Package 'saas'

January 24, 2017

Title An implementation of the Search All, Asses Subset strategy for FDR estimation shotgun proteomics.

Version 0.0.0.9000

Description An implementation of the Search All, Asses Subset strategy for FDR estimation in shotgun proteomics.

Depends R (>= 3.3.0)

License Apache License, Version 2.0

URL https://github.com/compomics/search-all-assess-subset

BugReports https://github.com/compomics/search-all-assess-subset/issues

Encoding UTF-8

LazyData true

Imports tidyverse (>= 1.0.0),
cowplot (>= 0.7.0),
markdown (>= 0.7.0)

Suggests mzR (>= 2.8.0)

RoxygenNote 5.0.1

R topics documented:

`calculate_fdr` *Calculates qvalues on the subset PSMs.*

Description

Calculates qvalues on the subset PSMs.

Usage

```
calculate_fdr(df, score_higher = TRUE)
```

Arguments

`df`	dataframe with at least 3 columns: **score** score assigned to the peptide to spectrum match (PSM). **subset** TRUE if PSM belongs to the subset in interest, FALSE or otherwise. **decoy** TRUE if decoy PSM, FALSE otherwise. Additional columns are allowed but ignored. Target and decoy PSMs are assumbed to be from a competitive target decoy database search.
`score_higher`	TRUE if a higher score means a better PSM.

Value

A data frame containing all columns in "df". Following columns are added:

pi_0_cons conservative estimation of π_0.

FDR estimated subset PSM qvalues calculated according the competitive target decoy approach.

FDR_BH estimated subset PSM qvalues calculated according the Benjamini Hochbergh procedure. When provided, non-subset decoy PSMs are used to stabilize estimates in small subsets

FDR_stable estimated subset PSM qvalues calculated with "pi_0_cons". When provided, non-subset decoy PSMs are used to stabilize estimates in small subsets

Examples

```
## Simulate a dataset with 140 correct target subset PSMs, 60 incorrect target subset PSMS,
## 60 decoy subset PSMs and 2000 additional decoy PSMs.
set.seed(10)
d = sample_dataset(H1_n = 140,H0_n = 60, decoy_n = 60 ,decoy_large_n = 2000,
                   H0_mean = 2.7, H1_mean = 3, decoy_mean = 2.7, decoy_large_mean = 2.7)

## calculate the qvalues in the subset target PSMS according the classical target-decoy approach
## and our more stable estimation method.
calculate_fdr(d)
```

dpi0 *Density function for the π_0 distribution.*

Description

Density function for the π_0 distribution.

Usage

```
dpi0(pi0, n_targets, n_decoys)
```

Arguments

pi0	vector of π_0 quantiles.
n_targets	vector of observed target PSMs.
n_decoys	vector of observed decoy PSMs.

Value

vector of densities. The length is the maximum length of the numerical arguments. Returns 'NaN' for 'pi0 < 0' and 'pi > 1'.

Examples

```
## density at pi0 = .5 when observing 10 targets and 3 decoys
dpi0(.5, 10, 3)

## visualize the pi0 distribution when observing 10 targets and 3 decoys
grid = seq(0,1,.01)
dens = dpi(grid,10 , 3)
plot(dens, xlab = 'pi0', ylab = 'density')
##Alternatively, you can also use the function pi0plot()
pi0plot(10,3)
```

id_is_present *Checks if protein id appears in the headers of a fasta file.*

Description

Checks if protein id appears in the headers of a fasta file.

Usage

```
id_is_present(protein_id, fastapath)
```

Arguments

protein_id	Vector of protein ids.
fastapath	Location of the fasta file.

Value

Logical vector, TRUE if protein id is present in provided fasta file, FALSE otherwise.

Examples

```
## Location of the zipped data files
zip_file_path = system.file("extdata", "extdata.zip", package = "saas")

## Unzip and get the (temporary) location of the mzid file with the MS-GF+ search results from a
## competitive target decoy search of the complete pyrococcus proteome against a pyrococcus dataset.
mzid_file_path = unzip(zip_file_path, 'pyrococcus.mzid',exdir = tempdir())
## Parse the mzid file
dat = parse_msgf_mzid(mzid_file_path)

## Unzip and get the (temporary) location of the file with fasta headers.
## Each fasta header contains a protein_id from the protein subset of interest.
## These protein_ids match the protein_ids in the mzid result file.
fasta_file_path = unzip(zip_file_path, 'transferase_activity_[GO:0016740].fasta', exdir = tempdir())
protein_ids = unique(dat$protein_id)
head(protein_ids)
is_subset = id_is_present(protein_ids, fasta_file_path)
## Check how many of the identified proteins are subset and non subset protiens.
table(is_subset)
```

---

parse_msgf_mzid *Parses a mzID file generated by MS-GF+.*

Description

See https://omics.pnl.gov/software/ms-gf for more info on how to perform a database search on MSMS dataset with MS-GF+ and how to generate a mzID file. Note that most functions in these package require data from a competitive target decoy search.

Usage

```
parse_msgf_mzid(mzid_path)
```

Arguments

mzid_path	Location of the mzID file.

Value

A data frame containing the following 7 columns:

spec_id Id of the spectrum from the searched dataset file.

sequence Amino acid sequence matching the spectra.

protein_id Id of the sequence from the database file.

score score assigned to the peptide to spectrum match (PSM).

database Name of the database file used to search the spectra.

decoy TRUE if decoy PSM, FALSE otherwise.

database_size Number of sequences in the database file.

Examples

```
## Location of the zipped data files
zip_file_path = system.file("extdata", "extdata.zip", package = "saas")

## Unzip and get the (temporary) location of the mzid file with the MS-GF+ search results from a
## competitive target decoy search of the complete pyrococcus proteome against a pyrococcus dataset.
mzid_file_path = unzip(zip_file_path, 'pyrococcus.mzid',exdir = tempdir())
## Parse the mzid file
parse_msgf_mzid(mzid_file_path)
```

`pi0plot` *Creates density plot of the pi0 distribution.*

Description

There is also a vertical line plotted that represent a conservative π_0 estimate. This estimate is used in our more stable FDR estimation in the subset PSMs of interest.

Usage

```
pi0plot(n_targets, n_decoys)
```

Arguments

`n_targets`	vector of observed target PSMs.
`n_decoys`	vector of observed decoy PSMs.

Value

ggplot object.

Examples

```
## Visualize the pi0 distribution when observing 10 targets and 3 decoys
## pi0plot(10,3)
```

`plot_diag` *Plot diagnostic plots to evaluate assumptions from the search all, search subset strategy.*

Description

Four diagnostic plots are created:

a pi0plot according the number of subset target and decoy PSMs.

b PPplot of the decoy distribution against the subset target distribution.

c PPplot of the decoy distribution against the subset decoy distribution.

d PPplot of the subset decoy distribution against the subset target distribution.

Usage

```
plot_diag(df, score_higher = TRUE)
```

Value

ggplot object.

Examples

```
## Simulate a dataset with 140 correct target subset PSMs, 60 incorrect target subset PSMS,
## 60 decoy subset PSMs and 2000 additional decoy PSMs.
set.seed(10)
d = sample_dataset(H1_n = 140,H0_n = 60, decoy_n = 60 ,decoy_large_n = 2000,
                   H0_mean = 2.7, H1_mean = 3.2, decoy_mean = 2.7, decoy_large_mean = 2.7)
##pi_0 can be estimated with the target-decoy aproach

plot_diag(d)
```

`PPplot` *Creates PP plot of two empirical distributions.*

Description

Creates PP plot of two empirical distributions.

Usage

```
PPplot(score, label, pi0 = 0, score_higher = TRUE,
  title = "PP plot of target PSMs", xlab = "Decoy percentile",
  ylab = "Target\npercentile")
```

Arguments

score	vector of quantiles of distribution 1 and 2
label	vector of logical values. TRUE if score belongs to distribution 1
pi0	mixture coefficient of distribution 1 in distribution 2
score_higher	TRUE if a higher score means a better PSM.
title	main title.
xlab	label on x-axis.
ylab	label on y-axis.

Value

ggplot object

Examples

```
## Simulate a dataset with 140 correct target subset PSMs, 60 incorrect target subset PSMS,
## 60 decoy subset PSMs and 2000 additional decoy PSMs.
set.seed(10)
d = sample_dataset(H1_n = 140,H0_n = 60, decoy_n = 60 ,decoy_large_n = 2000,
                   H0_mean = 2.7, H1_mean = 3, decoy_mean = 2.7, decoy_large_mean = 2.7)

##pi_0 can be estimated with the target-decoy approach
pi0 = sum(d$decoy & d$subset)/sum(!d$decoy & d$subset)
PPplot(d$score, d$decoy, pi0)
```

---

preprocess *Preprocess data from a MS-GF mzID file.*

Description

The parsed data frame from saas::parse_msgf_mzid function contains sometimes multiple entries for a spectrum. (eg. if sequence can be assigned to multiple protein ids). This function takes care of this by default.

Usage

```
preprocess(dat, remove_target_decoy_PSM = TRUE,
  remove_multiple_proteins_PSM = FALSE, is_subset = NULL)
```

Arguments

dat	Data frame generated by the saas::parse_msgf_mzid function.
remove_target_decoy_PSM	TRUE to remove PSMs that match both a target and decoy sequence.
remove_multiple_proteins_PSM	TRUE to remove PSMs that can be assigned to multiple protein ids.
is_subset	Location of fasta file with protein_id of the subset of interest in the fasta headers.

Value

Data frame with the same columns as "dat". The column protein_id contains all protein_ids that can be assigned to this PSM. Multiple protein_ids are separated by ";". When "is_subset" is specified, two columns are added:

subset TRUE if sequence can be assigned to a subset protein id

non_subset TRUE if sequence can be assigned to a non subset protein id

Every spectrum haves only 1 row in the data frame.

Examples

```
## Location of the zipped data files
zip_file_path = system.file("extdata", "extdata.zip", package = "saas")

## Unzip and get the (temporary) location of the mzid file with the MS-GF+ search results from a
## competitive target decoy search of the complete pyrococcus proteome against a pyrococcus dataset.
mzid_file_path = unzip(zip_file_path, 'pyrococcus.mzid',exdir = tempdir())
## Parse the mzid file
dat = parse_msgf_mzid(mzid_file_path)

## Unzip and get the (temporary) location of the file with fasta headers.
## Each fasta header contains a protein_id from the protein subset of interest.
## These protein_ids match the protein_ids in the mzid result file.
fasta_file_path = unzip(zip_file_path, 'transferase_activity_[GO:0016740].fasta', exdir = tempdir())

## Preprocess the data before FDR estimation.
data_prep = preprocess(dat, is_subset = fasta_file_path)

## Estimate the FDR in the subset.
data_result = calculate_fdr(data_prep, score_higher = FALSE)
## Check how many PSMs are retained at the 1% FDR threshold.
table(data_result$FDR_stable < .01)
```

---

rpi0 *Random generation for the π_0 distribution.*

Description

Random generation for the π_0 distribution.

Usage

```
rpi0(n, n_targets, n_decoys)
```

Arguments

n	number of observations.
n_targets	number of observed target PSMs.
n_decoys	number of observed decoy PSMs.

Value

vector of random deviates. The length equals 'n'.

Examples

```
## visualize the pi0 distribution when observing 10 targets and 3 decoys
x = rpi0(100000, 10 ,3 )
hist(x, breaks = 50 , xlab = 'pi0', ylab = 'counts')
```

`saas_gui` *Launches the GUI version of saas.*

Description

To easily launch the GUI outside an R session (eg. on a server), you can run R -e "library(saas);saas_gui()" from the terminal (on linux/mac).

Usage

```
saas_gui(options = list(port = 3320, host = "0.0.0.0"))
```

Arguments

`options`	See help of shiny::shinyApp for more details on available options

---

`simulate_subset` *Random generation of a dataset after TDA.*

Description

Random generation of number of decoy, correct target and incorrect target PSMs after a competitive target-decoy search.

Usage

```
simulate_subset(n, pi0, sims = 1)
```

Arguments

`n`	number of total PSMs.
`pi0`	theoretical π_0.
`sims`	number of observations.

Value

A data frame with "sims" rows and 6 rows:

n number of PSMs.

pi0 theoretical π_0.

decoy_n number of decoy PSMs.

target_n number of target PSMs.

H0_n number of incorrect target PSMs.

H1_n number of correct target PSMs.

Examples

```
## Simulate the number of decoys, correct targets and incorrect targets in 10 datasets that consist of
## 100 PSMs and that have on average 20% incorrect target PSMs.
simulate_subset(100, .2, 10)
```

Index