Package 'proteoQC'

October 2, 2019

Type Package

Title An R package for proteomics data quality control

Version 1.21.0

Author Bo Wen <wenbo@genomics.cn>, Laurent Gatto <lg390@cam.ac.uk>

Maintainer Bo Wen <wenbo@genomics.cn>

Description This package creates an HTML format QC report for MS/MS-based proteomics data. The report is intended to allow the user to quickly assess the quality of proteomics data.

Depends R (>= 3.0.0), XML, VennDiagram, MSnbase

Imports rTANDEM, plyr, seqinr, Nozzle.R1, ggplot2, reshape2, parallel, rpx, tidyr, dplyr, plotly, rmarkdown,

License LGPL-2

Suggests RforProteomics, knitr, BiocStyle, R.utils, RUnit,BiocGenerics

VignetteBuilder knitr

URL https://github.com/wenbostar/proteoQC

biocViews ImmunoOncology, Proteomics, MassSpectrometry, QualityControl, Visualization, ReportWriting

RoxygenNote 6.0.1

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

git_branch master

git_last_commit f5542d9

git_last_commit_date 2019-05-02

Date/Publication 2019-10-01

R topics documented:

`addSummaryChart` *Add PRIDE summary charts*

Description

Add PRIDE summary charts in the technical replicate level

Usage

```
addSummaryChart(res)
```

Arguments

`res`	An object returned by msQCpipe function

`calcMSQCMetrics` *Calculate the MS1 and MS2 level QC metrics*

Description

Calculate the MS1 level QC metrics

Usage

```
calcMSQCMetrics(spectraList = NULL, cpu = 2, outdir = "./")
```

Arguments

spectraList	An experiment design input file
cpu	The number of cpu used
outdir	Output directory

Value

A data frame

Author(s)

Bo Wen <wenbo@genomics.cn>

---

chargeStat *Charge distribution*

Description

Read the charge information from mgf file

Usage

```
chargeStat(mgf = NULL)
```

Arguments

mgf	A file of mgf.

Value

A data.frame object

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
mgf.zip <- system.file("extdata/mgf.zip", package = "proteoQC")
unzip(mgf.zip)
charge <- chargeStat("test.mgf")
```

cntStat *contaminants stat*

Description

Common Contaminants in Proteomics Mass Spectrometry Experiments

Usage

```
cntStat(res)
```

Arguments

res	An object of msQCres

Value

A data.frame will be shown in HTML report

combineRun *Combine multiple results*

Description

Combine multiple results

Usage

```
combineRun(pepFiles, fasta, outPathFile, outdir, prefix)
```

Arguments

pepFiles	peptideSummary files
fasta	database file
outPathFile	out file
outdir	output directory
prefix	output prefix

Value

A data.frame

Author(s)

Bo Wen <wenbo@genomics.cn>

createTargetDecoyDB *Create target-decoy database*

Description

Create target-decoy database

Usage

```
createTargetDecoyDB(fa, outdb)
```

Arguments

fa	target database
outdb	output target-decoy database

Value

target-decoy database file name

Author(s)

Bo Wen <wenbo@genomics.cn>

getEnzyme *Get the enzymes list*

Description

Get the enzymes list

Usage

```
getEnzyme()
```

Value

A data frame which contains all of the enzymes

Author(s)

Bo Wen <wenbo@genomics.cn>

getMods *Get the modification list*

Description

Get the modification list

Usage

```
getMods()
```

Value

A data frame which contains all of the modifications

Author(s)

Bo Wen <wenbo@genomics.cn>

labelRatio *Calculate the labeling efficiency of isobaric labeling data*

Description

Calculate the labeling efficiency of isobaric labeling data

Usage

```
labelRatio(ms = NULL, reporter = 1, plot = TRUE)
```

Arguments

ms	MS/MS file.
reporter	Isobaric tag class, 1=iTRAQ-4plex, 2=iTRAQ-8plex, 3=TMT-6plex. 4=TMT-10plex.
plot	Logical value

Value

A vector object

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
mgf.zip <- system.file("extdata/mgf.zip", package = "proteoQC")
unzip(mgf.zip)
a <- labelRatio("test.mgf",reporter=2)
```

loadmsQCres *Load the result of* msQCpipe

Description

Load the result of msQCpipe

Usage

```
loadmsQCres(outdir)
```

Arguments

outdir	The output directory of msQCpipe

Author(s)

Laurent Gatto <lg390@cam.ac.uk>, Bo Wen <wenbo@genomics.cn>

Examples

```
zpqc <- system.file("extdata/qc.zip", package = "proteoQC")
unzip(zpqc)
qcres <- loadmsQCres("./qc")
```

msQCpipe *The main function of msQC pipeline*

Description

This function is designed to automate generating of target-decoy database, database searcing, post-processing and report generation.

Usage

```
msQCpipe(spectralist = NULL, fasta = "", outdir = "./", mode = "",
  miss = 2, enzyme = 1, varmod = NULL, fixmod = NULL, tol = 10,
  tolu = "ppm", itol = 0.6, itolu = "Daltons", threshold = 0.01,
  cpu = 0, xmx = 2, refine = TRUE, ntt = 1, ...)
```

Arguments

spectralist	A file contains the experiment design or a single mgf file
fasta	database file, must contain decoy sequences
outdir	output directory
mode	identification or quantification
miss	max miss clevage
enzyme	enzyme

varmod	Variable modifications are those which may or may not be present.
fixmod	Fixed modifications are applied universally, to every instance of the specified residue(s) or terminus.
tol	The error window on experimental peptide mass values
tolu	Units can be selected from: ppm, Daltons(also da or Da).
itol	Error window for MS/MS fragment ion mass values.
itolu	Units can be selected from: Daltons(also da or Da)
threshold	FDR value for PSM
cpu	Max number of cpu used
xmx	JAVA -Xmx
refine	Refine search for X!Tandem, default is TRUE.
ntt	Semi-tryptic, 1; fully-tryptic, 2.
...	Additional parameters passed to `read.table` used to read the experimental design.

Value

A list which contains all of the information for data quality report generating

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
## Not run:
library("rpx")
px <- PXDataset("PXD000864")
mgfs <- grep("mgf", pxfiles(px), value = TRUE)
mgfs <- grep("-0[5-6]-[1|2]", mgfs, value=TRUE)
mgffiles <- pxget(px, mgfs)
library("R.utils")
mgffiles <- sapply(mgffiles, gunzip)
## Generate the lightweight qc report,
## trim the mgf files to 1/10 of their size.
trimMgf <- function(f, m = 1/10, overwrite = FALSE) {
  message("Reading ", f)
  x <- readLines(f)
  beg <- grep("BEGIN IONS", x)
  end <- grep("END IONS", x)
  n <- length(beg)
  message("Sub-setting to ", m)
  i <- sort(sample(n, floor(n * m)))
  k <- unlist(mapply(seq, from = beg[i], to = end[i]))
  if (overwrite) {
    unlink(f)
    message("Writing ", f)
    writeLines(x[k], con = f)
    return(f)
  } else {
    g <- sub(".mgf", "_small.mgf", f)
    message("Writing ", g)
```

```
    writeLines(x[k], con = g)
    return(g)
  }
}
set.seed(1)
mgffiles <- sapply(mgffiles, trimMgf, overwrite = TRUE)
fas <- pxget(px, "TTE2010.zip")
fas <- unzip(fas)
design <- system.file("extdata/PXD000864-design.txt", package = "proteoQC")
read.table(design, header = TRUE)
qcres <- msQCpipe(spectralist = design,
                  fasta = fas,
                  outdir = "./qc",
                  miss  = 0,
                  enzyme = 1, varmod = 2, fixmod = 1,
                  tol = 10, itol = 0.6, cpu = 2,
                  mode = "identification")

## End(Not run)
```

`plotBioRepVenn` *Venn plot in biological replicate level*

Description

Venn plot in biological replicate level

Usage

```
plotBioRepVenn(res)
```

Arguments

`res`	An object of msQCres

Value

The name of the figure

`plotFractionIDResult` *Barplot in different level for each fraction*

Description

Barplot in different level for each fraction

Usage

```
plotFractionIDResult(res, level = NA)
```

Arguments

res	An object of msQCres
level	1: total spectrum, 2: identified spectrum, 3: identified peptide, 4: identified protein.

Value

The name of the figure

---

plotMS1Error *plot MS1 mass error*

Description

plot MS1 mass error

Usage

```
plotMS1Error(res, plot.class = "ppm")
```

Arguments

res	An object of msQCres
plot.class	ppm or da

Value

The name of the figure

---

plotMS2Error *plot MS2 mass error*

Description

plot MS2 mass error

Usage

```
plotMS2Error(res)
```

Arguments

res	An object of msQCres

Value

The name of the figure

plotMS2Error_obsolete *plot MS2 mass error*

Description

plot MS2 mass error

Usage

```
plotMS2Error_obsolete(res)
```

Arguments

res	An object of msQCres

Value

The name of the figure

plotSampleIDResultErrorBar *Error barplot in different level for each fraction*

Description

Error Barplot in different level for each fraction

Usage

```
plotSampleIDResultErrorBar(res, level = NA)
```

Arguments

res	An object of parser result
level	1: total spectrum, 2: identified spectrum, 3: identified peptide, 4: identified protein.

Value

The name of the figure

plotSampleVenn *Venn plot in sample level*

Description

Venn plot in sample level

Usage

```
plotSampleVenn(res)
```

Arguments

res	An object of msQCres

Value

The name of the figure

plotTechRepVenn *Venn plot in technical replicate level*

Description

Venn plot in technical replicate level

Usage

```
plotTechRepVenn(res)
```

Arguments

res	An object of msQCres

Value

The name of the figure

print.msQCres *Print the information of msQCres object*

Description

Print the information of msQCres object

Usage

```
## S3 method for class 'msQCres'
print(x, ...)
```

Arguments

x	A msQCres object
...	Additional parameters

Author(s)

Laurent Gatto <lg390@cam.ac.uk>, Bo Wen <wenbo@genomics.cn>

Examples

```
zpqc <- system.file("extdata/qc.zip", package = "proteoQC")
unzip(zpqc)
qcres <- loadmsQCres("./qc")
print.msQCres(qcres)
```

proteinGroup *Protein inference*

Description

Protein inference

Usage

```
proteinGroup(file = NULL, db = "", pepColName = "peptide",
  proColName = "protein", spectrumColName = "index", proSep = ";",
  outfile = NULL, xmx = 1)
```

Arguments

file	A file containing the information of peptides to proteins.
db	A protein database of fasta format.
pepColName	The column name of peptide sequence.
proColName	The column name of protein ID.
spectrumColName	The column name of spectrum index.

proSep	The separator of protein ID, default is "".
outfile	The output file name of protein group result.
xmx	JAVA -Xm

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
pep.zip <- system.file("extdata/pep.zip", package = "proteoQC")
unzip(pep.zip)
proteinGroup(file = "pep.txt", outfile = "pg.txt")
```

---

reportHTML *HTML format report generator*

Description

HTML format report generator

Usage

```
reportHTML(res)
```

Arguments

res	An object returned by msQCpipe function

Value

null

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
zpqc <- system.file("extdata/qc.zip", package = "proteoQC")
unzip(zpqc)
qcres <- loadmsQCres("./qc")
html <- reportHTML(qcres)
```

runTandem *Run X!Tandem*

Description

Run X!Tandem

run X!Tandem

Usage

```
runTandem(spectra = "", fasta = "", outdir = "./", outprefix = "",
  cpu = 1, enzyme = 1, xmx = 2, varmod = NULL, fixmod = NULL,
  refine = TRUE, ntt = 1, tol = 10, tolu = "ppm", itol = 0.6,
  itolu = "Daltons", miss = 1)
```

Arguments

spectra	MS/MS peak list file
fasta	database file
outdir	output directory
outprefix	output file prefix
cpu	The number of CPU used for X!Tandem
enzyme	The ID of enzyme used for database searching. See showEnzyme.
xmx	Set for parameter of "Java -Xmx".
varmod	Variable modifications used for database searching. See showMods.
fixmod	Fixed modifications used for database searching. See showMods.
refine	Refine search, default is TRUE
ntt	Default is 1
tol	The error window on experimental peptide mass values
tolu	Units can be selected from: ppm, Daltons.
itol	Error window for MS/MS fragment ion mass values.
itolu	Units can be selected from: Daltons
miss	Max miss clevage

Value

a file path

Author(s)

Bo Wen <wenbo@genomics.cn>

showEnzyme *Shown all enzymes*

Description

Shown all enzymes

Usage

```
showEnzyme()
```

Value

A data frame which contains all of the enzymes

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
showEnzyme()
```

showMods *Shown all modifications*

Description

Shown all modifications

Usage

```
showMods()
```

Value

A data frame which contains all of the modifications

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
showMods()
```

Index