

Package ‘sytenyPlotteR’

May 9, 2026

Type Package

Title Genome Syteny Visualization

Version 1.0.0

Maintainer Marta Farre <m.farre-belmonte@kent.ac.uk>

Description Draw sytenic relationships between genome assemblies.

There are 3 functions which take a tab delimited file containing alignment data for sytenic blocks between genomes to produce either a linear alignment plot, an evolution highway style plot, or a painted ideogram representing sytenic relationships. There is also a function to convert alignment data in the DESCHRAMBLER/inferCAR format to the required data structure.

Encoding UTF-8

RoxygenNote 7.2.3.9000

Imports ggplot2, stringr, utils

NeedsCompilation no

BugReports <https://github.com/Farre-lab/sytenyPlotteR/issues>

Suggests knitr, rmarkdown

VignetteBuilder knitr

License MIT + file LICENSE

Author Marta Farre [aut, cre, cph],

Sarah Quigley [aut],

Joana Damas [aut],

Denis Larkin [aut]

Repository CRAN

Date/Publication 2023-10-03 16:50:02 UTC

Contents

draw.eh	2
draw.ideogram	4
draw.linear	6
reformat.sytenyData	8

`draw.eh`*Evolution Highway style plot*

Description

Draw Evolution Highway Plots

Usage

```
draw.eh(  
  output,  
  chrRange,  
  data_file,  
  directory = NULL,  
  fileformat = "png",  
  colour = "lightblue",  
  inverted.colour = "lightpink",  
  w = 5.5,  
  h = 10,  
  ps = 10  
)
```

Arguments

<code>output</code>	output file name
<code>chrRange</code>	range of chromosome numbers in the reference as numbers i.e. 1:29
<code>data_file</code>	file containing the syntentic blocks from the alignments
<code>directory</code>	string containing file path to chosen directory to save image file
<code>fileformat</code>	output file format desired using the format <code>fileformat = "png"</code> (default is "png")
<code>colour</code>	set colour for non-inverted syntenic blocks using the format <code>colour = "red"</code> (default is "lightblue")
<code>inverted.colour</code>	set colour for inverted syntenic blocks using the format <code>inverted.colour = "blue"</code> (default is "lightpink")
<code>w</code>	width of output image using the format <code>w = 5.5</code> (default)
<code>h</code>	height of output image using the format <code>h = 10</code> (default)
<code>ps</code>	point size of output image using the format <code>ps = 10</code> (default)

Details

This function draws Evolution Highway style plots.

It requires as input:

1. Desired output file name
2. The range of chromosomes of the reference species this is entered as either a single number i.e. 1 or a range of numbers i.e. 1:22. *Note: if you are inputting sex chromosomes or chromosomes with characters in the name input a concatenated string i.e. c(1,2,3,"X")*
3. File containing the syntenic blocks of alignments of one or more target species aligned to a single reference; following this format: reference chromosome, reference start position, reference end position, target chromosome, target start position, target end position, orient, reference species identifier, target species identifier

There are optional parameters for some customization of this function:

1. The format for saving the image i.e. png or pdf can be altered by inputting: `fileformat = "pdf"` (the default value is "png")
2. The colour of the syntenic blocks (not inverted blocks) can be changed by inputting: `colour = "red"` (the default value is "lightblue", see Rcolour palette for colour options)
3. The colour of the inverted syntenic blocks can be changed by inputting: `inverted.colour = "blue"` (the default value is "lightpink", see Rcolour palette for colour options)
4. The width of the image created can be changed by inputting: `w = 5.5`
5. The height of the image created can be changed by inputting: `h = 10`
6. The point size of the image created can be changed by inputting: `ps = 10`
7. The directory where the image file should be saved, as default the image is saved to temporary directory, change by inputting: `directory = "path/to/directory"`

The function works creating a graph for each reference chromosome using their start and end positions to create a block for the reference and the target chromosome positions are used to colour the region where synteny was identified

Example: `draw.eh("outputName",c(17,"X"), "example_eh_alignments_2.txt", directory = "path/to/directory", fileformat = "pdf")`

Value

An image file showing the comparative drawings

Examples

```
# Create object containing file path to external dataset
# (see vignette to follow examples with personal data)

file1 <- system.file("extdata", "example_eh_alignments_2.txt", package = "syntenyPlotteR")

# -----

# Run draw.eh function
```

```
# To run example and save file to working directory
# add directory parameter and set working directory
# To run example with personal data see vignette

draw.eh("outputName", c(17, "X"), file1, fileformat = "pdf")
```

draw.ideogram

Draw ideograms in chromosome painting style

Description

Draw syntenic ideograms in Chromosome painting style

Usage

```
draw.ideogram(
  file_data,
  sizefile,
  output,
  directory = NULL,
  fileformat = "png",
  colours = colours.default,
  w = 8.5,
  h = 10,
  ps = 5
)
```

Arguments

file_data	Path to the syntenic blocks file
sizefile	Chromosome size file
output	output file name
directory	string containing file path to chosen directory to save image file
fileformat	output file format specified using fileformat = "pdf" (the default is "png")
colours	concatenated string of chromosome IDs and assigned colours if desired using the format colours = c("1" = "red", "2" = "blue", "3" = "green", "X" = "grey") if the no colours are assigned default values will be used
w	width of output image using w = 8.5 (default)
h	height of output image using h = 10 (default)
ps	point size of output image using ps = 5 (default)

Details

This function draws pairwise synteny plots in chromosome painting style.

Inputs are tab separated files;

It requires as input:

1. File containing the syntenic blocks following this format: reference chromosome, reference start position, reference end position, target chromosome, target start position, target end position, orient, reference species identifier, target species identifier
2. Tab separated file of all chromosome, scaffold, or contig lengths and the species identifier, in order from first target species in the alignment files followed by the first reference species in the alignment files – top of file – to the last target species and reference species in the alignment files – end of file. in this format: chromosome ID, chromosome length, species identifier
3. The desired output file name

Please separate files by tab and ensure any species identifiers used between length and alignment files are matching (same identifiers and casing)

There are optional parameters for some customization of this function:

1. The format for saving the image i.e. png or pdf can be altered by inputting: `fileformat = "pdf"` (the default value is "png")
2. The colour of the ideograms can be altered by inputting a concatenated string of chromosome IDs with assigned colour values which can be found with R colour Palette e.g. `colours = c("1" = "red", "2" = "blue", "3" = "green", "4" = "orange", "5" = "purple", "X" = "grey")` if no colours are assigned default values will be used but colours MUST be assigned to all chromosomes
3. The width of the image created can be changed by inputting: `w = 8.5` (default)
4. The height of the image created can be changed by inputting: `h = 10` (default)
5. The point size of the image created can be changed by inputting: `ps = 5` (default)
6. The directory where the image file should be saved, as default the image is saved to temporary directory, change by inputting: `directory = "path/to/directory"`

Target is the species which chromosomes will be painted. Reference will be used for painting and diagonals. Chromosomes will be in the same order as in the target chromosomes in the chromosome length file

Example: `draw.ideogram("example_alignment_1.txt", "example_lengths.txt", "outputname", directory = "path/to/directory", fileformat = "pdf")`

Value

An image file showing the ideogram

Examples

```
# Create objects containing file paths to external dataset
# (see vignette to follow examples with personal data)

length.file <- system.file("extdata", "example_lengths.txt", package = "syntenPlotter")
file1 <- system.file("extdata", "example_alignment_1.txt", package = "syntenPlotter")

# -----

# Run draw.ideogram function
# To run example and save file to working directory
# add directory parameter and set working directory
# To run example with personal data see vignette

draw.ideogram(file1, length.file, "outputName", fileformat = "pdf")
```

draw.linear

Linear syteny plot

Description

Draw Linear Syteny Plots

Usage

```
draw.linear(
  output,
  sizefile,
  ...,
  directory = NULL,
  fileformat = "png",
  colours = colours.default,
  w = 13,
  h = 5,
  opacity = 0.5
)
```

Arguments

output	output file name
sizefile	Chromosome Size file
...	syteny files (any number of alignment files can be entered)
directory	string containing file path to chosen directory to save image file
fileformat	output file format specified using the format fileformat = "pdf" (the default is "png")

colours	concatenated string of chromosome IDs and assigned colours if desired using the format colours = c("1" = "red", "2" = "blue", "3" = "green", "X" = "grey") if the no colours are assigned default values will be used
w	width of output image using the format w = 13 (default)
h	height of output image using the format h = 5 (default)
opacity	opacity of syntenic bands using the format opacity = .5 (default)

Details

This function draws linear syteny plots.

It requires:

1. The desired output file name;
2. Tab separated file of all chromosome, scaffold, or contig lengths and the species identifier, in order from first target species in the alignment files followed by the first reference species in the alignment files – top of file – to the last target species and reference species in the alignment files – end of file. in this format: chromosome ID, chromosome length, species identifier
3. files containing the syntenic blocks - one file per alignment, in order from first target/reference (most recent species pairwise alignment in ancestral reconstruction data) alignment file to last target/reference (ancestor pairwise alignment in ancestral reconstruction data) alignment file following this format: reference chromosome, reference start position, reference end position, target chromosome, target start position, target end position, orient, reference species identifier, target species identifier

Please separate files by tab and ensure any species identifiers used between length and alignment files are matching (same identifiers and casing)

There are optional parameters for some customization of this function:

1. The format for saving the image i.e. png or pdf can be altered by inputting: fileformat = "pdf" (the default value is "png")
2. The colour of the syteny bands can be altered by inputting a concatenated string of chromosome IDs with assigned colour values which can be found with R colour Palette e.g. colours = c("1" = "red", "2" = "blue", "3" = "green", "4" = "orange", "5" = "purple", "X" = "grey") if no colours are assigned default values will be used but colours **MUST** be assigned to all chromosomes
3. The width of the image created can be changed by inputting: w = 13 (default)
4. The height of the image created can be changed by inputting: h = 5 (default)
5. The opacity of the ribbons can be changed by inputting: opacity = .5 (default)
6. The directory where the image file should be saved, as default the image is saved to temporary directory, change by inputting: directory = "path/to/directory"

The function works using the chromosome length file to order the Y axis and provide chromosome lengths to draw chromosome ideograms and the alignment files provides coordinates to draw the alignment bands between ideograms

Example: draw.linear("outputname", "example_lengths.txt", "example_alignment_1.txt", "example_alignment_2.txt", "example_alignment_3.txt", directory = "path/to/directory", fileformat = "pdf")

Value

An image file showing the linear comparison drawings

Examples

```
# Create objects containing file paths to external dataset
# (see vignette to follow examples with personal data)

length.file <- system.file("extdata", "example_lengths.txt", package = "syntenyPlotterR")
file1 <- system.file("extdata", "example_alignment_1.txt", package = "syntenyPlotterR")
file2 <- system.file("extdata", "example_alignment_2.txt", package = "syntenyPlotterR")
file3 <- system.file("extdata", "example_alignment_3.txt", package = "syntenyPlotterR")

# -----

# Run draw.linear function
# To run example and save file to working directory
# add directory parameter and set working directory
# To run example with personal data see vignette

draw.linear("outputName", length.file, file1, file2, file3, fileformat = "pdf")
```

reformat.syntenyData *Reformat synteny data*

Description

Reformat synteny data

Usage

```
reformat.syntenyData(
  file_data,
  filename,
  directory = NULL,
  reference.species = reference.sps,
  target.species = target.sps
)
```

Arguments

file_data input file name for descrambler .map data
filename output file name for reformatted data
directory string containing file path to chosen directory to save text file
reference.species reference species identifier as a character string
target.species target species identifier as a character string

Details

This function takes output from alignment softwares such as deschrabler and inferCARs and re-formats it for syntenyPlotteR - this does not curate files only re-formats it

It requires as input:

1. The data output from deschrabler or inferCARs
2. the desired output file name

There are optional parameters for some customization of this function:

1. reference.species allows you to set the reference species identifier that will be set in the final output table i.e. reference.species = "ref"
2. target.species allows you to set the target species identifier that will be set in the final output table i.e. target.species = "tar"
3. The directory where the text file should be saved, as default the file is saved to temporary directory, change by inputting: directory = "path/to/directory"

Example: reformat.syntenyData("deschrabler.output", "reformatted.data", directory = "path/to/directory", reference.species = "ref", target.species = "tar")

Value

A text file with the reformatted data

Examples

```
# Create object containing file path to external dataset
# (see vignette to follow examples with personal data)

file <- system.file("extdata", "example_map_1.map", package = "syntenyPlotteR")

# -----

# Run reformat.syntenyData function
# To run example and save file to working directory
# add directory parameter and set working directory
# To run example with personal data see vignette

reformat.syntenyData(file, "outputName")
```

Index

`draw.eh`, [2](#)

`draw.ideogram`, [4](#)

`draw.linear`, [6](#)

`reformat.syntenyData`, [8](#)