Package 'RGenEDA'

November 12, 2025

Title Genomic Exploratory Data Analysis

Version 1.0.2

Description Provides a unified and reproducible framework for performing genomic exploratory data analysis (EDA). Highly repetitive functions and analyses have been collated into an easy-to-use S4 object that produces quick results and customizable visualizations to assess variation in your data. See <https://github.com/mikemartinez99/RGenEDA>.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends methods, R (>= 4.1.0)

Imports pheatmap, RColorBrewer, dplyr, ggplot2, ggrepel, grid, scales, tidyr, vegan, stats

Suggests DESeq2, ggtext, pasilla, tidyverse, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Michael Martinez [aut, cre]

Maintainer Michael Martinez <f007qps@dartmouth.edu>

Contents

`DEGs` *Access DEGs container*

Description

Access DEGs container

Usage

```
DEGs(object, assay)
```

Arguments

`object`	A geneda object
`assay`	The name of the DEGs slot to return

Value

List with `DEG` (unfiltered data.frame) and optionally named filtered result sets

---

`DimReduction` *Access dimensional reduction results list*

Description

Access dimensional reduction results list

Usage

```
DimReduction(object)
```

Arguments

`object`	A geneda object

Value

List with `Loadings`, `Eigenvectors`, `percent_var`

distanceHeatmap *distanceHeatmap*

Description

Compute a sample-by-sample distance matrix from a geneda object and plot a ggplot2 heatmap. Optionally subset metadata columns and reorder rows/columns by hierarchical clustering.

Usage

```
distanceHeatmap(
  object,
  method = "euclidean",
  reorder = TRUE,
  meta_cols = NULL,
  palettes = NULL,
  return = c("object", "plot")
)
```

Arguments

object	A geneda object
method	Distance method for stats::dist (default "euclidean")
reorder	Logical; if TRUE, reorder rows/cols by hclust on the distance matrix. Default TRUE
meta_cols	Optional character vector of metadata columns to display in the title/subtitle for quick reference (no annotation strips are drawn).
palettes	Optional named list of color vectors that can be used downstream (returned) if the caller wants to annotate elsewhere.
return	One of c("object", "plot"). If "plot", draws the heatmap immediately via grid and still returns the result list. Default "object".

Value

A list with elements: dist_matrix, order (character vector of sample names), heatmap (pheatmap object), palettes (as passed through)

Examples

```
mock_norm <- matrix(rnorm(12, mean = 0, sd = 2), nrow = 4, ncol = 3)
colnames(mock_norm) <- paste0("Sample", 1:3)
rownames(mock_norm) <- paste0("Gene", 1:4)
mock_meta <- data.frame(condition = c("A","B","A"), row.names = colnames(mock_norm))
obj <- GenEDA(normalized = mock_norm, metadata = mock_meta)
colorList <- list(condition = c("A" = "red", "B" = "blue"))
distanceHeatmap(
    obj,
    meta_cols = c("condition"),
    palettes = colorList,
    return = "plot")
```

eigencorr *eigencorr*

Description

Calculate Eigen-correlations from a geneda object and plot a publication-quality heatmap using ggplot2 with numeric labels and significance stars. For continuous variables, Pearson correlation is used. P values follow this convention: p < 0.001, p < 0.01, p < 0.05 = three, two, one stars, respectively.

Requires PCA loadings in DimReduction(object); does not recompute PCA.

Usage

```
eigencorr(object, num_pcs = 10, meta_cols = NULL)
```

Arguments

object	A geneda object containing normalized and metadata, and optionally DimReduction loadings.
num_pcs	Number of principal components to correlate.
meta_cols	Optional character vector of metadata column names to include. Defaults to all metadata columns.

Value

A list with elements: cor_matrix, pval_matrix, stars, plot (ggplot)

Examples

```
mock_norm <- matrix(rnorm(5000 * 6, mean = 0, sd = 2), nrow = 5000, ncol = 6)
colnames(mock_norm) <- paste0("Sample", 1:6)
rownames(mock_norm) <- paste0("Gene", 1:5000)

# Sample metadata
mock_meta <- data.frame(condition = c("A","B","A","B","A","B"),
row.names = colnames(mock_norm))

# Construct GenEDA object
obj <- GenEDA(normalized = mock_norm, metadata = mock_meta)

# Run PCA
obj <- RunPCA(obj)

# Compute eigencorrelation plot using first 5 PCs
ec <- eigencorr(obj, num_pcs = 5)
ec$plot
```

extractEigen *extractEigen*

Description

Extract gene-level Eigen-vectors for PCs of interest and calculate percent variance.

Usage

```
extractEigen(object, component)
```

Arguments

object	A geneda object containing PCA results in the DimReduction slot
component	Principal component to use (i.e., PC1)

Value

A data-frame containing gene loading information and percent variation per gene

Examples

```
mock_norm <- matrix(rnorm(12, mean = 0, sd = 2), nrow = 4, ncol = 3)
colnames(mock_norm) <- paste0("Sample", 1:3)
rownames(mock_norm) <- paste0("Gene", 1:4)
mock_meta <- data.frame(condition = c("A","B","A"), row.names = colnames(mock_norm))
obj <- GenEDA(normalized = mock_norm, metadata = mock_meta)
extractEigen(object = obj, component = "PC1")
```

ExtractPCA *Extract PCA Loadings and Metadata*

Description

This function extracts PCA loadings stored in the DimReduction slot of a geneda object and combines them with the associated metadata. It ensures that the metadata has valid rownames and aligns the PCA loadings accordingly.

Usage

```
ExtractPCA(object)
```

Arguments

object	A geneda object containing PCA results in the DimReduction slot and sample-level metadata in the metadata slot.

Details

The function performs several checks:

- Ensures the input object is of class geneda.
- Verifies that the DimReduction slot contains PCA loadings.
- Confirms that the metadata has valid rownames.
- Reorders the PCA loadings to match the order of metadata rows.

If metadata rownames are missing or invalid, the function throws an error.

Value

A data.frame combining PCA loadings and sample metadata, where rows correspond to samples and columns include principal component loadings and metadata fields.

Examples

```
## Not run:
# Example usage:
pca_results <- ExtractPCA(my_geneda_object)
head(pca_results)

## End(Not run)
```

FilterDEGs *Filter DEGs by padj and absolute log2FoldChange*

Description

Filters the unfiltered DEGs in DEGs$DEG and stores the filtered results in a new named slot DEGs[[assayName]]. Multiple filtered result sets can be stored with different assay names.

Usage

```
FilterDEGs(object, padj_thresh = 0.05, log2FC_thresh = 1, assayName)
```

Arguments

object	A geneda object with DEGs$DEG set
padj_thresh	Adjusted p-value threshold (<=)
log2FC_thresh	Absolute log2 fold change threshold (>=)
assayName	Character name for the filtered result set (e.g., "padj05_lfc1")

Value

Updated geneda object with filtered results stored in DEGs[[assayName]]

FindVariableFeatures *Find and store highly variable genes (HVGs)*

Description

Computes per-feature variance on `normalized(object)`, ranks genes by decreasing variance, and stores the top `nfeatures` gene IDs into the `HVGs` slot. Returns the updated object.

Usage

```
FindVariableFeatures(object, nfeatures)
```

Arguments

`object`	A geneda object
`nfeatures`	Number of HVGs to store

Value

Updated geneda object

GenEDA *Construct a geneda object*

Description

Construct a geneda object

Usage

```
GenEDA(normalized, metadata, counts = NULL, DEGs = NULL)
```

Arguments

`normalized`	Normalized expression matrix (features x samples).
`metadata`	Sample metadata `data.frame` with row names matching `colnames(normalized)`.
`counts`	Optional counts matrix (features x samples).
`DEGs`	Optional DEG results from `DESeq2` Note: This constructor does not compute HVGs or PCA. Use `FindVariableFeatures()` and `RunPCA()` afterward.

Value

A geneda object.

geneda-class *geneda S4 class*

Description

An S4 container for exploratory genomic data analysis. Stores optional raw counts, normalized data, sample metadata, highly variable genes (HVGs), and PCA dimensionality reduction results.

Slots

`counts` Optional counts matrix (features x samples). Can be NULL.

`normalized` Normalized expression matrix (features x samples).

`metadata` Sample-level metadata `data.frame` (rows = samples).

`HVGs` Character vector of selected highly variable gene IDs (row names).

`DimReduction` List for PCA results with `Loadings`, `Eigenvectors`, `percent_var`.

`DEGs` List container for differential expression results. Contains `DEG` (data.frame) for unfiltered results, and optionally named slots for filtered results (e.g., `DEGs$assay1`, `DEGs$assay2`).

generatePCs *generatePCs*

Description

Generate principal component analysis data that can be used in downstream analyses.

Usage

```
generatePCs(mat, vars, nFeatures)
```

Arguments

`mat`	A data matrix where rows are features and columns are samples
`vars`	A vector of gene variances (can calculate using RGenEDA::plotVariance)
`nFeatures`	Number of top features to generate principal components on.

Value

A list consisting of 3 slots: Loadings, Eigenvectors, and percent_var

getCounts *Access counts matrix (avoid name clash with DESeq2::counts)*

Description

Access counts matrix (avoid name clash with DESeq2::counts)

Usage

```
getCounts(object)
```

Arguments

object	A geneda object

Value

Matrix or NULL

getMetadata *Access metadata (avoid name clash with S4 generics)*

Description

Access metadata (avoid name clash with S4 generics)

Usage

```
getMetadata(object)
```

Arguments

object	A geneda object

Value

data.frame

getNormalized *Access normalized matrix (avoid name clash with generics)*

Description

Access normalized matrix (avoid name clash with generics)

Usage

```
getNormalized(object)
```

Arguments

object	A geneda object

Value

Matrix

HVGs *Access HVG IDs*

Description

Access HVG IDs

Usage

```
HVGs(object)
```

Arguments

object	A geneda object

Value

Character vector

ordcorr *ordcorr*

Description

Calculate correlations between NMDS ordination axes (from beta diversity distances) and metadata, and plot a ggplot2 heatmap with numeric labels and significance stars.

For continuous variables, Pearson correlation is used. P values follow this convention: $p < 0.001$, $p < 0.01$, $p < 0.05$ = three, two, one stars, respectively.

Default num_mds is 10, but function will internally set max number of NMDS to be n-1 where n is your number of samples.

Usage

```
ordcorr(object, num_mds = 10, meta_cols = NULL, distance = "bray")
```

Arguments

object	A geneda object containing normalized and metadata.
num_mds	Number of NMDS axes to correlate. Default = 10
meta_cols	Optional character vector of metadata column names to include. Defaults to all metadata columns.
distance	Distance metric for vegdist (default "bray").

Value

A list with elements: cor_matrix, pval_matrix, stars, plot (ggplot), and variance explained (squared correlations between ordination distances and observed dissimilarities.)

PlotCountDist *PlotCountDist*

Description

Generates violin plots with embedded boxplots showing the distribution of normalized or VST-transformed RNA-Seq counts for each sample in a geneda object. Optionally, plots can be faceted by a metadata variable.

Usage

```
PlotCountDist(object, split_by = NULL)
```

Arguments

object	A geneda object containing normalized expression values in @normalized and sample metadata in @metadata.
split_by	Character. Optional column name from object@metadata used for faceting (default: NULL, no faceting).

Value

A ggplot2 object displaying the distribution of counts per sample.

Examples

```
## Not run:
# Basic plot without faceting
PlotCountDist(obj)

# Facet by a metadata variable "condition"
PlotCountDist(obj, split_by = "condition")

## End(Not run)


mock_norm <- matrix(rnorm(12, mean = 0, sd = 2), nrow = 4, ncol = 3)
colnames(mock_norm) <- paste0("Sample", 1:3)
rownames(mock_norm) <- paste0("Gene", 1:4)
mock_meta <- data.frame(condition = c("A","B","A"), row.names = colnames(mock_norm))
obj <- GenEDA(normalized = mock_norm, metadata = mock_meta)
PlotCountDist(obj)
```

`PlotEigenHeatmap` *PlotEigenHeatmap*

Description

Generates a pheatmap of the top genes contributing to a specific principal component from a geneda object. Genes are selected by absolute eigenvector loading for the chosen PC, and expression values are Z-score scaled per gene. Optionally annotate columns with metadata variables and add row annotations showing gene-level percent variance.

Usage

```
PlotEigenHeatmap(
  object,
  pc = "PC1",
  n = 25,
  annotate_by = NULL,
  annotate_colors = NULL,
  return = c("object", "plot")
)
```

Arguments

`object`	A geneda object containing normalized expression data and PCA in `DimReduction`.
`pc`	Character. Principal component to visualize (e.g., "PC1").
`n`	Integer. Number of genes to select by absolute loading (default: 25).
`annotate_by`	Character vector of metadata column names for column annotations (optional).
`annotate_colors`	Named list of color vectors for metadata columns. Names should match `annotate_by`.
`return`	One of c("object","plot"). If "plot", draws the heatmap via grid.

Value

A list with topGenes (data.frame with EigenVector and PctVar), expression (scaled matrix), and heatmap (pheatmap object)

Examples

```
## Not run:
mock_norm <- matrix(rnorm(12, mean = 0, sd = 2), nrow = 4, ncol = 3)
colnames(mock_norm) <- paste0("Sample", 1:3)
rownames(mock_norm) <- paste0("Gene", 1:4)
mock_meta <- data.frame(condition = c("A","B","A"), row.names = colnames(mock_norm))
obj <- GenEDA(normalized = mock_norm, metadata = mock_meta)
res <- PlotEigenHeatmap(obj, pc = "PC1", n = 25,
                        annotate_by = c("Condition"),
                    annotate_colors = list(Condition = c("A" = "#1b9e77", "B" = "#d95f02")))
grid::grid.newpage(); grid::grid.draw(res$heatmap$gtable)

## End(Not run)
```

PlotHVGVariance *PlotHVGVariance*

Description

Plot variance for all genes (features) on a geneda object using ggplot2.

Usage

```
PlotHVGVariance(object, transform = NULL, dropTopN = 0)
```

Arguments

object	A geneda object created by GenEDA().
transform	One of c("log") to log-transform variance, or NULL for none. Default = NULL
dropTopN	Number of top-most variable genes to drop prior to plotting. Default = 0

Value

A ggplot2 object visualizing the variance of HVGs

Examples

```
# obj <- GenEDA(vsd, meta)
# p <- PlotHVGVariance(obj, transform = "log")
# print(p)
```

PlotMA *PlotMA*

Description

Plot MA plot from a geneda object. Requires that you have ran SetDEGs on your object following differential expression in DESeq2.

Usage

```
PlotMA(object, alpha, fc, title = NULL)
```

Arguments

object	A geneda object containing DEGs from SetDEGs method
alpha	Threshold for adjusted p-values (padj column from DESeq2)
fc	Absolute value log2Fold-change magnitude threshold (log2FoldChange column)
title	Optional character vector of what plot should be titled.

Value

A ggplot2 object

PlotPCA *Plot PCA results from a GenEDA object*

Description

Visualize principal component analysis (PCA) results stored within a GenEDA object. This function extracts PCA data via ExtractPCA and provides a flexible ggplot2-based visualization interface.

Usage

```
PlotPCA(
  object,
  x = 1,
  y = 2,
  color_by,
  colors = NULL,
  split_by = NULL,
  shape_by = NULL,
  return_data = FALSE
)
```

Arguments

object	A GenEDA object containing PCA results in the DimReduction slot.
x	Numeric or character. The principal component to plot on the x-axis (e.g., 1 or "PC1").
y	Numeric or character. The principal component to plot on the y-axis (e.g., 2 or "PC2").
color_by	Character. Column name in the metadata used to color points.
colors	Vector. Custom colors to use for plotting
split_by	Character (optional). Column name in metadata used for faceting (creates separate panels).
shape_by	Character (optional). Column name in metadata used to control point shape.
return_data	Logical (default = FALSE), whether or not to return pca dataframe for more custom plotting.

Value

A ggplot object displaying the PCA scatter plot, or a list of pca_df and plot if return_data = TRUE

Examples

```
## Not run:
p <- PlotPCA(obj, x = 1, y = 2, color_by = "condition",
     colors = c("untreated" = "red", "treated" = "blue"),
     split_by = "library")
p

## End(Not run)
```

RunPCA *Run PCA and store in* DimReduction

Description

Uses generatePCs() under the hood. If HVGs are empty, selects HVGs via FindVariableFeatures(object, nfeatures) with default nfeatures = 2000. If HVGs are present, the PCA uses NFEATURES = length(HVGs(object)). This aligns the PCA feature count with the HVG selection while allowing an explicit override of nfeatures when empty.

Usage

```
RunPCA(object, nfeatures = 2000)
```

Arguments

object	A geneda object
nfeatures	Number of features to use when HVGs are empty. Default = 2000

Value

Updated geneda object with DimReduction filled

SetDEGs *Set unfiltered DEGs on the object*

Description

Set unfiltered DEGs on the object

Usage

```
SetDEGs(object, deg_table)
```

Arguments

object	A geneda object
deg_table	A data.frame of DESeq2-like results containing at least columns log2FoldChange and padj

Value

Updated geneda object with DEGs$DEG set

show,geneda-method *Show method for geneda*

Description

Show method for geneda

Show method for geneda

Usage

```
## S4 method for signature 'geneda'
show(object)

## S4 method for signature 'geneda'
show(object)
```

Arguments

object	A geneda object

Index