Package 'HiTMaP'

March 17, 2021

Type Package

Title A collection of tools for imaging MS data processing

Version 1.6.0

Date

Depends R (>= 3.5.0)

Author George GUO [aut, cre]

Maintainer George GUO <George.GUO@auckland.ac.nz>

Description

License GPL-2

LazyLoad yes

NeedsCompilation no

Encoding UTF-8

BugReports

Imports pryr,colortools,pacman,purrr,lattice,pls,
RColorBrewer,magick,ggplot2,reticulate,
dplyr,stringr,data.table,doParallel,
iterators,foreach,protViz,cleaver,
MALDIquant,Biostrings,XVector,IRanges,Cardinal,
ProtGenerics,S4Vectors,stats4,egg,
EBImage,BiocParallel,BiocGenerics,parallel,
stats,graphics,grDevices,utils,
datasets,methods,Rdisop,Rcpp,
pacman,devtools,fs,processx,S4Vectors,
spdep,zoo,KEGGREST,OrgMassSpecR,
enviPat,rgl,multtest,XML,
BiocManager,shinyFiles,shiny,
rcdk,colorspace,OneR,
gridExtra,grid,gtable,ChemmineR

VignetteBuilder knitr

Suggests knitr,
rmarkdown,
devtools,
plotly,
UniProt.ws,
bitops

RoxygenNote 7.1.1

biocViews Software, Infrastructure, Proteomics, Lipidomics, MassSpectrometry, ImagingMassSpectrometry, ImmunoOncology, Normalization, Clustering, Classification, Regression

R topics documented:

`Cleavage_rules_fun` *Cleavage_rules_fun*

Description

This is a function will return a list of pre-defined enzyme digestion specificity.

Usage

```
Cleavage_rules_fun()
```

Value

a table of Cleavage rules

Examples

```
Cleavage_rules_fun()
```

cluster_image_grid *cluster_image_grid*

Description

This function renders the clustered images for maldi imaging data set

Usage

```
cluster_image_grid(
  clusterID,
  SMPLIST,
  imdata,
  ClusterID_colname = "Protein",
  componentID_colname = "Peptide",
  Component_plot_threshold = 2,
  Component_plot_coloure = c("mono", "as.cluster"),
  smooth.image = "gaussian",
  contrast.enhance = "suppression",
  colorpallet = "Set1",
  plot_layout = c("line", "grid"),
  export_Header_table = F,
  export_footer_table = F,
  plot_style = c("fleximaging", "ClusterOnly", "rainbow"),
  protein_coverage = F,
  footer_style = "Length",
  output_png_width_limit = 1980,
  attach_summary_cluster = T,
  cluster_color_scale = c("blackwhite", "fleximaging"),
  remove_cluster_from_grid = T,
  img_brightness = 100,
  ppm = 20,
  list_of_protein_sequence,
  workdir = getwd(),
  pixel_size_um = 50,
  Score_thres = NULL
)
```

get_atoms *get_atoms*

Description

This is a function that prepare the atoms list for candidate molecules.

Usage

```
get_atoms(Symbol)
```

Arguments

Symbol	the Symbol of the molecule

Value

a atoms list

Examples

```
get_atoms(Symbol="C7H13O8P")
```

---

grid.ftable *grid.ftable*

Description

This is a function will plot table.

Usage

```
grid.ftable(d, padding = unit(4, "mm"), ...)
```

Value

a table image

Examples

```
grid.ftable()
```

HiTMaP_GUI *HiTMaP shiny-based graphical user interface (GUI)*

Description

HiTMaP shiny-based graphical user interface (GUI)

Usage

```
HiTMaP_GUI(wd = "", port = 3838)
```

Examples

```
## Start the GUI
## Not run:
HiTMaP_GUI()

## End(Not run)
```

imaging_identification
imaging_identification

Description

This is a peptide mass fingerprint search function for maldi imaging data analysis

Usage

```
imaging_identification(
  datafile,
  projectfolder = NULL,
  threshold = 0.005,
  ppm = 5,
  mode = c("Proteomics", "Metabolomics"),
  Digestion_site = "trypsin",
  missedCleavages = 0:1,
  Fastadatabase = "uniprot-bovin.fasta",
  adducts = c("M+H"),
  Modifications = list(fixed = NULL, fixmod_position = NULL, variable = NULL,
    varmod_position = NULL),
  Substitute_AA = NULL,
  Decoy_search = TRUE,
 Decoy_adducts = c("M+ACN+H", "M+IsoProp+H", "M+DMSO+H", "M+Co", "M+Ag", "M+Cu",
    "M+He", "M+Ne", "M+Ar", "M+Kr", "M+Xe", "M+Rn"),
  Decoy_mode = "isotope",
  mzrange = "auto-detect",
  Database_stats = F,
  adjust_score = FALSE,
  IMS_analysis = TRUE,
  PMFsearch = IMS_analysis,
  Load_candidatelist = IMS_analysis || plot_cluster_image_grid,
  Bypass_generate_spectrum = FALSE,
  peptide_ID_filter = 2,
  Protein_feature_summary = TRUE,
  Peptide_feature_summary = TRUE,
  plot_ion_image = FALSE,
  parallel = detectCores(),
  spectra_segments_per_file = 4,
 Segmentation = c("spatialKMeans", "spatialShrunkenCentroids", "Virtual_segmentation",
    "none", "def_file"),
  Segmentation_def = "Segmentation_def.csv",
  Segmentation_ncomp = "auto-detect",
  Segmentation_variance_coverage = 0.8,
 preprocess = list(force_preprocess = FALSE, use_preprocessRDS = TRUE, smoothSignal =
   list(method = "disable"), reduceBaseline = list(method = "locmin"), peakPick =
    list(method = "adaptive"), peakAlign = list(tolerance = ppm, units = "ppm"),
    normalize = list(method = c("rms", "tic", "reference")[1], mz = 1)),
  Smooth_range = 1,
  Virtual_segmentation_rankfile = NULL,
```

```
  Rotate_IMG = NULL,
  Region_feature_summary = FALSE,
  Spectrum_validate = TRUE,
  output_candidatelist = TRUE,
  use_previous_candidates = FALSE,
  score_method = "SQRTP",
  plot_cluster_image_grid = FALSE,
  plot_cluster_image_maxretry = 2,
  plot_cluster_image_overwrite = F,
  smooth.image = "gaussian",
  componentID_colname = "Peptide",
  ClusterID_colname = "Protein",
  Protein_desc_of_interest = ".",
  Protein_desc_of_exclusion = NULL,
  plot_unique_component = TRUE,
  FDR_cutoff = 0.05,
  use_top_rank = NULL,
  plot_matching_score = F,
  Component_plot_coloure = "mono",
  cluster_color_scale = "blackwhite",
  plot_layout = "line",
  export_Header_table = T,
  export_footer_table = T,
  attach_summary_cluster = T,
  remove_cluster_from_grid = attach_summary_cluster,
  pixel_size_um = 50,
  img_brightness = 100,
  Thread = NULL,
  cluster_rds_path = NULL,
  remove_score_outlier = F,
  Plot_score_IQR_cutoff = 0.75,
  Plot_score_abs_cutoff = -0.1,
  ...
)
```

Arguments

`datafile`	the data files' path for the analysis, leave it as blank to enable a graphical user interface to select the data
`projectfolder`	optional, if NULL script will extract the path from datafile(s), and use the first workdir as project folder
`threshold`	specify the intensities threshold (0 to 1 in percentage)to report a identified molecule
`ppm`	the mz tolerance (in ppm) for peak integration
`Digestion_site`	Set the enzyme digestion specificity by one or more regex expressions or the name of a enzyme
`missedCleavages`	miss cleavage number allowed in this PMF search
`Fastadatabase`	the fasta database used in this pmf search, the file should be placed in the same folder with data files
`adducts`	the adducts list to be used for generating the PMF search candidates
`Modifications`	set the modifications

`Substitute_AA` set the amino acid Substitutions

`Decoy_search` enable (default) or disable the decoy search

`Decoy_adducts` define the adduct list for decoy search. the decoy adducts could be "M+ACN+H","M+IsoProp+H","M or"M+Rn".

`Decoy_mode` select the decoy search mode between "isotope" (default), "element" and "adduct"

`mzrange` define the mz range for the experiment, default is 700 to 4000 m/z.

`IMS_analysis` Set `"true"` if you want to perform data pre-processing and proteomics search, set `"false"` if you want to bypass it

`peptide_ID_filter` set the minimal count of peptides needed to identify a protein

`Protein_feature_summary` `"IMS_analysis"` follow-up process that will collect all the identified peptide information and associate them with possible proteins

`Peptide_feature_summary` `"IMS_analysis"` follow-up process that will summarize all datafiles identified peptides and generats a `"peptide shortlist"` in the result summary folder

`plot_ion_image` `"Peptide_feature_summarya"` follow-up process that will plot every connponents in the `"peptide shortlist"`. please use the cluster image grid to output the images.

`parallel` the number of threads will be used in the PMF search, this option now only works for windows OS

`spectra_segments_per_file` optimal number of distinctive regions in the tissue section, a virtual segmentation will be applied to the image files with this value. To have a better PMF result you may set a value that in the sweet point of sensitivety and false discovery rate (FDR).

`Segmentation` set as "spatialKMeans" to enable a `"spatialKMeans"` Segmentation; set as "spatialShrunkenCentroids" to enable a `"spatialShrunkenCentroids"` Segmentation; If a region rank file was supplied, you can set this as "Virtual_segmentation" to perform a manual segmentation; Set it as "none" to bypass the segmentation.

`preprocess` a list of params that define the IMS data pre-processing procedure

`Smooth_range` `"Segmentation"` pixel smooth range

`Virtual_segmentation_rankfile` specify a region rank file contains region information for manualy region segmentation

`Rotate_IMG` specify a configuration file to further change the rotation of the images

`Region_feature_summary` `"IMS_analysis"` follow-up process that will summarize mz feature of all regions of all data files into the summary folder

`use_previous_candidates` set as TRUE to reload the previously generated candidate list.

`score_method` specify the peptide spectrum scoring method, "SQRTP" is recommended.

`plot_cluster_image_grid` set as `"TRUE"` to enable the protein cluster image function.

`plot_cluster_image_overwrite` Set as true to generate the cluster images regardless the existance of previously file(s)

componentID_colname
: Specify the component ID column in the result spreadsheet.

ClusterID_colname
: Specify the cluster ID column in the result spreadsheet.

Protein_desc_of_interest
: Specify a list of protein descriptions for cluster image plotting. Default setting will plot all reported proteins.

Protein_desc_of_exclusion
: Specify a list of protein descriptions to be excluded from cluster image plotting.

plot_unique_component
: Set as "TRUE" to plot only the unique components in the cluster image plotting.

FDR_cutoff
: set the protein FDR cutoff threshold, default is 5 percent

plot_matching_score
: enable the spectrum matching overlay plot

Component_plot_coloure
: set as "mono" to use a pre-defined color scale to plot component images. Set as "as.cluster" to use the previously assigned mono color in the additive cluster binning process.

cluster_color_scale
: Set as "blackwhite" to use only black and white color in the cluster image plotting. using "blackwhite" in cluster_color_scale will overwrite the components' color setting.

plot_layout
: Set as "line" to plot cluster and component images for multiple data file or as "grid" to plot cluster images for single data file. In "grid" mode, Image's will be rendered into a grid with 5 columns.

export_Header_table
: Set as "TRUE" to plot the header in the cluster image plotting. Header table includes the basic information of cluster and components.

export_footer_table
: Set as "TRUE" to plot the footer in the cluster image plotting. Footer shows the protein coverage in the Proteomics mode.

attach_summary_cluster
: Set as "TRUE" to attach an enlarged cluster image to the bottom of the cluster image.

remove_cluster_from_grid
: Set as "TRUE" to remove the cluster image from the cluster image grid. it is recommended to set this same as the attach_summary_cluster.

cluster_rds_path
: set as NULL if there is not preprocessed.rds available for a single file, script will load the raw data file which may reduce the signal intensities. For multiple samples, scripts will try to load the RDS file from each "ID" folder and merge the mz features via instrument resolution setting and output a combined RDS file to the project folder. For multiple files cluster images rendering user should set the attach_summary_cluster as False, and set remove_cluster_from_grid as true.

Value

None

Examples

```
imaging_identification(threshold=0.05, ppm=5,Digestion_site="[G]",
                       missedCleavages=0:1,Fastadatabase="murine_matrisome.fasta",
                       adducts=c("M+H","M+NH4","M+Na"),IMS_analysis=TRUE,
                       Protein_feature_summary=TRUE,plot_cluster_image=TRUE,
                       Peptide_feature_summary=TRUE,plot_ion_image=FALSE,
                      parallel=3,spectra_segments_per_file=5,Segmentation="spatialKMeans"
                       )
```

---

imaging_Spatial_Quant *imaging_Spatial_Quant*

Description

This is a spatial quantitation function for maldi imaging data set this function will read the candidate list file and generate quantification result

Usage

```
imaging_Spatial_Quant(
 datafile = tk_choose.files(filter = matrix(c("imzml file", ".imzML", "Text", ".txt",
    "All files", "*"), 3, 2, byrow = TRUE), caption =
    "Choose single or multiple file(s) for analysis"),
 threshold = 0,
 ppm = 2.5,
 Quant_list = "Metabolites of Interest.csv",
 adducts = c("M-H", "M+Cl"),
 cal.mz = F,
 mzlist_bypass = T,
 IMS_analysis = TRUE,
 Protein_feature_summary = T,
 plot_cluster_image = T,
 plot_style = "fleximaging",
 Peptide_feature_summary = T,
 plot_ion_image = FALSE,
 parallel = detectCores()/2,
 spectra_segments_per_file = 5,
 Smooth_range = 1,
 Segmentation = c("spatialKMeans", "spatialShrunkenCentroids", "Virtual_segmentation",
    "none"),
 Virtual_segmentation_rankfile = tk_choose.files(default =
  "Z:/George skyline results/maldiimaging/Maldi_imaging - Copy/radius_rank.csv",
    caption = "Choose Virtual segmentation rank info file"),
 Spectrum_feature_summary = T,
 Region_feature_summary = T,
 Region_feature_analysis = T,
 plot_each_metabolites = T,
 Cluster_level = "High",
 ClusterID_colname = "Name",
 Region_feature_analysis_bar_plot = T,
```

```
  norm_datafiles = T,
  norm_Type = "Median",
  mzrange = "auto-detect",
  BPPARAM = bpparam(),
  Rotate_IMG = NULL,
  ...
)
```

Arguments

datafile	specify the imzML data files
threshold	specify the intensities threshold (0 to 1 in percentage)to report a identified molecule
ppm	the mz tolerance (in ppm) for peak integration
Quant_list	the quantifiaction candidate list, spatial quantification will go through every datafile and collect the ion intensities for each listed component
adducts	the adducts list to be used for generating the PMF search candidates
cal.mz	If set with "true", the function will recalculate the mz value according to the column named "formular" in the Quant_list and the specified adducts.
mzlist_bypass	Set "true" if you want to bypass the mzlist generating process
Protein_feature_summary	"IMS_analysis" follow-up process that will collect all the identified peptide information and associate them with possible proteins
plot_cluster_image	"Protein_feature_summary" follow-up process that will plot the protein cluster image
plot_ion_image	"Peptide_feature_summarya" follow-up process that will plot every connponents in the "peptide shortlist"
parallel	the number of threads will be used in the PMF search, this option now only works for windows OS
spectra_segments_per_file	optimal number of distinctive regions in the imaging, a virtual segmentation will be applied to the image files with this value. To have a better PMF result you may set a value that in the sweet point of sensitivety and false discovery rate (FDR).
Smooth_range	"Segmentation" pixel smooth range
Segmentation	set as "spatialKMeans" to enable a "spatialKMeans" Segmentation; set as "spatialShrunkenCentroids" to enable a "spatialShrunkenCentroids" Segmentation; If a region rank file was supplied, you can set this as "Virtual_segmentation" to perform a manual segmentation; Set it as "none" to bypass the segmentation.
Virtual_segmentation_rankfile	specify a region rank file contains region information for manualy region segmentation
Peptide_feature_summarya	"IMS_analysis" follow-up process that will summarize all datafiles identified peptides and generats a "peptide shortlist" in the result summary folder

Value

None

Examples

```
imaging_Spatial_Quant(threshold=0.05, ppm=5,Digestion_site="[G]",
                      missedCleavages=0:1,Fastadatabase="murine_matrisome.fasta",
                      adducts=c("M+H","M+NH4","M+Na"),IMS_analysis=TRUE,
                      Protein_feature_summary=TRUE,plot_cluster_image=TRUE,
                      Peptide_feature_summary=TRUE,plot_ion_image=FALSE,
                      parallel=3,spectra_segments_per_file=5,spatialKMeans=TRUE
                      )
```

`merge_atoms` *merge_atoms*

Description

This is a function that merge the atoms to a melecule atom list .

Usage

```
merge_atoms(
  atoms,
  addelements,
  check_merge = F,
  mode = c("add", "ded"),
  multiplier = c(1, 1)
)
```

Arguments

`atoms`	the file name of candidate list
`addelements`	the folder that contains candidate list
`check_merge`	Check the final formula for any negative number of atom(s).
`mode`	add the elements to the formula by using "add", and deduct the elements by using "ded".
`multiplier`	Two number to define the atom and adding elements' proportion respectively.

Value

an atom list

Examples

```
merge_atoms(atoms=get_atoms("C7H13O8P"),addelements=list(H=1),check_merge=T,mode="ded",multiplier=c(1,2))
```

Meta_feature_list_fun *Meta_feature_list_fun*

Description

This is a function that prepare the candiate list for maldi imaging data qualitative or quantitative analysis. this function will read the candidate list file and generate mz for the adducts list defined in "adducts".

Usage

```
Meta_feature_list_fun(
  database,
  workdir = getwd(),
  adducts = c("M-H", "M+Cl"),
  cal.mz = TRUE,
  bypass = FALSE,
  BPPARAM = bpparam()
)
```

Arguments

database	the file name of candidate list
workdir	the folder that contains candidate list
adducts	the adducts list to be used for generating the PMF search candidates
cal.mz	If set with "TRUE", the function will recalculate the mz value according to the column named "formula" in the database and the specified adducts.
BPPARAM	parallel processing parameter for BiocParallel
mzlist_bypass	Set "TRUE" if you want to bypass the mzlist generating process, the function will keep the mz and adduct as it is for the furture analysis. Be sure that the table contains "mz", "adduct" and "moleculeNames" as they are essential for later steps.

Value

a table of candiate list

Examples

```
Meta_feature_list_fun(database="lipid candidates.csv",adducts=c("M-H","M+Cl"))
```

Peptide_modification *Peptide_modification*

Description

This is a function that prepare the modification list for maldi imaging data qualitative or quantitative analysis. this function will load a pre-built Unimod database (unimod.df) into global environment.

Usage

```
Peptide_modification(retrive_ID = NULL, mod_position = NULL, update_unimod = F)
```

Arguments

retrive_ID	the file name of candidate list
mod_position	the folder that contains candidate list
update_unimod	the adducts list to be used for generating the PMF search candidates

Value

a table of modification list

Examples

```
Peptide_modification()
```

Protein_feature_list_fun *Protein_feature_list_fun*

Description

This is a function that prepare the candiate list for maldi imaging data qualitative or quantitative analysis. this function will read the fasta file and generate mz for the adducts list defined in "adducts".

Usage

```
Protein_feature_list_fun(
  workdir = getwd(),
  database,
  Digestion_site = "[G]",
  missedCleavages = 0:1,
  Multiple_mode = c("sequential", "parallel"),
  adducts = c("M+H", "M+NH4", "M+Na"),
  BPPARAM = bpparam(),
  Decoy_search = T,
  Decoy_mode = c("adducts", "elements", "isotope", "sequence"),
```

```
  Decoy_adducts = c("M+He", "M+Ne", "M+Ar", "M+Kr", "M+Xe", "M+Rn"),
 Substitute_AA = list(AA = c(NULL), AA_new_formula = c(NULL), Formula_with_water =
   c(NULL)),
 mzrange = c(500, 4000),
 output_candidatelist = T,
 Modifications = list(fixed = NULL, fixmod_position = NULL, variable = NULL,
   varmod_position = NULL),
 use_previous_candidates = F,
 Protein_desc_of_exclusion = NULL,
 Database_stats = F
)
```

Arguments

workdir	the folder that contains fasta file
database	the file name of fasta file
Digestion_site	Digestion pattern, this is the
missedCleavages	Define a number range for misscleaved peptides to be considered
adducts	the adducts list to be used for generating the PMF search candidates
BPPARAM	parallel processing parameter for BiocParallel

Value

a table of peptide candiate list

Examples

```
Protein_feature_list_fun(database="lipid candidates.csv",adducts=c("M+H","M+NH4","M+Na"))
```

Index