Package 'TargetDecoy'

November 12, 2026

Title Diagnostic Plots to Evaluate the Target Decoy Approach

Version 1.18.0

Date 2022-10-21

Description A first step in the data analysis of Mass Spectrometry (MS) based proteomics data is to identify peptides and proteins. With this respect the huge number of experimental mass spectra typically have to be assigned to theoretical peptides derived from a sequence database. Search engines are used for this purpose. These tools compare each of the observed spectra to all candidate theoretical spectra derived from the sequence data base and calculate a score for each comparison. The observed spectrum is then assigned to the theoretical peptide with the best score, which is also referred to as the peptide to spectrum match (PSM). It is of course crucial for the downstream analysis to evaluate the quality of these matches. Therefore False Discovery Rate (FDR) control is used to return a reliable list PSMs. The FDR, however, requires a good characterisation of the score distribution of PSMs that are matched to the wrong peptide (bad target hits). In proteomics, the target decoy approach (TDA) is typically used for this purpose. The TDA method matches the spectra to a database of real (targets) and nonsense peptides (decoys). A popular approach to generate these decoys is to reverse the target database. Hence, all the PSMs that match to a decoy are known to be bad hits and the distribution of their scores are used to estimate the distribution of the bad scoring target PSMs. A crucial assumption of the TDA is that the decoy PSM hits have similar properties as bad target hits so that the decoy PSM scores are a good simulation of the target PSM scores. Users, however, typically do not evaluate these assumptions. To this end we developed TargetDecoy to generate diagnostic plots to evaluate the quality of the target decoy method.

License Artistic-2.0

URL https://www.bioconductor.org/packages/TargetDecoy, https://statomics.github.io/TargetDecoy/, https://github.com/statOmics/TargetDecoy/

BugReports https://github.com/statOmics/TargetDecoy/issues

biocViews MassSpectrometry, Proteomics, QualityControl, Software, Visualization

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Depends R (>= 4.1)

Imports ggplot2, ggpubr, methods, miniUI, mzID, mzR, shiny, stats

Suggests BiocStyle, knitr, msdata, sessioninfo, rmarkdown, gridExtra, testthat (>= 3.0.0), covr

VignetteBuilder knitr

Config/testthat/edition 3

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

git_branch RELEASE_3_23

git_last_commit c7a3a7a

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Elke Debrie [aut, cre], Lieven Clement [aut] (ORCID: <https://orcid.org/0000-0002-9050-4370>), Milan Malfait [aut] (ORCID: <https://orcid.org/0000-0001-9144-3701>)

Maintainer Elke Debrie <elkedebrie@gmail.com>

Contents

TargetDecoy-package 2
createPPlotObjects 3
createPPlotScores 4
decoyScoreTable 5
evalTargetDecoys 6
ModSwiss 9
ModSwissXT 9

Index **10**

`TargetDecoy-package` *TargetDecoy: Diagnostic Plots to Evaluate the Target Decoy Approach*

Description

A first step in the data analysis of Mass Spectrometry (MS) based proteomics data is to identify peptides and proteins. With this respect the huge number of experimental mass spectra typically have to be assigned to theoretical peptides derived from a sequence database. Search engines are used for this purpose. These tools compare each of the observed spectra to all candidate theoretical spectra derived from the sequence data base and calculate a score for each comparison. The observed spectrum is then assigned to the theoretical peptide with the best score, which is also referred to as the peptide to spectrum match (PSM). It is of course crucial for the downstream analysis to evaluate the quality of these matches. Therefore False Discovery Rate (FDR) control is used to return a reliable list PSMs. The FDR, however, requires a good characterisation of the score distribution of PSMs that are matched to the wrong peptide (bad target hits). In proteomics, the target decoy approach (TDA) is typically used for this purpose. The TDA method matches the spectra to a database of real (targets) and nonsense peptides (decoys). A popular approach to generate these decoys is to reverse the target database. Hence, all the PSMs that match to a decoy are known to be bad hits and the distribution of their scores are used to estimate the distribution of the bad scoring target PSMs. A crucial assumption of the TDA is that the decoy PSM hits have similar properties as bad target hits so that the decoy PSM scores are a good simulation of the target PSM scores. Users, however, typically do not evaluate these assumptions. To this end we developed TargetDecoy to generate diagnostic plots to evaluate the quality of the target decoy method.

Author(s)

Maintainer: Elke Debrie <elkedebrie@gmail.com>

Authors:

- Lieven Clement <lieven.clement@ugent.be> (ORCID)
- Milan Malfait <milan.malfait@ugent.be> (ORCID)

See Also

Useful links:

- https://www.bioconductor.org/packages/TargetDecoy
- https://statomics.github.io/TargetDecoy/
- https://github.com/statOmics/TargetDecoy/
- Report bugs at https://github.com/statOmics/TargetDecoy/issues

---

`createPPlotObjects` *Create all the PP plots in one figure for scores from multiple objects*

Description

Create all the PP plots in one figure for scores from multiple objects

Usage

```
createPPlotObjects(object_list, decoy, score, log10 = TRUE)
```

Arguments

object_list	List of mzID or mzRident objects. If named, the names will be used in the legend of the plot. If not, names will be extracted from the data files in case of *mzID* or *mzRident* objects.
decoy	character, name of the variable that indicates if the peptide matches to a target or to a decoy. When this value is missing, a Shiny gadget is launched to select it interactively.
score	numeric, indicating the score of the peptide match, obtained by the search engine. When this value is missing, a Shiny gadget is launched to select it interactively.
log10	logical to indicate if the score should be -log10-transformed. Default: TRUE. When this value is missing, a Shiny gadget is launched to select it interactively.

Value

One PP plot with all original pi0, and a standardized / rescaled PP plot with all pi0 set to 0.

Author(s)

Elke Debrie, Lieven Clement

Examples

```
library(mzID)

## Use two example files from the mzID package
exampleFiles <- system.file(
    "extdata", c("55merge_omssa.mzid", "55merge_tandem.mzid"),
    package = "mzID"
)
mzObjects <- lapply(exampleFiles, mzID)

createPPlotObjects(mzObjects,
    decoy = "isdecoy",
    score = c("omssa:evalue", "x\\!tandem:expect"),
    log10 = TRUE
)
```

---

createPPlotScores *Evaluate assumptions of the Target Decoys Approach for multiple search engines*

Description

Create diagnostic PP plots in one figure to evaluate the TDA assumptions for multiple search engines. The function provides the possibility to evaluate each of the sub-engines and the overall itself.

Usage

```
createPPlotScores(object, scores, decoy, log10 = TRUE)
```

Arguments

object	A data.frame, mzID or mzRident object.
scores	A character vector of multiple score names from the input object. Typically from different search engines.
decoy	character, name of the variable that indicates if the peptide matches to a target or to a decoy. When this value is missing, a Shiny gadget is launched to select it interactively.
log10	logical to indicate if the score should be -log10-transformed. Default: TRUE. When this value is missing, a Shiny gadget is launched to select it interactively.

Value

One PP plot with all original pi0, and a standardized / rescaled PP plot with all pi0 set to 0.

Author(s)

Elke Debrie, Lieven Clement

Examples

```
library(mzID)

## Use one of the example files in the mzID package
exampleFile <- system.file("extdata", "55merge_tandem.mzid", package = "mzID")
mzIDexample <- mzID(exampleFile)

plots <- createPPlotScores(mzIDexample,
    scores = c("x\\!tandem:hyperscore", "x\\!tandem:expect"),
    decoy = "isdecoy", log10 = TRUE
)
```

---

decoyScoreTable *Prepare score table for decoys*

Description

Takes an input object and returns a score table for the decoys.

Usage

```
decoyScoreTable(object, decoy, score, log10 = TRUE)
```

Arguments

object	A data.frame, mzID or mzRident object.
decoy	character, name of the variable that indicates if the peptide matches to a target or to a decoy. When this value is missing, a Shiny gadget is launched to select it interactively.
score	numeric, indicating the score of the peptide match, obtained by the search engine. When this value is missing, a Shiny gadget is launched to select it interactively.
log10	logical to indicate if the score should be -log10-transformed. Default: TRUE. When this value is missing, a Shiny gadget is launched to select it interactively.

Value

A data.frame with a logical "decoy" column and numeric "scores".

Author(s)

Elke Debrie, Lieven Clement

Examples

```
library(mzID)

## Use one of the example files in the mzID package
exampleFile <- system.file("extdata", "55merge_tandem.mzid", package = "mzID")
mzIDexample <- mzID(exampleFile)

decoyScoreTable(mzIDexample, decoy = "isdecoy", score = "x\\!tandem:expect")
```

---

evalTargetDecoys *Evaluate assumptions of the Target Decoys approach*

Description

Create diagnostic plots to evaluate the TDA assumptions. A histogram and PP plot allow to check both necessary assumptions.

Usage

```
evalTargetDecoys(
  object,
  decoy = NULL,
  score = NULL,
  log10 = TRUE,
  nBins = 50,
  maxPoints = 1000
)
```

```
evalTargetDecoysPPPlot(
  object,
  decoy = NULL,
  score = NULL,
  log10 = TRUE,
  zoom = FALSE,
  maxPoints = 1000
)

evalTargetDecoysHist(
  object,
  decoy = NULL,
  score = NULL,
  log10 = TRUE,
  nBins = 50,
  zoom = FALSE
)
```

Arguments

object	A data.frame, mzID or mzRident object.
decoy	character, name of the variable that indicates if the peptide matches to a target or to a decoy. When this value is missing, a Shiny gadget is launched to select it interactively.
score	numeric, indicating the score of the peptide match, obtained by the search engine. When this value is missing, a Shiny gadget is launched to select it interactively.
log10	logical to indicate if the score should be -log10-transformed. Default: TRUE. When this value is missing, a Shiny gadget is launched to select it interactively.
nBins	numeric indicating the number of bins in the histogram. When this value is missing, a Shiny gadget is launched to select it interactively.
maxPoints	numeric indicating the maximum number of dots shown in the PP plot. If maxPoints is larger than the number of target scores, a dot in the PP plot corresponds with each target score in the object. The default is 1000 points.
zoom	Logical value indicating whether a zoomed version of the plot should be returned. Default: FALSE.

Value

evalTargetDecoys returns an overview of the following four plots:

1. A PP plot showing the empirical cumulative distribution of the target distribution in function of that of the decoy distribution
2. A histogram showing the score distributions of the decoys and non-decoys
3. A zoomed PP plot
4. A zoomed histogram

`evalTargetDecoysPPPlot` generates the PP plot only (1.) or the zoomed version (3.) if `zoom = TRUE`.

`evalTargetDecoysHist` generates the histogram only (2.) or the zoomed version (4.) if `zoom = TRUE`.

The Shiny gadget

Sometimes the variable names are not known up front. If this is the case, the `evalTargetDecoys*()` functions can be called with only an input object. This launches a Shiny gadget that allows selecting the variables interactively. A histogram and PP-plot of the selected variables are created on the fly for previewing, together with a snapshot of the selected data.

Author(s)

Elke Debrie, Lieven Clement, Milan Malfait

Examples

```
library(mzID)

## Use one of the example files in the mzID package
exampleFile <- system.file("extdata", "55merge_tandem.mzid", package = "mzID")
mzIDexample <- mzID(exampleFile)

# Plot the overview of the four plots
evalTargetDecoys(mzIDexample,
    decoy = "isdecoy", score = "x\\!tandem:expect", log10 = TRUE
)

# Plot the PP plot only
evalTargetDecoysPPPlot(mzIDexample,
    decoy = "isdecoy", score = "x\\!tandem:expect", log10 = TRUE
)

# Plot the zoomed PP plot only
evalTargetDecoysPPPlot(mzIDexample,
    decoy = "isdecoy", score = "x\\!tandem:expect", log10 = TRUE,
    zoom = TRUE
)

# Plot the histogram only
evalTargetDecoysHist(mzIDexample,
    decoy = "isdecoy", score = "x\\!tandem:expect", log10 = TRUE
)

# Plot the zoomed histogram only
evalTargetDecoysHist(mzIDexample,
    decoy = "isdecoy", score = "x\\!tandem:expect", log10 = TRUE,
    zoom = TRUE
)

## mzRident objects can also be used
```

```
library(mzR)

if (requireNamespace("msdata", quietly = TRUE)) {
    ## Using example file from msdata
    file <- system.file("mzid", "Tandem.mzid.gz", package = "msdata")
    mzid <- openIDfile(file)
}
evalTargetDecoys(mzid,
    decoy = "isDecoy", score = "X.Tandem.expect", log10 = TRUE
)
```

`ModSwiss` *Swiss-Prot MS-GF+ data*

Description

Data from a Pyrococcus furiosis sample run on a LTQ-Orbitrap Velos mass spectrometer. The data can be found in the PRIDE repository with identifier PXD001077. The Pyrococcus furiosis reference proteome fasta files were downloaded from UniProtKB/Swiss-Prot on April 22, 2016. The Pyrococcus data was searched against all Pyrococcus proteins with MS-GF+ search engines using the reference proteome from UniProtKB/Swiss-Prot.

Usage

```
data(ModSwiss)
```

Format

An mzID object.

`ModSwissXT` *Swiss-Prot X!Tandem data*

Description

Data from a Pyrococcus furiosis sample run on a LTQ-Orbitrap Velos mass spectrometer. The data can be found in the PRIDE repository with identifier PXD001077. The Pyrococcus furiosis reference proteome fasta files were downloaded from UniProtKB/Swiss-Prot on April 22, 2016. The Pyrococcus data was searched against all Pyrococcus proteins with a combined search (omssa, X!Tandem and MS-GF+) using the reference proteome from UniProtKB/Swiss-Prot.

Usage

```
data(ModSwissXT)
```

Format

An mzID object.

Index

∗ **datasets**
ModSwiss, 9
ModSwissXT, 9
∗ **internal**
TargetDecoy-package, 2

createPPlotObjects, 3
createPPlotScores, 4

decoyScoreTable, 5

evalTargetDecoys, 6
evalTargetDecoysHist *(*evalTargetDecoys*)*, 6
evalTargetDecoysPPPlot *(*evalTargetDecoys*)*, 6

ModSwiss, 9
ModSwissXT, 9
mzID, *4–7*, *9*
mzRident, *4–7*

TargetDecoy *(*TargetDecoy-package*)*, 2
TargetDecoy-package, 2