLSPFP for an example.
```

`printSelectedPeptides` *Only print peptides*

Description

This function allows printing of different peptide plots after running wrapperLSPFP once.

Usage

```
printSelectedPeptides(path, fname, mysorteddf = NULL)
```

Arguments

`path`	Character vector giving the path to a data folder in the analysis directory.
`fname`	Character vector containing the name of the new file.
`mysorteddf`	A data.frame that can be a sorted and/or shortened data.frame of the `feature_table` from `path`.

Details

This function offers the possibility to print more selectively peptide plots after wrapperLSPFP was applied. To get a smaller plot, it is necessary to commit an existing anlaysis directory from the AnalysisData directory. The new file is named like `fname` and stored in the specified directory `path`. If a smaller dataset should be printed, a row wise sorted or shortened `feature_table` is to be used as basis for printing but no columns should be removed. This new data.frame (`mysorteddf`) will be processed for printing.

Value

A PDF file named like fname in the analysis directory path. Returns TRUE if the printing was successful and FALSE if not.

Note

A detailed vignette will follow soon.

Author(s)

Rafael Dellen
`<Rafael.Dellen@uni-duesseldorf.de>`

Examples

```
## Not run:
#To run this example you need to run the example from wrapperLSPFP

path <- paste0(getwd(),"/AnalysisData/Test_Mouse")
test <- read.csv(paste0(path, "/feature_table.csv"))


#Rows can be deleted or sorted but columns should not be removed
#Print the first 25 rows
myprint <- test[1:25,]
#Sort by accession
myprint2 <- test[sort(test$Accession), ]

printSelectedPeptides(path, "newfeaturetable", mysorteddf = myprint)


## End(Not run)
```

---

wrapperLSPFP *wrapperLSPFP*

Description

THIS IS AN ALPHA BUILD OF 1.0.1 This function plots the positions of peptides with associated proteins from shotgun proteomics data from MaxQuant or Progenesis as input. The plots contain informations about: intensity, position, protein structure, location of protein domains, genename, protein accession, secretion score and truncation score. The plots are written to a PDF file and a data.frame containing protein feature information is saved as a rds-file and csv-file.

Usage

```
wrapperLSPFP(globpath, expname, sourcefiles, org, grlocationdf,
              version = "actual", species = c("HUMAN", "MOUSE", "RAT", "PIG"),
        proteomeid = c("UP000005640", "UP000000589", "UP000002494", "UP000008227"),
        taxid = c("9606", "10090", "10116", "9823"), domain = rep("Eukaryota", 4),
              forcedl = FALSE, pepstack = 2, pepque = 2, sortprint = "fcsmall",
              unipep = TRUE, localfasta = "none")
```

Arguments

`globpath`	Character string indicating the path of the global directory.
`expname`	Character string indicating the name of the directory where the files for this run are saved.
`sourcefiles`	Character string indicating the path to the peptides-file.
`org`	Character string specifying the organism the peptides are from.

`grlocationdf` Data.frame including the following columns: Expname, Location, Treatment, Sample, Group.

`version` Character string indicating what version of the '`BasicData`' file should be used.

`species` Character strings specifying the UniProt species names of the data sets for download from UniProt to '`BasicData`'.

`proteomeid` Character strings specifying the UniProt proteome IDs of the organisms for download. Must be in the same order as `species`.

`taxid` Character strings containing the UniProt taxonomic IDs of the organisms for download. Must be in the same order as `species`.

`domain` Character strings containing the UniProt domain description. Must be in the same order as `species`.

`forcedl` Logical, TRUE: indicates that the actual `version` of '`BasicData`' should be downloaded again, FALSE: if not.

`pepstack` Numerical indicating the minimal number of runs per group where the same peptide should have been measured.

`pepque` Numerical indicating the minimal number of peptides that should have been measured in each groupe.

`sortprint` Character string that indicates how the peptide plot will be sorted. "fcsmall": decreasing `fcsmall` values , "lflf": decreasing `fclf` values, "trunc": decreasing `tscore` values, "acc": increasing `accessions`, "ntts": decreasing values of `NTT*NTTcov`, "ctts": decreasing values of `CTT*CTTcov`.

`unipep` Logical that indicates if only unique peptides should be used. TRUE: only uniques, FALSE: all peptides in the file.

`localfasta` Character string indicating that a local FASTA file should be used. Default: "none", if the FASTA file should be downloaded from UniProt, "...": any valid `path` to a FASTA file on the system.

Details

Worklfow

1. Create a data structure.
2. Download and prepare data from UniProt.
3. Calculate features.
4. Print plots to PDF.
5. Save features as .rds and .csv.

Input

peptide-files: the program uses the file extension to decide if the input was created by MaxQuant (.txt) or by Progenesis (.csv).

`grlocationdf`**:** Should be a data.frame that contains the following columns:

`Expname`**:** This column should contain all experiment names from the peptide file that should be used for the feature plotting. Put each experiment in a single row. It should be a character string. Spaces will be filled with underscores automatically.

Location: Filled with "Secretome" or "Proteome" character strings that indicates if the experiment is from a lysate or not .
Treatment: If the experiments are based on different treatments of the cell, should be marked here. The value must be a character string that matches [A-Z][A-Z].
Sample: Here should be marked from which sample each experiment is from. The values should be simple numbers of type numeric.
Group: These values are used to set experiment groups for the secretome scores. They have to be numbers of type numeric. Set at least two groups if there are to different Locations (e.g. "Secretome" = 1 and "Proteome" = 2). Choose more groups if there are dependencies between treatment, sample and location.

grlocationdf is used to assign the experiments correctly during the different scoring and plotting functions.

Datastructure
At the beginning the following file-structure is created:
'globalpath/BasicData'
'globalpath/AnalysisData'
If it already exists, nothing new will be created and the old one is used. In BasicData all the different versions of UniProt download files are stored. In AnalysisData the output-files are stored.

UniProt
For annotation and protein sequence information the organisem specific gff-files and fasta-files are downloaded from UniProt database http://www.uniprot.org/. As default the actual data-sets are retrieved. If an already downloaded dataset should be used, set version to the foldername of the existing dataset. No archived version can be downloaded from UniProt.

Output
The PDF output file will contain a plot for every unique protein a peptide was identified by MaxQuant or Progenesis.

A data.frame containing values that are used for the score calculation will be at:

'globpath/AnalysisData/expname/feature_table.rds' and
'globpath/AnalysisData/expname/feature_table.csv'
It contains the following columns: Accession, NTT, NTTcov, CTT, CTTcov, TotalPep, ProtL, MeanSec, MeanProt, MeanSecLF, MeanProtLF, ntts, ctts, tscore, fcsmall, fclf
A .csv-file of grlocationdf will be saved in 'AnalysisData', too.
Further more 'intenscount_table.csv' and 'namesdf_table.csv' are stored in the 'AnalysisData' directory. They contain information that enables printing with print_selected_peptides.
Further more there should be two tables ('metaIntens.csv', 'metaIntens.csv') that contain different length and intensity values based on annotated extracellular and cytoplasmic regions.

Value

The return value is TRUE if no error occured and FALSE otherwise. The plots and the feature table can be found in 'globpath/AnalysisData/expname'.

Note

The download from UniProt may take a while, between 5 and 10 minutes per organism. A detailed vignette will follow soon.

Author(s)

Rafael Dellen
<Rafael.Dellen@uni-duesseldorf.de>

Examples

```
#The download of gff-files and FASTA-sequences from UniProt
# might be time consuming.
# Please consider this before running the example.

## Not run:
#please choose a path
globpath <- getwd()

expname <- "Test_Mouse"

sourcefiles <- system.file("extdata", "Mouse.txt",
                        package="LSPFP")

org <- "Mouse"


#prepare grlocationdf
expnames <- c("Lysat_PB1a","Lysat_PB2a","Lysat_PB3a","Lysat_PB4a",
  "Lysat_PB5a","Lysat_PK1a", "Lysat_PK2a","Lysat_PK3a","Lysat_PK4a",
  "Lysat_PK5a","Sekretom_PB1a","Sekretom_PB2a","Sekretom_PB3a",
  "Sekretom_PB4a","Sekretom_PB5a","Sekretom_PK1a","Sekretom_PK2a",
  "Sekretom_PK3a","Sekretom_PK4a","Sekretom_PK5a")

# Are the values from the secretome or the proteome of the cells?
explocation <- c(rep("Proteome",10),rep("Secretome",10))

# Are the cells from the same culture eg. patient?
expsample <- c(rep(1:5,4))

# Are the samples differently treated?
#(different environments, chemicals, tissue extraction technique)?
exptreatment <- c(rep("AA",5),rep("BB",5),rep("AB",5),rep("BC",5))

#Group specifies which experiments belong together
group <- c(rep(1,10),rep(2,10))
grlocationdf <- data.frame(Expname = expnames, Location = explocation,
                              Treatment = exptreatment, Sample = expsample,
                              Group = group, stringsAsFactors = FALSE)


species <- "MOUSE"
proteomeid <- "UP000000589"
taxid <- "10090"
domain <- "Eukaryota"


res <- wrapperLSPFP(globpath, expname, sourcefiles, org, grlocationdf,
                    species= species, proteomeid = proteomeid,
                    taxid = taxid, domain = domain)


## End(Not run)
```

Index

∗ **package**
    LSPFP-package, 2
∗ **print**
    printSelectedPeptides, 3
∗ **utilities**
    wrapperLSPFP, 4

LSPFP *(*LSPFP-package*)*, 2
LSPFP-package, 2

printSelectedPeptides, 3

wrapperLSPFP, 4