Package 'pRolocGUI'

February 22, 2018

Title Interactive visualisation of spatial proteomics data

Version 1.12.0

Author Lisa M Breckels, Thomas Naake and Laurent Gatto

Maintainer Laurent Gatto <lg390@cam.ac.uk>,
Lisa M Breckels <lms79@cam.ac.uk>

Description The package pRolocGUI comprises functions to interactively visualise organelle (spatial) proteomics data on the basis of pRoloc, pRolocdata and shiny.

Depends methods, R (>= 3.1.0), pRoloc (>= 1.11.1), Biobase, MSnbase (>= 2.1.11)

Imports shiny (>= 0.9.1), scales, dplyr, DT (>= 0.1.40), graphics, utils, ggplot2

Suggests pRolocdata, knitr, BiocStyle (>= 2.5.19), rmarkdown

License GPL-2

URL http://ComputationalProteomicsUnit.github.io/pRolocGUI/

BugReports https://github.com/ComputationalProteomicsUnit/pRolocGUI/issues

VignetteBuilder knitr

Video https://www.youtube.com/playlist?list=PLvIXxpatSLA2loV5Srs2VBpJIYUlVJ4ow

biocViews Proteomics, Visualization, GUI

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

pRolocVis *Interactive visualisation of spatial proteomics data*

Description

These functions allow one to explore spatial proteomics data interactively.

Usage

```
pRolocVis(object, app = "main", fcol, ...)

pRolocVis_aggregate(object, fcol, groupBy, fig.height = "600px",
  fig.width = "100%", legend.width = "200%", legend.cex = 1,
  nchar = 40, all = TRUE, mirrorX = FALSE, mirrorY = FALSE, ...)

pRolocVis_classify(object, fcol, scol, mcol = "markers", legend.cex = 1,
  ...)

pRolocVis_compare(object, fcol1, fcol2, foi, fig.height = "600px",
  fig.width = "100%", legend.width = "200%", legend.cex = 1,
  remap = TRUE, nchar = 40, all = TRUE, mirrorX = FALSE,
  mirrorY = FALSE, ...)

pRolocVis_pca(object, fcol = "markers", foi, fig.height = "600px",
  fig.width = "100%", legend.width = "200%", legend.cex = 1,
  nchar = 40, all = TRUE, ...)
```

Arguments

object	An instance of class MSnSet, or an MSnSetList of length 2 if using "compare" application.
app	The type of application requested: "main" (default), "classify", "compare" or "aggregate". See description below.
fcol	The feature meta-data label (fData column name) to be used for colouring. Default is "markers". This will correspond to the prediction column if using "classify", or the markers (labelled data) to be plotted otherwise. If set to NULL, no annotation is expected.
...	Additional parameters passed to plot2D for the "main", "classify", "compare" apps. For the "aggregate" app this is for additional parameters to be passed to combineFeatures.
groupBy	The feature meta-data label (fData column name) to be used for summarising the features to be combined.
fig.height	Height of the figure. Default is "600px".
fig.width	Width of the figure. Default is "100px".
legend.width	Width of the legend. Default is "200%".
legend.cex	Point character expansion for the the legend. Default is 1.
nchar	Maximum number of characters of the markers class names, before their names are truncated. Default is 10.

all	If TRUE all clusters are displayed on startup, if the total number of clusters is less than including 15. If FALSE or otherwise, only the first cluster in the list is displayed.
mirrorX	Should the first PC of the second MSnSet in object be mirrored (default is FALSE). Only relevant when remap is FALSE.
mirrorY	Should the second PC of the second MSnSet in object be mirrored (default is FALSE). Only relevant when remap is FALSE.
scol	The feature meta data column containing the classification scores.
mcol	The feature meta data column containing the labelled training data, for use with "classify".
fcol1	In yhe compare app this is the feature meta-data label (fData column name) for the first dataset in the MSnSetList. Default is markers.
fcol2	In the compare app this is the feature meta-data label (fData column name) for the second dataset in the MSnSetList. Default is markers.
foi	A FeaturesOfInterest or FoICollection object.
remap	A logical indicating whether the second dataset in the MSnSetList should be remapped to the first dataset. The default is TRUE.

Details

The function pRolocVis is a wrapper for pRolocVis_main, pRolocVis_classify,pRolocVis_compare. and pRolocVis_aggregate. These Shiny apps allow to explore and analyse interactively spatial proteomics data.

The main Shiny app allows exploration of quantitative data (1) visually through Principle Component Analysis (PCA), (2) protein profiles, and (3) a searchable feature data table, allowing visualisation of particular proteins of interest.

The classify Shiny app is used to visualise classification results and set user-specified thresholds for sub-cellular location predictions.

The compare Shiny app is meant for comparing protein localisation between two conditions, or two different experiments, replicates etc. Please note that passing the argument method to ... will not work as it is already specified internally.

The aggregation Shiny app displays a scatter plot of the maximum or mean distances within each feature (e.g. protein group) according to its components (e.g. peptides) defined by the groupBy argument. A PCA plot of the components is also displayed. It can be used for visualising peptides, PSMs or any other features defined in the feature data of the MSnSet and their distributions.

Value

For the classify app a numeric vector of thresholds, one per class, to use with getPredictions

For the main, compare and aggregate apps a character vector of featureNames names of the object loaded that have been selected in the app upon application closure.

Author(s)

Laurent Gatto, Lisa Breckels and Thomas Naake

See Also

The package vignette: vignette("pRolocGUI").

Examples

```
library("pRoloc")
library("pRolocdata")
data(hyperLOPIT2015)
## Load the "main" PCA app
if (interactive()) {
  pRolocVis(hyperLOPIT2015)
}

## Load classification results from hyperLOPIT stored in fData
if (interactive()) {
  myThreshold <- pRolocVis(hyperLOPIT2015, app = "classify",
                           fcol = "svm.classification",
                           scol = "svm.score")
  newPredictions <- getPredictions(hyperLOPIT2015, fcol = "svm.classification",
                                   scol = "svm.score", t = myThreshold)
}

## Visualise the location and distribution of peptides per protein group
data("hyperLOPIT2015ms2psm")
if (interactive()) {
  ## Combine PSM data to peptides
  hl <- combineFeatures(hyperLOPIT2015ms2psm,
                        groupBy = fData(hyperLOPIT2015ms2psm)$Sequence,
                        fun = median)
  ## Visualise peptides according to protein group
  pRolocVis(hl, app = "aggregate", fcol = "markers",
            groupBy = "Protein.Group.Accessions")
}
```

Index