Package ‘esetVis’

March 28, 2017

Type Package
title Visualizations of expressionSet Bioconductor object
version 1.0.1
date 2016-08-11
author Laure Cougnaud <laure.cougnaud@openanalytics.eu>
maintainer Laure Cougnaud <laure.cougnaud@openanalytics.eu>
description Utility functions for visualization of expressionSet (or
SummarizedExperiment) Bioconductor object, including spectral
map, tsne and linear discriminant analysis. Static plot via the
ggplot2 package or interactive via the ggvis or rbokeh packages
are available.
import mpm, hexbin, Rtsne, MLP, grid, Biobase, MASS, stats, utils,
grDevices
suggests ggplot2, ggvis, rbokeh, ggrepel, knitr, rmarkdown, ALL,
hgu95av2.db, AnnotationDbi, pander, SummarizedExperiment
biocViews Visualization, DataRepresentation, DimensionReduction,
PrincipalComponent, Pathways
vignettebuilder knitr
license GPL-3
roxygennote 5.0.1
needscompilation no

R topics documented:

esetLda .......................................................... 2
esetPlotWrapper ............................................. 6
esetSpectralMap ............................................. 10
esetTsne ..................................................... 14
getCoordGeneSets .......................................... 18
getGeneSetsForPlot ......................................... 18
getMethodInputObjectEsetVis ........................... 20
plotTopElements ............................................ 20

Index 22
esetLda  

plot a biplot of a linear discriminant analysis of an eSet object

Description

esetLda reduces the dimension of the data contained in the eSet via a linear discriminant analysis on the specified grouping variable with the lda function and plot the subsequent biplot, possibly with sample annotation and gene annotation contained in the eSet.

Usage

esetLda(eset, ldaVar, psids = 1:nrow(eset), dim = c(1, 2),
        colorVar = NULL, color = if (is.null(colorVar)) "black" else NULL,
        shapeVar = NULL, shape = if (is.null(shapeVar)) 15 else NULL,
        sizeVar = NULL, size = if (is.null(sizeVar)) 2.5 else NULL,
        sizeRange = NULL, alphaVar = NULL, alpha = if (is.null(alphaVar)) 1 else NULL,
        alphaRange = NULL, title = "", symmetryAxes = c("combine",
        "separate", "none"), packageTextLabel = c("ggrepel", "ggplot2"),
        cloudGenes = TRUE, cloudGenesColor = "black",
        cloudGenesNBins = sqrt(length(psids)), cloudGenesIncludeLegend = FALSE,
        cloudGenesTitleLegend = "nGenes", topGenes = 10, topGenesCex = 2.5,
        topGenesVar = NULL, topGenesJust = c(0.5, 0.5), topGenesColor = "black",
        topSamples = 10, topSamplesCex = 2.5, topSamplesVar = NULL,
        topSamplesJust = c(0.5, 0.5), topSamplesColor = "black",
        geneSets = list(), geneSetsVar = NULL, geneSetsMaxNChar = NULL,
        topGeneSets = 10, topGeneSetsCex = 2.5, topGeneSetsJust = c(0.5, 0.5),
        topGeneSetsColor = "black", includeLegend = TRUE,
        includeLineOrigin = TRUE, typePlot = c("static", "interactive"),
        packageInteractivity = c("rbokeh", "ggvis"), figInteractiveSize = c(600,
        400), ggvisAdjustLegend = TRUE, interactiveToolTip = TRUE,
        interactiveToolTipExtraVars = NULL, returnAnalysis = FALSE)

Arguments

eset  expressionSet (or SummarizedExperiment) object with data
ldaVar  name of variable (in varLabels of the eset) used for grouping for lda, NULL by default
psids  featureNames of genes to include in the plot, all by default
dim  dimensions of the analysis to represent, first two dimensions by default
colorVar  name of variable (in varLabels of the eset) used for coloring, NULL by default
color  specified color(s) for the points, replicated if needed, used only if colorVar is NULL, a factor or character by default: 'black' if colorVar is not specified and default ggplot palette otherwise
shapeVar  name of variable (in varLabels of the eset) used for the shape, NULL by default
shape  specified shape(s) (pch) for the points, replicated if needed, used only if shapeVar is NULL, a factor or character by default: '15' (filled square) if shapeVar is not specified and default ggplot shape(s) otherwise
sizeVar  name of variable (in varLabels of the eset) used for the size, NULL by default
size
specified size(s) (cex) for the points, replicated if needed, used only if sizeVar
is NULL, a factor or character by default: '2.5' if sizeVar is not specified and
default ggplot size(s) otherwise

sizeRange
size (cex) range used in the plot, possible only if the sizeVar is 'numeric' or
'integer'

alphaVar
name of variable (in varLabels of the eset) used for the transparency, NULL by
default. This parameter is currently only available for static plot.

alpha
specified transparency(s) for the points, replicated if needed, used only if shapeVar
is NULL, a factor or character by default: '1' if alphaVar is not specified and
default ggplot alpha otherwise. This parameter is currently only available for
static plot.

alphaRange
transparency (alpha) range used in the plot, possible only if the alphaVar is
'numeric' or 'integer'. This parameter is currently only available for static plot.

title
plot title, " by default

symmetryAxes
set symmetry for axes, either:
• 'combine' (by default): both axes are symmetric and with the same limits
• 'separate': each axis is symmetric and has its own limits
• 'none': axes by default (plot limits)

packageTextLabel
package used to label the outlying genes/samples/gene sets, either ggrepel (by
default, only used if package ggrepel is available), or ggplot2

cloudGenes
logical, if TRUE (by default), include the cloud of genes in the spectral map

cloudGenesColor
if cloudGenes is TRUE, color for the cloud of genes, black by default

cloudGenesNBins
number of bins to use for the clouds of genes, by default the square root of the
number of genes

cloudGenesIncludeLegend
logical, if TRUE (FALSE by default) include the legend for the cloud of genes
(in the top position if multiple legends)

cloudGenesTitleLegend
string with title for the legend for the cloud of genes 'nGenes' by default

topGenes
numeric indicating which percentile (if <1) or number (if >=1) of genes most
distant to the origin of the plot to annotate, by default: 10 genes are selected. If
no genes should be annotated, set this parameter to 0. Currently only available
for static plot.

topGenesCex
cex for gene annotation (used when topGenes > 0)

topGenesVar
variable of the featureData used to label the genes, by default: NULL, the featureNames
are used for labelling (used when topGenes > 0)

topGenesJust
text justification for the genes (used when topGenes > 0 and if packageTextLabel
is ggplot2), by default: c(0.5, 0.5) so centered

topGenesColor
text color for the genes (used when topGenes > 0), black by default

topSamples
numeric indicating which percentile (if <1) or number (if >=1) of samples most
distant to the origin of the plot to annotate, by default: 10 samples are selected.
If no samples should be annotated, set this parameter to 0. Currently available
for static plot.

topSamplesCex
cex for sample annotation (used when topSamples > 0)
topSamplesVar: variable of the phenoData used to label the samples, by default: NULL, the sampleNames are used for labelling (used when topSamples > 0).

topSamplesJust: text justification for the samples (used when topSamples > 0 and if packageTextLabel is ggplot2), by default: c(0.5, 0.5) so centered.

topSamplesColor: text color for the samples (used when topSamples > 0), black by default.

geneSets: list of gene sets/pathways, each containing identifiers of genes contained in the set. E.g. pathways from Gene Ontology databases output from the `getGeneSetsForPlot` function or any custom list of pathways. The genes identifiers should correspond to the variable geneSetsVar contained in the phenoData, if not specified the featureNames are used. If several gene sets have the same name, they will be combine to extract the top gene sets.

geneSetsVar: variable of the featureData used to match the genes contained in geneSets, most probably ENTREZID, if not specified the featureNames of the eSet are used. Only used when topGeneSets > 0 and the parameter geneSets is specified.

geneSetsMaxNChar: maximum number of characters for pathway names, by default keep entire names. Only used when topGeneSets > 0 and the parameter geneSets is specified. If returnAnalysis is set to TRUE and geneSetsMaxNChar specified, the top pathways will be returned in the output object, named with the identifiers used in the plot (so with maximum geneSetsMaxNChar number of characters).

topGeneSets: numeric indicating which percentile (if <=1) or number (if >1) of gene sets most distant to the origin of the plot to annotate, by default: 10 gene sets are selected. If no gene sets should be annotated, set this parameter to 0. Currently available for static plot. Only used when topGeneSets > 0 and the parameter geneSets is specified.

topGeneSetsCex: cex for gene sets annotation. Only used when topGeneSets > 0 and the parameter geneSets is specified.

topGeneSetsJust: text justification for the gene sets by default: c(0.5, 0.5) so centered. Only used when topGeneSets > 0, the parameter geneSets is specified and if packageTextLabel is ggplot2.

topGeneSetsColor: color for the gene sets (used when topGeneSets > 0 and geneSets is specified), black by default. Only used when topGeneSets > 0 and the parameter geneSets is specified.

includeLegend: logical if TRUE (by default) include a legend, otherwise not.

includeLineOrigin: if TRUE (by default) include vertical line at x = 0 and horizontal line at y = 0.

typePlot: type of the plot returned, either 'static' (static) or interactive' (potentially interactive).

packageInteractivity: if typePlot is 'interactive', package used for interactive plot, either 'rbokeh' (by default) or 'ggvis'.

figInteractiveSize: vector containing the size of the interactive plot, as [width, height] by default: c(600, 400). This is passed to the width and height parameters of:

- for rbokeh plots: the bokeh::figure function
- for ggvis plots: the ggvis::set_options function
esetLda

ggvisAdjustLegend
  logical, if TRUE (by default) adjust the legends in ggvis to avoid overlapping legends when multiple legends

interactiveTooltip
  logical, if TRUE, add hoover functionality showing sample annotation (variables used in the plot) in the plot

interactiveTooltipExtraVars
  name of extra variable(s) (in varLabels of the eset) to add in tooltip to label the samples, NULL by default

returnAnalysis
  logical, if TRUE (FALSE by default), return also the output of the analysis, and the outlying samples in the topElements element if any, otherwise only the plot object

Value

if returnAnalysis is TRUE, return a list:
  • analysis: output of the spectral map analysis, whose parameters can be given as input to the esetPlotWrapper function
    – dataPlotSamples: coordinates of the samples
    – dataPlotGenes: coordinates of the genes
    – esetUsed: expressionSet used in the plot
  • topElements: list with top outlying elements if any, possibly genes, samples and gene sets
  • plot: the plot output

otherwise return only the plot

Author(s)

Laure Cougnaud

References


See Also

the function used internally: lda

Examples

# load data
library(ALL)
data(ALL)

# specify several variables in ldaVar (this might take a few minutes to run...)

# sample subsetting: currently cannot deal with missing values
samplesToRemove <- which(apply(pData(ALL)[, c("sex", "BT")], 1, anyNA))

# extract random features, because analysis is quite time consuming
retainedFeatures <- sample(featureNames(ALL), size = floor(nrow(ALL)/5))
# create the plot
esetLda(eset = ALL[retainedFeatures, -samplesToRemove],
    ldaVar = "BT", colorVar = "BT", shapeVar = "sex", sizeVar = "age",
    title = "Linear discriminant analysis on the ALL dataset")

esetPlotWrapper  wrapper for biplot of features/samples contained in a eSet object

Description
Wrapper function used for all plots of the visualizations contained in the package.

Usage
esetPlotWrapper(dataPlotSamples, dataPlotGenes = NULL, esetUsed, xlab = "", ylab = "", colorVar = NULL, color = if (is.null(colorVar)) "black" else NULL, shapeVar = NULL, shape = if (is.null(shapeVar)) 15 else NULL, sizeVar = NULL, size = if (is.null(sizeVar)) 2.5 else NULL, sizeRange = NULL, alphaVar = NULL, alpha = if (is.null(alphaVar)) 1 else NULL, alphaRange = NULL, title = "", symmetryAxes = c("combine", "separate", "none"), cloudGenes = TRUE, cloudGenesColor = "black", cloudGenesNBins = if (!is.null(dataPlotGenes)) sqrt(nrow(dataPlotGenes)) else NULL, cloudGenesIncludeLegend = FALSE, cloudGenesTitleLegend = "nGenes", packageTextLabel = c("ggrepel", "ggplot2"), topGenes = 10, topGenesCex = 2.5, topGenesVar = NULL, topGenesJust = c(0.5, 0.5), topGenesColor = "black", geneSets = list(), geneSetsVar = NULL, geneSetsMaxNChar = NULL, topGeneSets = 10, topGeneSetsCex = 2.5, topGeneSetsJust = c(0.5, 0.5), topGeneSetsColor = "black", includeLegend = TRUE, includeLineOrigin = TRUE, typePlot = c("static", "interactive"), figInteractiveSize = c(600, 400), ggvisAdjustLegend = TRUE, interactiveTooltip = TRUE, interactiveTooltipExtraVars = NULL, packageInteractivity = c("rbokeh", "ggvis"), returnTopElements = FALSE)

Arguments
dataPlotSamples  data.frame with columns 'X', 'Y' with coordinates for the samples and with rownames which should correspond and be in the same order as the sample-Names of esetUsed
dataPlotGenes  data.frame with two columns 'X' and 'Y' with coordinates for the genes
esetUsed  expressionSet (or SummarizedExperiment) object with data
xlab  label for the x axis
ylab  label for the y axis
colorVar  name of variable (in varLabels of the eset) used for coloring, NULL by default

color  specified color(s) for the points, replicated if needed, used only if colorVar is NULL, a factor or character by default: 'black' if colorVar is not specified and default ggplot palette otherwise
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shapeVar</td>
<td>name of variable (in varLabels of the eset) used for the shape, NULL by default</td>
</tr>
<tr>
<td>shape</td>
<td>specified shape(s) (pch) for the points, replicated if needed, used only if shapeVar is NULL, a factor or character by default: '15' (filled square) if shapeVar is not specified and default ggplot shape(s) otherwise</td>
</tr>
<tr>
<td>sizeVar</td>
<td>name of variable (in varLabels of the eset) used for the size, NULL by default</td>
</tr>
<tr>
<td>size</td>
<td>specified size(s) (cex) for the points, replicated if needed, used only if sizeVar is NULL, a factor or character by default: '2.5' if sizeVar is not specified and default ggplot size(s) otherwise</td>
</tr>
<tr>
<td>sizeRange</td>
<td>size (cex) range used in the plot, possible only if the sizeVar is 'numeric' or 'integer'</td>
</tr>
<tr>
<td>alphaVar</td>
<td>name of variable (in varLabels of the eset) used for the transparency, NULL by default. This parameter is currently only available for static plot.</td>
</tr>
<tr>
<td>alpha</td>
<td>specified transparency(s) for the points, replicated if needed, used only if shapeVar is NULL, a factor or character by default: '1' if alphaVar is not specified and default ggplot alpha otherwise This parameter is currently only available for static plot.</td>
</tr>
<tr>
<td>alphaRange</td>
<td>transparency (alpha) range used in the plot, possible only if the alphaVar is 'numeric' or 'integer' This parameter is currently only available for static plot.</td>
</tr>
<tr>
<td>title</td>
<td>plot title, &quot; &quot; by default</td>
</tr>
<tr>
<td>symmetryAxes</td>
<td>set symmetry for axes, either:</td>
</tr>
<tr>
<td></td>
<td>• 'combine' (by default): both axes are symmetric and with the same limits</td>
</tr>
<tr>
<td></td>
<td>• 'separate': each axis is symmetric and has its own limits</td>
</tr>
<tr>
<td></td>
<td>• 'none': axes by default (plot limits)</td>
</tr>
<tr>
<td>cloudGenes</td>
<td>logical, if TRUE (by default), include the cloud of genes in the spectral map</td>
</tr>
<tr>
<td>cloudGenesColor</td>
<td>if cloudGenes is TRUE, color for the cloud of genes, black by default</td>
</tr>
<tr>
<td>cloudGenesNBins</td>
<td>number of bins to used for the clouds of genes, by default the square root of the number of genes</td>
</tr>
<tr>
<td>cloudGenesIncludeLegend</td>
<td>logical, if TRUE (FALSE by default) include the legend for the cloud of genes (in the top position if multiple legends)</td>
</tr>
<tr>
<td>cloudGenesTitleLegend</td>
<td>string with title for the legend for the cloud of genes 'nGenes' by default</td>
</tr>
<tr>
<td>packageTextLabel</td>
<td>package used to label the outlying genes/samples/gene sets, either ggrepel (by default, only used if package ggrepel is available), or ggplot2</td>
</tr>
<tr>
<td>topGenes</td>
<td>numeric indicating which percentile (if &lt;1) or number (if &gt;=1) of genes most distant to the origin of the plot to annotate, by default: 10 genes are selected If no genes should be annotated, set this parameter to 0 Currently only available for static plot.</td>
</tr>
<tr>
<td>topGenesCex</td>
<td>cex for gene annotation (used when topGenes &gt; 0)</td>
</tr>
<tr>
<td>topGenesVar</td>
<td>variable of the featureData used to label the genes, by default: NULL, the featureNames are used for labelling (used when topGenes &gt; 0)</td>
</tr>
<tr>
<td>topGenesJust</td>
<td>text justification for the genes (used when topGenes &gt; 0 and if packageTextLabel is ggplot2), by default: c(0.5, 0.5) so centered</td>
</tr>
</tbody>
</table>
**esetPlotWrapper**

topGenesColor text color for the genes (used when topGenes > 0), black by default

topSamples numeric indicating which percentile (if <1) or number (if >=1) of samples most distant to the origin of the plot to annotate, by default: 10 samples are selected
If no samples should be annotated, set this parameter to 0. Currently available for static plot.

topSamplesCex cex for sample annotation (used when topSamples > 0)

topSamplesVar variable of the phenoData used to label the samples, by default: NULL, the sampleNames are used for labelling (used when topSamples > 0)

topSamplesJust text justification for the samples (used when topSamples > 0 and if packageTextLabel is ggplot2), by default: c(0.5, 0.5) so centered

topSamplesColor text color for the samples (used when topSamples > 0), black by default

geneSets list of gene sets/pathways, each containing identifiers of genes contained in the set. E.g. pathways from Gene Ontology databases output from the getGeneSetsForPlot function or any custom list of pathways. The genes identifiers should correspond to the variable geneSetsVar contained in the phenoData, if not specified the featureNames are used. If several gene sets have the same name, they will be combine to extract the top gene sets.

geneSetsVar variable of the featureData used to match the genes contained in geneSets, most probably ENTREZID, if not specified the featureNames of the eSet are used
Only used when topGeneSets > 0 and the parameter geneSets is specified.

geneSetsMaxNChar maximum number of characters for pathway names, by default keep entire names
Only used when topGeneSets > 0 and the parameter geneSets is specified. If returnAnalysis is set to TRUE and geneSetsMaxNChar specified, the top pathways will be returned in the output object, named with the identifiers used in the plot (so with maximum geneSetsMaxNChar number of characters)

topGeneSets numeric indicating which percentile (if <1) or number (if >1) of gene sets most distant to the origin of the plot to annotate, by default: 10 gene sets are selected
If no gene sets should be annotated, set this parameter to 0. Currently available for static plot. Only used when topGeneSets > 0 and the parameter geneSets is specified.

topGeneSetsCex cex for gene sets annotation Only used when topGeneSets > 0 and the parameter geneSets is specified.

topGeneSetsJust text justification for the gene sets by default: c(0.5, 0.5) so centered Only used when topGeneSets > 0, the parameter geneSets is specified and if packageTextLabel is ggplot2.

topGeneSetsColor color for the gene sets (used when topGeneSets > 0 and geneSets is specified), black by default Only used when topGeneSets > 0 and the parameter geneSets is specified.

includeLegend logical if TRUE (by default) include a legend, otherwise not

includeLineOrigin if TRUE (by default) include vertical line at x = 0 and horizontal line at y = 0

typePlot type of the plot returned, either 'static' (static) or interactive’ (potentially interactive)
esetPlotWrapper

figInteractiveSize
vector containing the size of the interactive plot, as [width, height] by default: c(600, 400). This is passed to the width and height parameters of:
  • for rbokeh plots: the bokeh::figure function
  • for ggvis plots: the ggvis::set_options function

ggvisAdjustLegend
logical, if TRUE (by default) adjust the legends in ggvis to avoid overlapping legends when multiple legends

interactiveTooltip
logical, if TRUE, add hover functionality showing sample annotation (variables used in the plot) in the plot

interactiveTooltipExtraVars
name of extra variable(s) (in varLabels of the eset) to add in tooltip to label the samples, NULL by default

packageInteractivity
if typePlot is 'interactive', package used for interactive plot, either 'rbokeh' (by default) or 'ggvis'

returnTopElements
logical, if TRUE (FALSE by default) return the outlying elements labelled in the plot (if any)

Value

if typePlot is:
  • static:
    – if returnTopElements is TRUE, and top elements can be displayed, a list with:
      * 'topElements': the top elements labelled in the plot
      * 'plot': the ggplot object
    – otherwise, the ggplot object only
  • interactive: a ggvis or rbokeh object, depending on the packageInteractivity parameter

Author(s)
Laure Cougnaud

Examples

library(ALL)
data(ALL)

## run one spectral map analysis

# create custom color palette
colorPalette <- c("dodgerblue", colorRampPalette(c("white", "dodgerblue2", "darkblue"))(5)[-1], "red", colorRampPalette(c("white", "red3", "darkred"))(5)[-1])

# run the analysis
# with 'returnAnalysis' set to TRUE to have all objects required for the esetPlotWrapper
outputEsetSPM <- esetSpectralMap(eset = ALL, title = "Acute lymphoblastic leukemia dataset \nSpectral map complete", returnTopElements = TRUE)
esetSpectralMap

esetSpectralMap is a function in the eSet package for R that performs spectral mapping of gene expression data. It reduces the dimension of the data contained in the eSet with the `mpm` function and plots the subsequent biplot of the specified dimensions, possibly with gene and sample annotation contained in the eSet. A spectral map with the default parameters is equivalent to a principal component analysis on the log-transformed, double centered and global normalized data (from documentation of the `mpm` function).

### Usage

```r
esetSpectralMap(eset, psids = 1:nrow(eset), dim = c(1, 2),
                  colorVar = NULL, color = if (is.null(colorVar)) "black" else NULL,
                  shapeVar = NULL, shape = if (is.null(shapeVar)) 15 else NULL,
                  sizeVar = NULL, size = if (is.null(sizeVar)) 2.5 else NULL,
                  sizeRange = NULL, alphaVar = NULL, alpha = if (is.null(alphaVar)) 1 else NULL,
                  alphaRange = NULL, title = "", mpm.args = list(closure = "none",
                  center = "double", normal = "global", row.weight = "mean", col.weight = "constant",
                  logtrans = FALSE), plot.mpm.args = list(scale = "uvc"),
                  symmetryAxes = c("combine", "separate", "none"),
                  packageTextLabel = c("ggrepel", "ggplot2"), cloudGenes = TRUE,
                  cloudGenesColor = "black", cloudGenesNBins = sqrt(length(psids)),
                  cloudGenesIncludeLegend = FALSE, cloudGenesTitleLegend = "nGenes",
                  topGenes = 10, topGenesCex = 2.5, topGenesVar = NULL,
                  topGenesJust = c(0.5, 0.5), topGenesColor = "black",
                  topSamples = 10, topSamplesCex = 2.5, topSamplesVar = NULL,
                  topSamplesJust = c(0.5, 0.5), topSamplesColor = "black",
                  geneSets = list(), geneSetsVar = NULL)
```

### Description

esetSpectralMap reduces the dimension of the data contained in the eSet with the `mpm` function and plots the subsequent biplot of the specified dimensions, possibly with gene and sample annotation contained in the eSet. A spectral map with the default parameters is equivalent to a principal component analysis on the log-transformed, double centered and global normalized data (from documentation of the `mpm` function).
geneSetsMaxNChar = NULL, topGeneSets = 10, topGeneSetsCex = 2.5, 
topGeneSetsJust = c(0.5, 0.5), topGeneSetsColor = "black",  
includeLegend = TRUE, includeLineOrigin = TRUE, typePlot = c("static",  
"interactive"), packageInteractivity = c("rbokeh", "ggvis"),  
figInteractiveSize = c(600, 400), ggvisAdjustLegend = TRUE,  
interactiveTooltip = TRUE, interactiveTooltipExtraVars = NULL,  
returnAnalysis = FALSE)

Arguments

eset                expressionSet (or SummarizedExperiment) object with data
psids               featureNames of genes to include in the plot, all by default
dim                 dimensions of the analysis to represent, first two dimensions by default
colorVar            name of variable (in varLabels of the eset) used for coloring, NULL by default
color               specified color(s) for the points, replicated if needed, used only if colorVar is NaN, a factor or character by default: ‘black’ if colorVar is not specified and default ggplot palette otherwise
shapeVar            name of variable (in varLabels of the eset) used for the shape, NULL by default
shape               specified shape(s) (pch) for the points, replicated if needed, used only if shapeVar is NULL, a factor or character by default: ‘15’ (filled square) if shapeVar is not specified and default ggplot shape(s) otherwise
sizeVar             name of variable (in varLabels of the eset) used for the size, NULL by default
size                specified size(s) (cex) for the points, replicated if needed, used only if sizeVar is NULL, a factor or character by default: ‘2.5’ if sizeVar is not specified and default ggplot size(s) otherwise
sizeRange           size (cex) range used in the plot, possible only if the sizeVar is ‘numeric’ or ‘integer’
alphaVar            name of variable (in varLabels of the eset) used for the transparency, NULL by default. This parameter is currently only available for static plot.
alpha               specified transparency(s) for the points, replicated if needed, used only if shapeVar is NULL, a factor or character by default: ‘1’ if alphaVar is not specified and default ggplot alpha otherwise This parameter is currently only available for static plot.
alphaRange          transparency (alpha) range used in the plot, possible only if the alphaVar is ‘numeric’ or ‘integer’ This parameter is currently only available for static plot.
title               plot title, ” by default
mpm.args            list with input parameters for the mpm function. The default value is: list(closure = 'none', center = TRUE). This assumes that the data are already in a log scale.
plot.mpm.args       list with input parameters for the plot.mpm function. The default value is: list(scale = "uvc").
symmetryAxes        set symmetry for axes, either:
                     • ‘combine’ (by default): both axes are symmetric and with the same limits
                     • ‘separate’: each axis is symmetric and has its own limits
                     • ‘none’: axes by default (plot limits)
packageTextLabel    package used to label the outlying genes/samples/gene sets, either ggrepel (by default, only used if package ggrepel is available), or ggplot2
cloudGenes logical, if TRUE (by default), include the cloud of genes in the spectral map

cloudGenesColor
  if cloudGenes is TRUE, color for the cloud of genes, black by default

cloudGenesNBins
  number of bins to used for the clouds of genes, by default the square root of the number of genes

cloudGenesIncludeLegend
  logical, if TRUE (FALSE by default) include the legend for the cloud of genes (in the top position if multiple legends)

cloudGenesTitleLegend
  string with title for the legend for the cloud of genes 'nGenes' by default

topGenes numeric indicating which percentile (if <1) or number (if >=1) of genes most distant to the origin of the plot to annotate, by default: 10 genes are selected If no genes should be annotated, set this parameter to 0 Currently only available for static plot.

topGenesCex cex for gene annotation (used when topGenes > 0)

topGenesVar variable of the featureData used to label the genes, by default: NULL, the featureNames are used for labelling (used when topGenes > 0)

topGenesJust text justification for the genes (used when topGenes > 0 and if packageTextLabel is ggplot2), by default: c(0.5, 0.5) so centered

topGenesColor text color for the genes (used when topGenes > 0), black by default

topSamples numeric indicating which percentile (if <1) or number (if >=1) of samples most distant to the origin of the plot to annotate, by default: 10 samples are selected If no samples should be annotated, set this parameter to 0. Currently available for static plot.

topSamplesCex cex for sample annotation (used when topSamples > 0)

topSamplesVar variable of the phenoData used to label the samples, by default: NULL, the sampleNames are used for labelling (used when topSamples > 0)

topSamplesJust text justification for the samples (used when topSamples > 0 and if packageTextLabel is ggplot2), by default: c(0.5, 0.5) so centered

topSamplesColor text color for the samples (used when topSamples > 0), black by default

geneSets list of gene sets/pathways, each containing identifiers of genes contained in the set. E.g. pathways from Gene Ontology databases output from the getGeneSetsForPlot function or any custom list of pathways. The genes identifiers should correspond to the variable geneSetsVar contained in the phenoData, if not specified the featureNames are used. If several gene sets have the same name, they will be combine to extract the top gene sets.

geneSetsVar variable of the featureData used to match the genes contained in geneSets, most probably ENTREZID, if not specified the featureNames of the eSet are used Only used when topGeneSets > 0 and the parameter geneSets is specified.

geneSetsMaxNChar maximum number of characters for pathway names, by default keep entire names Only used when topGeneSets > 0 and the parameter geneSets is specified. If returnAnalysis is set to TRUE and geneSetsMaxNChar specified, the top pathways will be returned in the output object, named with the identifiers used in the plot (so with maximum geneSetsMaxNChar number of characters)
esetSpectralMap

**topGeneSets**
numeric indicating which percentile (if \( \leq 1 \)) or number (if \( >1 \)) of gene sets most distant to the origin of the plot to annotate, by default: 10 gene sets are selected. If no gene sets should be annotated, set this parameter to 0. Currently available for static plot. Only used when \( \text{topGeneSets} > 0 \) and the parameter \( \text{geneSets} \) is specified.

**topGeneSetsCex**
cex for gene sets annotation Only used when \( \text{topGeneSets} > 0 \) and the parameter \( \text{geneSets} \) is specified.

**topGeneSetsJust**
text justification for the gene sets by default: \( c(0.5, 0.5) \) so centered Only used when \( \text{topGeneSets} > 0 \), the parameter \( \text{geneSets} \) is specified and if \( \text{packageTextLabel} \) is ggplot2.

**topGeneSetsColor**
color for the gene sets (used when \( \text{topGeneSets} > 0 \) and \( \text{geneSets} \) is specified). black by default Only used when \( \text{topGeneSets} > 0 \) and the parameter \( \text{geneSets} \) is specified.

**includeLegend**
logical if TRUE (by default) include a legend, otherwise not

**includeLineOrigin**
if TRUE (by default) include vertical line at \( x = 0 \) and horizontal line at \( y = 0 \)

**typePlot**
type of the plot returned, either 'static' (static) or interactive' (potentially interactive)

**packageInteractivity**
if \( \text{typePlot} \) is 'interactive', package used for interactive plot, either 'rbokeh' (by default) or 'ggvis'

**figInteractiveSize**
vector containing the size of the interactive plot, as [width, height] by default: \( c(600, 400) \). This is passed to the width and height parameters of:
- for rbokeh plots: the bokeh::figure function
- for ggvis plots: the ggvis::set_options function

**ggvisAdjustLegend**
logical, if TRUE (by default) adjust the legends in ggvis to avoid overlapping legends when multiple legends

**interactiveTooltip**
logical, if TRUE, add hoover functionality showing sample annotation (variables used in the plot) in the plot

**interactiveTooltipExtraVars**
name of extra variable(s) (in varLabels of the eset) to add in tooltip to label the samples, NULL by default

**returnAnalysis**
logical, if TRUE (FALSE by default), return also the output of the analysis, and the outlying samples in the topElements element if any, otherwise only the plot object

**Value**
if \( \text{returnAnalysis} \) is TRUE, return a list:
- analysis: output of the spectral map analysis, can be given as input to the esetPlotWrapper function
  - dataPlotSamples: coordinates of the samples
  - dataPlotGenes: coordinates of the genes
esetTsne

- esetUsed: expressionSet used in the plot
- axisLabels: axes labels indicating percentage of variance explained by the selected axes
- axesContributionsPercentages: percentages of variance explained by each axis (not only the ones specified in `dim`)
- topElements: list with top outlying elements if any, possibly genes, samples and gene sets
- plot: the plot output

otherwise return only the plot

Author(s)
Laure Cougnaud

References
Lewi, P.J. (1976). Spectral mapping, a technique for classifying biological activity profiles of chemical compounds. Arzneimittel Forschung (Drug Research), 26, 1295–1300

See Also
the function used internally: `mpm` and `spectralMap` for spectral map in base R graphics

Examples

```r
library(ALL)
data(ALL)

## complete example (most of the parameters are optional)
# create custom color palette
colorPalette <- c("dodgerblue", colorRampPalette(c("white","dodgerblue2","darkblue"))(5)[-1],
"red", colorRampPalette(c("white", "red3", "darkred"))(5)[-1])
# plot the spectral map
print(esetSpectralMap(eset = ALL,
  title = "Acute lymphoblastic leukemia dataset \n Spectral map complete",
  colorVar = "BT", color = colorPalette,
  shapeVar = "sex", shape = 15:16,
  sizeVar = "age", sizeRange = c(2, 6),
  symmetryAxes = "separate",
  topGenes = 10, topGenesJust = c(1, 0), topGenesCex = 2, topGenesColor = "darkgrey",
  topSamples = 15, topSamplesVar = "cod", topSamplesColor = "black",
  topSamplesJust = c(1, 0), topSamplesCex = 3)
)
```

# see vignette for other examples, especially one with gene sets specification

---

**esetTsne**

plot a t-SNE of an **eSet** object

**Description**

esetTsne reduces the dimension of the data contained in the **eSet** via t-Distributed Stochastic Neighbor Embedding with the **Rtsne** function and plot the subsequent biplot, possibly with sample annotation contained in the **eSet**.
esetTsne

Usage

esetTsne(eset, psids = 1:nrow(eset), trace = TRUE, colorVar = NULL,
color = if (is.null(colorVar)) "black" else NULL, shapeVar = NULL,
shape = if (is.null(shapeVar)) 1 else NULL, sizeVar = NULL, size = if
(is.null(sizeVar)) 2.5 else NULL, sizeRange = NULL, alphaVar = NULL,
alpha = if (is.null(alphaVar)) 1 else NULL, alphaRange = NULL,
title = "", Rtsne.args = list(perplexity = floor((ncol(eset) - 1)/3),
theta = 0.5, dims = 2, initial_dims = 50),
fctTransformDataForInputTsne = NULL, symmetryAxes = c("combine",
"separate", "none"), packageTextLabel = c("ggrepel", "ggplot2"),
topSamples = 10, topSamplesCex = 2.5, topSamplesVar = NULL,
topSamplesJust = c(0.5, 0.5), topSamplesColor = "black",
includeLegend = TRUE, includeLineOrigin = TRUE, typePlot = c("static",
"interactive"), packageInteractivity = c("rbokeh", "ggvis"),
figInteractiveSize = c(600, 400), ggvisAdjustLegend = TRUE,
interactiveTooltip = TRUE, interactiveTooltipExtraVars = NULL,
returnAnalysis = FALSE)

Arguments

- **eset**: expressionSet (or SummarizedExperiment) object with data
- **psids**: featureNames of genes to include in the plot, all by default
- **trace**: logical, if TRUE (by default), print some messages during tsne is running
- **colorVar**: name of variable (in varLabels of the eset) used for coloring, NULL by default
- **color**: specified color(s) for the points, replicated if needed, used only if colorVar is
  NULL, a factor or character by default: 'black' if colorVar is not specified and
default ggplot palette otherwise
- **shapeVar**: name of variable (in varLabels of the eset) used for the shape, NULL by default
- **shape**: specified shape(s) (pch) for the points, replicated if needed, used only if shapeVar is
  NULL, a factor or character by default: '15' (filled square) if shapeVar is not specified and default ggplot shape(s) otherwise
- **sizeVar**: name of variable (in varLabels of the eset) used for the size, NULL by default
- **size**: specified size(s) (cex) for the points, replicated if needed, used only if sizeVar is
  NULL, a factor or character by default: '2.5' if sizeVar is not specified and default ggplot size(s) otherwise
- **sizeRange**: size (cex) range used in the plot, possible only if the sizeVar is 'numeric' or
  'integer'
- **alphaVar**: name of variable (in varLabels of the eset) used for the transparency, NULL by default.
  This parameter is currently only available for static plot.
- **alpha**: specified transparency(s) for the points, replicated if needed, used only if alphaVar is
  NULL, a factor or character by default: '1' if alphaVar is not specified and default ggplot alpha otherwise
  This parameter is currently only available for static plot.
- **alphaRange**: transparency (alpha) range used in the plot, possible only if the alphaVar is
  'numeric' or 'integer' This parameter is currently only available for static plot.
- **title**: plot title, " by default
- **Rtsne.args**: arguments for the Rtsne function, by default: perplexity parameter = optimal
  number of neighbours, theta = speed/accuracy trade-off (increase for less accuracy), set to 0.0 for exact TSNE
fctTransformDataForInputTsne
function which transform the data in the eSet object before calling the Rtsne function. This should be a function which takes a matrix as input and return a matrix, e.g. the dist function.

symmetryAxes
set symmetry for axes, either:
  • 'combine' (by default): both axes are symmetric and with the same limits
  • 'separate': each axis is symmetric and has its own limits
  • 'none': axes by default (plot limits)

packageTextLabel
package used to label the outlying genes/samples/gene sets, either ggrepel (by default, only used if package ggrepel is available), or ggplot2

topSamples
numeric indicating which percentile (if <1) or number (if >=1) of samples most distant to the origin of the plot to annotate, by default: 10 samples are selected If no samples should be annotated, set this parameter to 0. Currently available for static plot.

topSamplesCex
cex for sample annotation (used when topSamples > 0)

topSamplesVar
variable of the phenoData used to label the samples, by default: NULL, the sampleNames are used for labelling (used when topSamples > 0)

topSamplesJust
text justification for the samples (used when topSamples > 0 and if packageTextLabel is ggplot2), by default: c(0.5, 0.5) so centered

topSamplesColor
text color for the samples (used when topSamples > 0), black by default

includeLegend
logical if TRUE (by default) include a legend, otherwise not

includeLineOrigin
if TRUE (by default) include vertical line at x = 0 and horizontal line at y = 0

typePlot
type of the plot returned, either 'static' (static) or interactive' (potentially interactive)

packageInteractivity
if typePlot is 'interactive', package used for interactive plot, either 'rbokeh' (by default) or 'ggvis'

figInteractiveSize
vector containing the size of the interactive plot, as [width, height] by default: c(600, 400). This is passed to the width and height parameters of:
  • for rbokeh plots: the bokeh::figure function
  • for ggvis plots: the ggvis::set_options function

ggvisAdjustLegend
logical, if TRUE (by default) adjust the legends in ggvis to avoid overlapping legends when multiple legends

interactiveTooltip
logical, if TRUE, add hoover functionality showing sample annotation (variables used in the plot) in the plot

interactiveTooltipExtraVars
name of extra variable(s) (in varLabels of the eset) to add in tooltip to label the samples, NULL by default

returnAnalysis
logical, if TRUE (FALSE by default), return also the output of the analysis, and the outlying samples in the topElements element if any, otherwise only the plot object
Value

if returnAnalysis is TRUE, return a list:

- analysis: output of the spectral map analysis, whose elements can be given to the \texttt{esetPlotWrapper} function
  - dataPlotSamples: coordinates of the samples
  - esetUsed: expressionSet used in the plot
- topElements: list with top outlying elements if any, possibly genes, samples and gene sets
- plot: the plot output

otherwise return only the plot

Author(s)

Laure Cougnaud

References


See Also

the function used internally: \texttt{Rtsne} or \url{http://homepage.tudelft.nl/19j49/t-SNE.html} for further explanations about this technique.

Examples

library(ALL)
data(ALL)

## complete example (most of the parameters are optional)

# create custom color palette
colorPalette <- c("dodgerblue", colorRampPalette(c("white", "dodgerblue2", "darkblue"))(5)[-1],
"red", colorRampPalette(c("white", "red3", "darkred"))(5)[-1])

# create tsne
print(esetTsne(eset = ALL,
title = "Acute lymphoblastic leukemia dataset \n Tsne complete",
colorVar = "BT", color = colorPalette,
shapeVar = "sex", shape = 15:16,
sizeVar = "age", sizeRange = c(2, 6),
symmetryAxes = "separate",
topSamples = 15, topSamplesVar = "cod", topSamplesColor = "black",
topSamplesJust = c(1, 0), topSamplesCex = 3)
)}
**getCoordGeneSets**  
*extract coordinates gene sets*

**Description**

extract coordinates gene sets

**Usage**

`getCoordGeneSets(dataPlotGenes, geneSets, esetUsed, geneSetsVar = NULL)`

**Arguments**

- `dataPlotGenes`: data.frame with two columns 'X' and 'Y' with coordinates for the genes
- `geneSets`: geneSets list of gene sets, e.g. gene pathways, output from the 'getGeneSets' function in MLP the genes IDs must correspond to the sampleNames in the eset
- `esetUsed`: expressionSet (or SummarizedExperiment) object with data
- `geneSetsVar`: variable of the featureData used to match the genes contained in geneSets, most probably ENTREZID, if NULL the featureNames of the eSet are used

**Value**

data.frame with two columns 'X' and 'Y' with coordinates for the gene sets

**Author(s)**

Laure Cougnaud
Laure Cougnaud

---

**getGeneSetsForPlot**  
*get gene sets for plot of eSet object.*

**Description**

get and format gene sets to be used as geneSets for the functions: `esetSpectralMap`, `esetLda`, or `esetPlotWrapper`. Use the `getGeneSets` function to get the gene sets, combine all databases, and format the gene sets name if required.

**Usage**

```r
getGeneSetsForPlot(entrezIdentifiers, species = "Human", geneSetSource = c("GOBP", "GOMF", "GOCC", "KEGG"), useDescription = TRUE, trace = TRUE)
```
getGeneSetsForPlot

Arguments

entrezIdentifiers    
string with Entrez Gene identifiers of the genes of interest

species              
species to use, given to the getGeneSets function

geneSetSource        
gene set source, either 'GOBP', 'GOMF', 'GOCC' or 'KEGG'. Multiple choices are available

useDescription       
logical, if TRUE (by default) use the description to label the gene sets, otherwise use the original gene set identifiers Function 'substr' is used.

trace                 
logical, if TRUE (by default) a few extra information are printed during the process

Value

list with gene sets, each element is a gene set and contains the ENTREZ IDs of the genes contained in this set. If useDescription is:

• FALSE: pathways are labelled with identifiers (Gene Ontology IDs for GOBP, GOMF and GOCC, KEGG IDs for KEGG)
• TRUE: pathways are labelled with gene sets descriptions

Author(s)

Laure Cougnaud

See Also

the function used internally: getGeneSets

Examples

# example dataset
library(ALL)
data(ALL)

# get gene annotation from probe IDs
library("hgu95av2.db")
probeIDs <- featureNames(ALL)
geneInfo <- select(hgu95av2.db, probeIDs, "ENTREZID", "PROBEID")

# get pathway annotation for the genes contained in the ALL dataset (can take a few minutes)
geneSets <- getGeneSetsForPlot(entrezIdentifiers = geneInfo$ENTREZID, species = "Human",
genеSetSource = 'GOBP',
useDescription = FALSE, trace = TRUE)
head(geneSets) # returns a pathway list of genes

# gene sets labelled with gene sets description

geneSets <- getGeneSetsForPlot(entrezIdentifiers = geneInfo$ENTREZID, species = "Human",
genеSetSource = 'GOBP', useDescription = TRUE, trace = TRUE)
head(geneSets) # returns a pathway list of genes

# see also vignette for an example of the use of this function as input for the esetSpectralMap, esetLda or esetPlotWrapper functions.
getMethodsInputObjectEsetVis

wrapper to extract useful functions, depending if the object is an ExpressionSet or a SummarizedExperiment.

Description

This returns an error is x is not of the correct class. The package SummarizedExperiment should be available if x is of class SummarizedExperiment.

Usage

getMethodsInputObjectEsetVis(x)

Arguments

x  
  object

Value

if the object is an ExpressionSet or a SummarizedExperiment, returns a list with the functions specific of the class of x, and equivalent of the ExpressionSet functions: 'sampleNames', 'featureNames', 'fData', 'pData', 'exprs'
  - sampleNames: sample names
  - featureNames: feature names
  - fData: feature annotation
  - pData: sample annotation
  - exprs: data matrix
  - varLabels: sample annotation variables
  - fvarLabels: feature annotation variables

Author(s)

Laure Cougnaud

plotTopElements  

create geom_text object with top genes/sample/pathways

Description

create geom_text object with top genes/sample/pathways

Usage

plotTopElements(top, type = c("gene", "sample", "geneSets"), var = NULL, cex = 1, just = c(0.5, 0.5), color = "black", dataPlotGenes = NULL, dataPlotSamples = NULL, esetUsed, geneSets = NULL, geneSetsVar = NULL, geneSetsMaxNChar = NULL, returnTopElements = FALSE, packageTextLabel = c("ggrepel", "ggplot2"))
**plotTopElements**

**Arguments**

- `top` numeric, number of top elements
- `type` type of elements to plot, either 'gene', 'sample', or 'geneSets'
- `var` variable used to annotate the elements, only used for 'gene' and 'sample'
- `cex` cex of text in the plot
- `just` justification of elements in the plot, only use if packageTextLabel is 'ggplot2'
- `color` color for the elements in the plot
- `dataPlotGenes` data.frame with two columns 'X' and 'Y' with coordinates for the genes
- `dataPlotSamples` data.frame with two columns 'X' and 'Y' with coordinates for the samples
- `esetUsed` expressionSet (or SummarizedExperiment) object with data
- `geneSets` list of gene sets, e.g. gene pathways, output from the 'getGeneSets' function in MLP the genes IDs must correspond to the sampleNames in the eset. If several gene sets have the same name, they will be combine to extract the top gene sets.
- `geneSetsVar` variable of the featureData used to match the genes contained in geneSets, most probably ENTREZID, if not specified the featureNames of the eSet are used
- `geneSetsMaxNChar` maximum number of characters for pathway names, by default keep entire names
- `returnTopElements` logical if TRUE (FALSE by default) return the outlying elements
- `packageTextLabel` package used to label the outlying genes/samples/gene sets, either 'ggrepel' (by default, only used if package ggrepel is available), or 'ggplot2'

**Value**

- if the elements are present in the data: if returnTopElements is:
  - TRUE: return a list with two arguments:
    * topElements: string with top elements labelled in the plot
    * geomText: output of geom_text
  - FALSE: only return the output of geom_text
- if not, return NULL

**Author(s)**

Laure Cougnaud
Index

eSet, 2, 6, 10, 14, 18
esetLda, 2, 18
esetPlotWrapper, 5, 6, 17, 18
esetSpectralMap, 10, 18
esetTsne, 14

getCoordGeneSets, 18
geneSets, 18, 19
geneSetsForPlot, 4, 8, 12, 18
getMethodsInputObjectEsetVis, 20

lda, 5

mpm, 10, 11, 14

plot.mpm, 11
plotTopElements, 20

Rtsne, 14, 16, 17

spectralMap, 14