Package 'orthoRibbon'

November 25, 2026

Type Package

Title Ribbon-Diagram Visualization of Orthologous Genes Across Species

Version 1.0.0

Description OrthoRibbon is a visualization toolkit for displaying orthologous gene relationships across species using ribbon-based diagrams. It enables intuitive comparison of conserved genes and genomic correspondence between chromosomes or genomic regions. The package supports flexible input formats and produces publication-quality figures for comparative genomics and evolutionary analysis.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

biocViews Visualization, Annotation, ComparativeGenomics

Depends R (>= 4.6.0)

Imports AnnotationDbi, BiocGenerics, biomaRt, dplyr, geneClusterPattern, GenomeInfoDb, GenomicRanges, ggforce, ggplot2, ggrepel, methods, rlang, S4Vectors, Seqinfo, stats, seriation

Suggests BiocStyle, knitr, org.Hs.eg.db, rmarkdown, testthat

Enhances ChIPpeakAnno

URL https://github.com/jianhong/orthoRibbon

BugReports https://github.com/jianhong/orthoRibbon/issues

Config/roxygen2/version 8.1.1

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

git_branch RELEASE_3_24

git_last_commit 31fc715

git_last_commit_date 2026-11-23

Repository Bioconductor 3.24

Date/Publication 2026-11-24

Author Jianhong Ou [aut, cre] (ORCID: <https://orcid.org/0000-0002-8652-2488>), Kenneth Poss [aut, fnd]

Maintainer Jianhong Ou <jou@morgridge.org>

Contents

orthoRibbon-package . . . 2
buildBezierDF . . . 2
buildPlotData . . . 3
getGeneGRs . . . 5
getGeneIDs . . . 5
getHomologGRs . . . 6
getHomologIDs . . . 7
HomologPlotData-class . . . 7
human_fly . . . 9
orthoRibbonPlot . . . 10

Index **12**

`orthoRibbon-package` *Ribbon-based visualization of orthologous genes across species*

Description

OrthoRibbon is an R package designed to visualize orthologous gene relationships across species using ribbon diagrams. The package connects homologous genes between genomes through ribbons linking their genomic coordinates, enabling comparison of conserved genes and genomic organization across species. OrthoRibbon transforms ortholog tables into clear cross-species visualizations and produces customizable, publication-quality figures suitable for comparative genomics and evolutionary analysis.

Author(s)

Jianhong Ou

See Also

Useful links:

- https://github.com/jianhong/orthoRibbon
- Report bugs at https://github.com/jianhong/orthoRibbon/issues

---

`buildBezierDF` *Build the data.frame for B\'ezier curve*

Description

Build the data.frame for B\'ezier curve

Usage

```
buildBezierDF(
  homolog_df_list,
  colname1,
  colname2,
  col1 = "seq_top",
  col2 = "seq_bottom",
  cl1 = 4,
  alpha = 1
)
```

Arguments

homolog_df_list	A list with the data.frame of plot data for homologs.
colname1, colname2	The column names for top and bottom final positions.
col1, col2	The column names for color.
cl1	A numeric. The B\'ezier curve is created with two control points. And cl1 should be no more than 4 and no less than 0. If it is set to 4, all points in B\'ezier curve will be set to col1. If it is set to 3, the top point, two control points will be set to col1 and the bottom point will be set to col2. If it is set to 2, the top point, top control point will be set to col1 and the bottom control point, bottom point will be set to col2. If it is set to 1, the top point will be set to col1 and others col2. If it is set to 0, all points will be set to col2.
alpha	The alpha value for color.

Value

A data.frame with the coordinates, colors and group id of B\'ezier curve.

Examples

```
data("human_fly")
common_name <- setNames(nm=c("hsapiens", "dmelanogaster"))
plotData <- buildPlotData(common_name, homologs_df,
                          chrom_infos = chrom_infos,
                          chromosome_order_method = 'max',
                          max_links=1000)
bezier_chr <- buildBezierDF(plotData$homolog_df_list,
                            colname1='topChr_finalOffset',
                            colname2='bottomChr_finalOffset')
```

---

buildPlotData *Build the plot data*

Description

Build the plot data from homolog data.frame, gene annotations and chromosome informations.

Usage

```
buildPlotData(
  common_name,
  homolog_df,
  genes_gr,
  chrom_infos,
  sp_min_chr_size = 1e+07,
  chr_orders = NULL,
  filterByCoordSystem = TRUE,
  max_links = Inf,
  chromosome_order_method = "max"
)
```

Arguments

common_name	Species abbreviations e.g. "hsapiens", "mmusculus", "drerio"
homolog_df	The data.frame for homologs with column names "gene_id1", "gene_id2".
genes_gr	A GRanges object for genes. It must contain the information for all the homolog ids. 'gene_name', 'species' must be metadata column name of 'genes_gr'.
chrom_infos	A list of chromosome information. The Chromosome information for each species must be a data.frame with columns "name" and "length".
sp_min_chr_size	The minimal chromosome size.
chr_orders	Chromosome orders for plot.
filterByCoordSystem	Filter the chromosomes or not.
max_links	A numeric. The maximal link number to show in the plot.
chromosome_order_method	A character string with the name of the seriation method or 'spearman' or 'max'. If using 'max', it will try to find the best order by maximal number of homologous for each chromosome pairs. If using 'spearman', it will try to find the best order by the Spearman distance. Otherwise see seriate. It will be worth to try 'TSP' first.

Value

A HomologPlotData object with elements "homolog_df_list", "chrom_bars_df", "chrom_label_df" , "symbol_df_top" and, "symbol_df_bottom" for plot.

Examples

```
data("human_fly")
common_name <- setNames(nm=c("hsapiens", "dmelanogaster"))
plotData <- buildPlotData(common_name, homologs_df,
                          chrom_infos = chrom_infos,
                          chromosome_order_method = 'max',
                          max_links=1000)
```

getGeneGRs *Retrieve all the gene ranges*

Description

Retrieve all the gene ranges

Usage

```
getGeneGRs(ids, marts, homologs)
```

Arguments

ids	A named list with the gene ids for each species
marts	A named list with the Mart object for each species
homologs	A list of homologs. It must be the output from function getHomologGRsl

Value

A list with gene information GRanges.

Examples

```
if(interactive()){
 library(biomaRt)
 marts <- useEnsembl(
  biomart = "ENSEMBL_MART_ENSEMBL",
  dataset = "hsapiens_gene_ensembl",
  host='https://jun2026.archive.ensembl.org')
 common_name <- setNames('hsapiens', 'hsapiens')
 ids <- getGeneIDs(common_name, marts)
 homologs <- getHomologGRs(ids, common_name, marts)
 genes_gr <- getGeneGRs(ids, marts, homologs)
}
```

getGeneIDs *Retrieve all the Ensembl gene IDs by given species abbreviations*

Description

Retrieve all the Ensembl gene IDs by given species abbreviations

Usage

```
getGeneIDs(common_name, marts)
```

Arguments

common_name	species abbreviations e.g. "hsapiens", "mmusculus", "drerio"
marts	A named list with the Mart object for each species

Value

A list of character with Ensembl gene IDs.

Examples

```
if(interactive()){
library(ChIPpeakAnno)
library(org.Hs.eg.db)
ids <- getGeneIDs('hsapiens')
}
```

`getHomologGRs` *Retrieve the homolog pairs*

Description

Retrieve the homolog pairs

Usage

```
getHomologGRs(ids, common_name, marts)
```

Arguments

`ids`	A named list with the gene ids for each species
`common_name`	species abbreviations e.g. "hsapiens", "mmusculus", "drerio"
`marts`	A named list with the Mart object for each species

Value

A list with homolog GRanges.

Examples

```
## Not run:
common_name <- setNames(nm=c('hsapiens', 'mmusculus'))
# This will retrieve data via bioMart
# prepare mart for BioMart
marts <- geneClusterPattern::guessSpecies(
  common_name, output='mart',
  host='https://jun2026.archive.ensembl.org')

# prepare all the Ensembl ids
ids <- orthoRibbon::getGeneIDs(common_name, marts)

# retrieve all the homologs
homologs <- orthoRibbon::getHomologGRs(ids, common_name, marts)

## End(Not run)
```

getHomologIDs *Retrieve the homolog pairs*

Description

Retrieve the homolog pairs

Usage

```
getHomologIDs(homologs)
```

Arguments

homologs	A named list with the homolog GRanges. The output from function getHomologGRs.

Value

A data.frame with homologs.

Examples

```
library(GenomicRanges)
homologs <- list('a2b'=GRangesList(
 x=GRanges('seq1', IRanges(seq.int(5), width=1, name=letters[seq.int(5)]),
  homolog_ensembl_gene_ids=LETTERS[seq.int(5)]),
 y=GRanges('seq2', IRanges(seq.int(3), width=1, names=letters[seq.int(3)]),
  homolog_ensembl_gene_ids=c('m', 'n', 'k'))
))
getHomologIDs(homologs)
```

HomologPlotData-class *HomologPlotData S4 class*

Description

Wraps the collection of data frames/lists used for an orthoRibbon plot into a single S4 object, with list-like `$`, `[[`, and `[` access.

Usage

```
HomologPlotData(
  homolog_df_list,
  chrom_bars_df,
  chrom_label_df,
  symbol_df_top,
  symbol_df_bottom,
  common_name
)
```

```
## S4 method for signature 'HomologPlotData'
x$name

## S4 replacement method for signature 'HomologPlotData'
x$name <- value

## S4 method for signature 'HomologPlotData,ANY,ANY'
x[[i, j, ...]]

## S4 method for signature 'HomologPlotData,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'HomologPlotData'
names(x)

## S3 method for class 'HomologPlotData'
.DollarNames(x, pattern = "")

## S4 method for signature 'HomologPlotData'
subset(x, subset, ...)
```

Arguments

homolog_df_list	list of data.frame with top and bottom plot information.
chrom_bars_df	data.frame of chromosome plot information.
chrom_label_df	data.frame of chromosome labels information.
symbol_df_top, symbol_df_bottom	data.frame of symbols for top and bottom layer.
common_name	species abbreviations e.g. "hsapiens", "mmusculus", "drerio"
x	A HomologPlotData object.
name, i	slot name of HomologPlotData.
value	value to be set
j, drop, ...	not used.
pattern	A regular expression. Only matching names are returned.
subset	is a named list, e.g. list(speciesA = c("chr1", "chr2"), speciesB = c("chr1", "chrX")) Names must match values in chrom_bars_df$sp; each element's values must match either chrom_bars_df$chrom or chrom_bars_df$label for that species. The first name is treated as the "top" species, the last as the "bottom" species, and homolog_df_list / symbol_df_top / symbol_df_bottom are filtered to match.

Value

An object of HomologPlotData.

Slots

homolog_df_list	list of data.frame with top and bottom plot information.
chrom_bars_df	data.frame of chromosome plot information.

`chrom_label_df` data.frame of chromosome labels information.

`symbol_df_top`, `symbol_df_bottom` data.frame of symbols for top and bottom layer.

`common_name` species abbreviations e.g. "hsapiens", "mmusculus", "drerio"

Examples

```
HomologPlotData(
  homolog_df_list=list(),
  chrom_bars_df=data.frame(),
  chrom_label_df=data.frame(),
  symbol_df_top=data.frame(),
  symbol_df_bottom=data.frame(),
  common_name=setNames(nm=c('hsapiens', 'mmusculus'))
)
```

---

human_fly *Human and Drosophila Ortholog Groups*

Description

Ortholog groups for human (*Homo sapiens*) and fruit fly (*Drosophila melanogaster*) identified using OrthoFinder.

Preprocessed homolog information for zebrafish, human, and mouse retrieved from Ensembl. The original data were filtered and reduced to minimize the dataset size.

Format

The dataset contains three objects:

chrom_infos A list of data frames for `hsapiens` and `dmelanogaster`. Each data frame contains the columns `name`, `length`, `coord_system`, `synonyms`, `toplevel`, `non_ref`, and `circular`.

homologs_df A data frame containing the columns ' `gene_id`, `ortholog_group`, `symbol`, `species`, `seq`, and `start`.

genes_gr A `GRanges` object containing 40,528 ranges with two metadata columns: `gene_name` and `species`.

The dataset contains three objects:

chrom_infos A list of data frames for `hsapiens`, `mmusculus`, and `drerio`. Each data frame contains the columns `name`, `length`, `coord_system`, `synonyms`, `toplevel`, `non_ref`, and `circular`.

homologs_df A data frame containing two columns: `gene_id1` and `gene_id2`.

genes_gr A `GRanges` object containing 44,392 ranges with two metadata columns: `gene_name` and `species`.

Source

Data retrieved from Ensembl using BioMart on March 13, 2026.

`orthoRibbonPlot` *Plot the data*

Description

Plot data by ribbon plot.

Usage

```
orthoRibbonPlot(
  common_name,
  bezier_df,
  chrom_bars_df,
  chrom_label_df,
  symbol_df_top,
  symbol_df_bottom,
  link_lwd = 0.25,
  link_alpha = 1,
  chr_lwd = 3,
  chr_lineend = "round",
  chr_size = 6,
  label_size = 12,
  symbol_size = 2,
  xlim = c(-0.02, 1.02),
  show_symbol = FALSE
)
```

Arguments

`common_name`	Species abbreviations e.g. "hsapiens", "mmusculus", "drerio".
`bezier_df`	A data.frame. The coordinates for Bezier curves. It must have columns "x", "y", "id", and "col". The "id" column saves group id for each pair of homologs.
`chrom_bars_df`	A data.frame. The coordinates for chromosome line. It must have columns "x", "y", "sp" and "chrom". Here "sp" means species and "chrom" indicates the chromosome.
`chrom_label_df`	A data.frame. The coordinates for chromosome names. It must have columns "x", "y" and "labels"
`symbol_df_top`, `symbol_df_bottom`	The data.frame for the gene labels
`link_lwd`, `chr_lwd`	The line width for Bezier curve, chromosome bar.
`link_alpha`	The alpha value for Bezier curve.
`chr_lineend`	The line end for chromosome bar.
`chr_size`, `label_size`, `symbol_size`	The size for chromosome label, species, and symbols.
`xlim`	Two numeric values, specifying the x limit of the scale.
`show_symbol`	A logical value. Show the symbols or not.

Value

A ggplot object

Examples

```
common_name <- setNames(nm=c("hsapiens", "dmelanogaster"))
data("human_fly")
plotData <- buildPlotData(common_name, homologs_df,
                          chrom_infos = chrom_infos,
                          chromosome_order_method = 'max',
                          max_links=1000)
bezier_chr <- buildBezierDF(plotData$homolog_df_list,
                            colname1='topChr_finalOffset',
                            colname2='bottomChr_finalOffset')
orthoRibbonPlot(common_name, bezier_chr,
                plotData$chrom_bars_df,
                plotData$chrom_label_df)
```

Index

∗ **datasets**
human_fly, 9
.DollarNames,HomologPlotData,character-method
(HomologPlotData-class), 7
.DollarNames.HomologPlotData
(HomologPlotData-class), 7
[,HomologPlotData,ANY,ANY,ANY-method
(HomologPlotData-class), 7
[[,HomologPlotData,ANY,ANY-method
(HomologPlotData-class), 7
$,HomologPlotData-method
(HomologPlotData-class), 7
$<-,HomologPlotData-method
(HomologPlotData-class), 7

buildBezierDF, 2
buildPlotData, 3

chrom_infos *(*human_fly*)*, 9

ensembl_hs_mm_dr *(*human_fly*)*, 9

genes_gr *(*human_fly*)*, 9
getGeneGRs, 5
getGeneIDs, 5
getHomologGRs, *5*, 6, *7*
getHomologIDs, 7

HomologPlotData, *4*
HomologPlotData
(HomologPlotData-class), 7
HomologPlotData-class, 7
homologs_df *(*human_fly*)*, 9
human_fly, 9

names,HomologPlotData-method
(HomologPlotData-class), 7

orthoRibbon *(*orthoRibbon-package*)*, 2
orthoRibbon-package, 2
orthoRibbonPlot, 10

seriate, *4*
subset,HomologPlotData,list-method
(HomologPlotData-class), 7
subset,HomologPlotData-method
(HomologPlotData-class), 7