Package 'macrosyntR'

July 22, 2025

Type Package

Title Draw Ordered Oxford Grids and Chord Diagrams

Version 0.3.3

Maintainer Sami El Hilali <elhilali.sami@gmail.com>

Depends R (>= 4.1.0)

Imports stats, utils, ggplot2, igraph, tidyr, reshape2, dplyr, stringr, rlang

Description Use standard genomics file format (BED) and a table of orthologs to illustrate synteny conservation at the genome-wide scale. Significantly conserved linkage groups are identified as described in Simakov et al. (2020) <doi:10.1038/s41559-020-1156-z> and displayed on an Oxford Grid (Edwards (1991) <doi:10.1111/j.1469-1809.1991.tb00394.x>) or a chord diagram as in Simakov et al. (2022) <doi:10.1126/sciadv.abi5884>. The package provides a function that uses a network-based greedy algorithm to find communities (Clauset et al. (2004) <doi:10.1103/PhysRevE.70.066111>) and so automatically order the chromosomes on the plot to improve interpretability.

Encoding UTF-8

License GPL-3

URL https://github.com/SamiLhll/macrosyntR

BugReports https://github.com/SamiLhll/macrosyntR/issues

RoxygenNote 7.2.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Sami El Hilali [aut, cre] (ORCID: <https://orcid.org/0000-0003-4417-8399>), Richard Copley [aut] (ORCID: <https://orcid.org/0000-0001-7846-4954>)

Repository CRAN

Date/Publication 2023-11-14 16:40:02 UTC

Contents

`compute_linkage_groups`

Compute Linkage groups

Description

This is a function to compute the conserved linkage groups shared between two or more species. It computes the significant associations between chromosomes of all species versus all (pairwise) using the fischer test implemented in compute_macrosynteny(). It outputs a dataframe shaped as following : sp1.Chr,sp2.Chr,..., spN.chr,n,LGs where n is the number of shared orthologs in the group and LGs are the IDs for the linkage groups

Usage

```
compute_linkage_groups(orthologs_df)
```

Arguments

`orthologs_df`	dataframe. orthologs with genomic coordinates loaded with load_orthologs()

Value

A dataframe object

Examples

```
# basic usage of compute_linkage_groups:

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_macrosynteny <- compute_linkage_groups(my_orthologs)
```

compute_macrosynteny *Compute significant macrosynteny blocks*

Description

This is a function to generate the contingency table of an orthologs dataframe and apply fischer test to calculate the significant associations. It outputs a dataframe shaped as following : sp1.Chr,sp2.Chr,a,pval,significant,pval_a

Usage

```
compute_macrosynteny(orthologs_df, pvalue_threshold = 0.001)
```

Arguments

orthologs_df	dataframe. orthologs with genomic coordinates loaded with load_orthologs()
pvalue_threshold	numeric. threshold for significancy. (default equals 0.001)

Value

A dataframe object

Examples

```
# basic usage of compute_macrosynteny :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_macrosynteny <- compute_macrosynteny(my_orthologs)
```

get_syntenic_genes *get the syntenic genes as a table*

Description

This is a function to extract all the syntenic genes from an orthologs_df. It requires as input an orthologs_df loaded by load_orthologs().

Usage

```
get_syntenic_genes(orthologs_df)
```

Arguments

orthologs_df	dataframe. orthologs with genomic coordinates loaded by load_orthologs()

Value

dataframe composed of details for each detected syntenic block of genes. It contains the following columns : sp1.Chr, sp1.Start, sp1.End, sp2.Chr, sp2.Start, sp2.End, size, sp1.IDs, sp2.IDs

See Also

load_orthologs()

Examples

```
# basic usage of get_syntenic_genes :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_syntenic_block_of_genes <- get_syntenic_genes(my_orthologs)
```

---

load_orthologs *load orthologs with their genomic coordinates.*

Description

Puts together the table of orthologous genes with their genomic coordinates in the two or more species. It outputs a data.frame shaped as following : sp1.ID,sp1.Chr,sp1.Start,sp1.End,sp1.Index,sp2.ID,sp2.Chr,sp2.Start,sp

Usage

```
load_orthologs(
  orthologs_table,
  sp1_bed = NULL,
  sp2_bed = NULL,
  bedfiles = NULL
)
```

Arguments

orthologs_table	character. Full path to the orthologs table (format : geneID_on_species1 geneID_on_species2 geneID_on_speciesN)
sp1_bed	(deprecated) character. Full path to the genomic coordinates of the genes on species1

sp2_bed	(deprecated) character. Full path to the genomic coordinates of the genes on species2
bedfiles	array. List of full paths to the genomic coordinates ordered as in the appearing order of the orthologs_table (BED format)

Value

dataframe composed of genomic coordinates and relative index of orthologs on both species

Examples

```
# basic usage of load_orthologs for two species :

orthologs_file <- system.file("extdata","Bflo_vs_Pyes.tab",package="macrosyntR")
bedfile_sp1 <- system.file("extdata","Bflo.bed",package="macrosyntR")
bedfile_sp2 <- system.file("extdata","Pyes.bed",package="macrosyntR")


my_orthologs <- load_orthologs(orthologs_table = orthologs_file,
                               bedfiles = c(bedfile_sp1,bedfile_sp2))
# example with 3 species :
orthologs_file <- system.file("extdata","Single_copy_orthologs.tsv",package="macrosyntR")
bedfile_sp3 <- system.file("extdata","Pech.bed",package="macrosyntR")

my_orthologs <- load_orthologs(orthologs_table = orthologs_file,
                               bedfiles = c(bedfile_sp1,bedfile_sp2,bedfile_sp3))
```

---

plot_chord_diagram *plot the Macro-synteny as a chord diagram*

Description

This is a function to plot the chord diagrams to compare the macro synteny of two or more species. It requires as input an orthologs_df loaded by load_orthologs()

Usage

```
plot_chord_diagram(
  orthologs_df,
  species_labels = NULL,
  species_labels_size = 5,
  color_by = "sp1.Chr",
  custom_color_palette = NULL,
  reorder_chromosomes = TRUE,
  remove_non_linkage_orthologs = TRUE,
  species_labels_hpos = -400,
```

```
  label_size = 2,
  ideogram_fill = "white",
  ideogram_color = "black",
  ideogram_height = 4,
  ribbons_curvature = 0.1,
  ribbons_alpha = 0.5
)
```

Arguments

orthologs_df	dataframe. orthologs with genomic coordinates loaded by the load_orthologs()
species_labels	list of characters. names of the species to display on the plot
species_labels_size	integer. size of the labels (default = 2)
color_by	string. name of the column in the orthologs_df to color the links by (default = "sp1.Chr")
custom_color_palette	list of characters. palette to use for the coloring of the links following the argument color_by
reorder_chromosomes	logical. (default = TRUE) tells whether to reorder the chromosomes in clusters as implemented in reorder_macrosynteny()
remove_non_linkage_orthologs	logical. (default = TRUE) tells wether to remove the orthologs that are not within significant linkage groups as calculated by comput_linkage_groups().
species_labels_hpos	(default =-400)
label_size	integer. size of the labels to display on the ideograms (default = 2)
ideogram_fill	character. name of the colors to fill the ideograms with (default = "white")
ideogram_color	character. name of the colors to draw the borders of the ideograms with (default = "black")
ideogram_height	integer. height of the ideograms (default = 4)
ribbons_curvature	float. curvature of the ribbons (default = 0.1)
ribbons_alpha	float. alpha of the ribbons (default = 0.5)

Value

A ggplot2 object

See Also

load_orthologs()

reorder_macrosynteny()

compute_linkage_groups()

Examples

```
# basic usage of plot_oxford_grid :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

plot_chord_diagram(my_orthologs,species_labels = c("B. flo","P. ech"))
```

`plot_macrosynteny` *Plot Macro-synteny*

Description

This is a function to generate the contingency table of an MBH dataframe and apply fischer test to calculate the significant associations.

Usage

```
plot_macrosynteny(macrosynt_df, sp1_label = "", sp2_label = "")
```

Arguments

`macrosynt_df`	dataframe of contingency table with p-values calculated by the compute_macrosynteny() function
`sp1_label`	character. The name of the species1 to display on the plot
`sp2_label`	character. The name of the species2 to put on the plot

Value

ggplot2 object

See Also

`compute_macrosynteny()`

Examples

```
# basic usage of plot_macrosynteny :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_macrosynteny <- compute_macrosynteny(my_orthologs)

plot_macrosynteny(my_macrosynteny,
```

```
                    sp1_label = "B. floridae",
                    sp2_label = "P. yessoensis")
```

`plot_oxford_grid` *plot the Macro-synteny as an oxford grid.*

Description

This is a function to plot the oxford grided plot to compare the macro synteny of two species. It requires as input an orthologs_df loaded by load_orthologs()

Usage

```
plot_oxford_grid(
  orthologs_df,
  sp1_label = "",
  sp2_label = "",
  dot_size = 0.5,
  dot_alpha = 0.4,
  reorder = FALSE,
  keep_only_significant = FALSE,
  color_by = NULL,
  pvalue_threshold = 0.001,
  color_palette = NULL,
  shade_non_significant = TRUE,
  reverse_species = FALSE,
  keep_sp1_raw_order = FALSE
)
```

Arguments

`orthologs_df`	dataframe. orthologs with genomic coordinates loaded by the load_orthologs()
`sp1_label`	character. name of 1st species to display on the plot
`sp2_label`	character. name of 2nd species to display on the plot
`dot_size`	numeric. (default = 0.5)
`dot_alpha`	numeric. (default = 0.4)
`reorder`	logical. (default = FALSE) tells whether to reorder the chromosomes in clusters as implemented in reorder_macrosynteny()
`keep_only_significant`	logical. (default = FALSE)
`color_by`	string/variable name. (default = NULL) column of the orthologs_df to use to color the dots.
`pvalue_threshold`	numeric. (default = 0.001)

`color_palette` vector. (default = NULL) list of colors (as string under double quote) for the clusters. The amount of colors must match the amount of clusters.

`shade_non_significant` logical. (default = TRUE) When TRUE the orthologs located on non-significant linkage groups are displayed in "grey"

`reverse_species` logical. (default = FALSE) When TRUE the x and y axis of the plot are reversed. sp1 is displayed on the y axis and sp2 is displayed on the x axis.

`keep_sp1_raw_order` logical.(default equals FALSE) tells if the reordering should be constrained on the species1 and change just the order of the species2

Value

A ggplot2 object

See Also

`load_orthologs()`

`reorder_macrosynteny()`

Examples

```
# basic usage of plot_oxford_grid :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

plot_oxford_grid(my_orthologs,
                 sp1_label = "B. floridae",
                 sp2_label = "P. echinospica")

# plot a reordered Oxford Grid and color by cluster :

plot_oxford_grid(my_orthologs,
                 sp1_label = "B. floridae",
                 sp2_label = "P. echinospica",
                 reorder = TRUE,
                 color_by = "clust")
```

---

reorder_macrosynteny *Reorder the mbh_df before plotting*

Description

This is a function to reorder an orthologs_df, that was generated with load_orthologs(). It retrieves communities using igraph::cluster_fast_greedy.

Usage

```
reorder_macrosynteny(
  orthologs_df,
  pvalue_threshold = 0.001,
  keep_only_significant = FALSE,
  keep_sp1_raw_order = FALSE
)
```

Arguments

orthologs_df	dataframe. mutual best hits with genomic coordinates loaded with load_orthologs()
pvalue_threshold	numeric. threshold for significancy. (default equals 0.001)
keep_only_significant	logical. (default equals FALSE) tells if the non significant linkage groups should be removed. It drastically speeds up the computation when using one highly fragmented genome.
keep_sp1_raw_order	logical. (default equals FALSE) tells if the reordering should be constrained on the species1 and change just the order of the species2

Value

A dataframe object

See Also

load_orthologs()

compute_macrosynteny()

Examples

```
# basic usage of reorder_macrosynteny :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_orthologs_reordered <- reorder_macrosynteny(my_orthologs)
```

reorder_multiple_macrosyntenies *Reorder the chromosomes of two or more species before plotting*

Description

This is a function to reorder an orthologs_df, same as reorder_macrosynteny, but it handles tables with more than 2 species.

Usage

```
reorder_multiple_macrosyntenies(orthologs_df)
```

Arguments

orthologs_df	dataframe. orthologs with genomic coordinates loaded with load_orthologs()

Value

A dataframe object

See Also

load_orthologs()

compute_macrosynteny()

reorder_macrosynteny()

Examples

```
# basic usage of reorder_macrosynteny :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_orthologs_reordered <- reorder_multiple_macrosyntenies(my_orthologs)
```

reverse_species_order *Reverse order of the species in an orthologs_df.*

Description

Returns an orthologs_df (data.frame) with reversed species order compared to the inputted orthologs_df. sp1 becomes sp2 and the otherway around. It intends at facilitating the integration of more than just two datasets. It outputs a data.frame shaped as following : sp1.ID,sp1.Chr,sp1.Start,sp1.End,sp1.Index,sp2.ID,s

Usage

```
reverse_species_order(orthologs_df)
```

Arguments

orthologs_df	orthologs_df dataframe. mutual best hits with genomic coordinates loaded with load_orthologs()

Value

dataframe composed of genomic coordinates and relative index of orthologs on both species

See Also

load_orthologs()

Examples

```
# basic usage of reverse_species_order :

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_orthologs_reversed <- reverse_species_order(my_orthologs)
```

subset_linkage_orthologs *Subset Orthologs contained in conserved linkage groups*

Description

This is a function to subset an orthologs_df and keep only the orthologs that are within significant linkage groups computed by the function compute_linkage_groups().

Usage

```
subset_linkage_orthologs(orthologs_df, linkages = NULL)
```

Arguments

orthologs_df	dataframe. orthologs with genomic coordinates loaded with load_orthologs()
linkages	dataframe. table listing the linkage groups as returned by the function compute_linkage_groups()

Value

A dataframe object

See Also

load_orthologs()

compute_linkage_groups()

Examples

```
# basic usage of compute_linkage_groups:

orthologs_table <- system.file("extdata","my_orthologs.tab",package="macrosyntR")

my_orthologs <- read.table(orthologs_table,header=TRUE)

my_macrosynteny <- compute_linkage_groups(my_orthologs)
```

Index