Package 'RCandy'

August 24, 2021

Title RCandy: an R package for rapid visualisation of homologous recombination events in bacterial genomes

Version 1.0.0

Description Plots a phylogenetic tree in context of taxon metadata and recombination events identified by Gubbins (Croucher et al. 2015. <https://pubmed.ncbi.nlm.nih.gov/25414349/>) and BRATNextGen (Marttinen et al. 2012. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3245952/>).

License MIT + file LICENSE

URL https://github.com/ChrispinChaguza/RCandy/

BugReports https://github.com/ChrispinChaguza/RCandy/issues

Depends R (>= 3.6)

Imports ape,
dplyr,
graphics,
grDevices,
magrittr,
phytools,
shape,
stats,
stringr,
tibble,
tidyr,
utils,
viridis

Suggests knitr,
rmarkdown,
markdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

R topics documented:

`"RCandy"` *Gubbins recombination data*

Description

This dataset provides an example phylogenetic tree, strain metadata, reference genome and recombination events for 170 Streptococcus pneumoniae isolates collected globally belonging to the multilocus sequence typing (MLST) clone ST320.

Usage

```
data("RCandy")
```

Format

A list containing four data frames:

`tree` phylo: Phylogenetic tree in ape's phylo format

`metadata` data frame: Data frame containing metadata for the strains in the tree

`gubbins.GFF` data frame: Data frame containing recombination events identified in each strain (GFF format)

`refgenome.GFF` data frame: Data frame containing genomic features in the reference genome (GFF format)

A list containing pre-loaded phylogenetic tree, taxon metadata, Gubbins GFF file, and reference genome GFF file

Source

Gladstone RA, Lo SW et al. International genomic definition of pneumococcal lineages, to contextualise disease, antibiotic resistance and vaccine impact. EBioMedicine. 2019 May;43:338-346. doi: 10.1016/j.ebiom.2019.04.021. Epub 2019 Apr 16. PMID: 31003929; PMCID: PMC6557916.

The Global Pneumococcal Sequencing (GPS) Consortium: https://www.pneumogen.net/gps/

The reference whole genome sequence was obtained from GenBank (accession number: NC_010380): https://www.ncbi.nlm.nih.gov/nuccore/NC_010380.1.

`count.rec.events.per.base` *Count number of overlapping recombination events at each genomic position*

Description

This function reads a GFF file or data frame containing identified recombination events in the genome identified by Gubbins, and counts the frequency of recombination events at each genomic position i.e. number of unique overlapping recombination events. The data frame can be generated using the "load.gubbins.rec.events.gff" function

Usage

```
count.rec.events.per.base(gubbins.gff.file, recom.input.type = "Gubbins")
```

Arguments

`gubbins.gff.file`
: Path to the input Gubbins GFF recombination file or data frame

`recom.input.type`
: Type of input recombination data, either "Gubbins" GFF or "BRATNextGen" tabular data.

Value

A data frame containing number of unique recombination events at genomic positions where recombination events were identified

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Read genome in GFF formatted file (generated usign readseq) and plot
the genomic features

This function may take some minutes to finish depending on the number
of recombination events identifiedand genome size

gubbins.gff<-system.file("extdata", "ST320.recombination_predictions.gff",
package = "RCandy",mustWork = TRUE)

rec.freq<-count.rec.events.per.base(gubbins.gff)

## End(Not run)
```

`count.rec.events.per.genome` *Count number of overlapping recombination events at each genomic position*

Description

This function reads a GFF file or data frame containing identified recombination events in the genome identified by Gubbins, and counts the frequency of recombination events at each genomic position i.e. number of unique overlapping recombination events. The data frame can be generated using the "load.gubbins.rec.events.gff" function

Usage

```
count.rec.events.per.genome(
  gubbins.gff.file,
  recom.input.type = "Gubbins",
  taxon.names
)
```

Arguments

`gubbins.gff.file`	Path to the input Gubbins GFF recombination file or data frame
`recom.input.type`	Type of input recombination data, either "Gubbins" GFF or "BRATNextGen" tabular data.
`taxon.names`	Vector containing taxon names.

Value

A data frame containing number of unique recombination events at genomic positions where recombination events were identified

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Read genome in GFF formatted file (generated usign readseq) and plot
the genomic features

This function may take some minutes to finish depending on the number
of recombination events identifiedand genome size

gubbins.gff<-system.file("extdata", "ST320.recombination_predictions.gff",
```

```
package = "RCandy",mustWork = TRUE)

rec.freq<-count.rec.events.per.base(gubbins.gff)

## End(Not run)
```

`is.color` *Check if string is a valid colour name*

Description

This function checks if a string or a vector contains valid colour names

Usage

```
is.color(colour.name)
```

Arguments

`colour.name`	Input character or vector containing colour names

Value

A Boolean showing whether or not a string or vector contains valid colour names

Examples

```
## Not run:
Read the colour names as a string or vector of characters

col.names<-is.color(c("red","blue"))
col.names<-is.color("blue")

## End(Not run)
```

`load.genome.GFF` *This function loads a reference genome file in GFF format.*

Description

This function loads a reference genome file in GFF format.

Usage

```
load.genome.GFF(reference.genome)
```

Arguments

`reference.genome`	Input file name or data frame in GFF file format.

Value

A data frame.

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Read genome in GFF formatted file, generated usign readseq, and plot the genomic features

ref.genome.gff<-system.file("extdata", "Hungary19A-6.gff", package = "RCandy",mustWork = TRUE)

new.ref.genome<-load.genome.GFF(ref.genome.gff)

## End(Not run)
```

---

load.gubbins.GFF *Read recombination events from the output GFF file from Gubbins*

Description

This function reads a GFF format generated by Gubbins containing genomic regions where recombination events were identified.

Usage

```
load.gubbins.GFF(gubbins.gff.file, recom.input.type = "Gubbins")
```

Arguments

gubbins.gff.file	Path to the input Gubbins GFF recombination file or data frame
recom.input.type	Type of input recombination data, either "Gubbins" GFF or "BRATNextGen" tabular data.

Value

A data frame containing number of unique recombination events at genomic positions where recombination events were identified

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Read genome in GFF formatted file (generated usign readseq) and plot the genomic features

gubbins.gff<-system.file("extdata", "ST320.recombination_predictions.gff", package = "RCandy",
mustWork = TRUE)

rec.events<-load.gubbins.rec.events.gff(gubbins.gff)

## End(Not run)
```

---

load.taxon.metadata *Function to read and process taxon metadata file*

Description

This function reads and process a user specified metadata file or data frame. It assumes that the file is in text format and that the columns are tab-delimited. Metadata provided in any other format other than as a 'character' class file name or a data frame with class "tbl_df", "tbl","grouped_df","data.frame" or "rowwise_df" will not be accepted. When not specified, it will assume that the first column represents the taxon names. The taxon names should match those included in the phylogenetic tree file or object

Usage

```
load.taxon.metadata(
  taxon.metadata.file,
  taxon.metadata.columns = NULL,
  taxon.names = NULL,
  taxon.id.column = NULL,
  include.first.col = FALSE,
  taxon.metadata.delimeter = "\t"
)
```

Arguments

`taxon.metadata.file`
: Path to the input metadata file name or data frame.

`taxon.metadata.columns`
: A vector containing name of columns in the matadata file or data frame to view in the phylogenetic tree.

`taxon.names`
: A vector containing taxon names to select from the metadata file or data frames. These names must match the taxon names in the phylogenetic tree.

`taxon.id.column`
: Column name in the matadata file or data frame containing taxon names.

include.first.col
: A Boolean value specifying whether to use the first column as the taxon names.

taxon.metadata.delimeter
: A delimiter separating metadata columns.

Value

A data frame containing selected metadata columns and taxon names in the phylogenetic tree.

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Read a tab-delimited file containing metadata

metadata.file<-system.file("extdata", "ST320.tsv", package = "RCandy",mustWork = TRUE)

metadata.df<-load.taxon.metadata(metadata.file)

## End(Not run)
```

RCandyVis *Draw and annotate phylogenetic tree with taxon metadata, and recombination events.*

Description

This function reads a reference genome in GFF format and then plots the genetic features (coding sequences) on both forward and reverse strands.

Usage

```
RCandyVis(
  tree.file.name,
  taxon.metadata.file = NULL,
  taxon.metadata.columns = NULL,
  gubbins.gff.file = NULL,
  recom.input.type = "Gubbins",
  ref.genome.name = NULL,
  metadata.column.label.angle = 90,
  show.gene.label = FALSE,
  gene.label.angle = 45,
  show.metadata.columns = TRUE,
  subtree.tips = NULL,
  color.pallette = "inferno",
  taxon.id.column = NULL,
```

```
  show.genome.ticks = TRUE,
  show.genome.axis = TRUE,
  rec.heatmap.color = c("red", "blue"),
  tree.scale.length = NULL,
  show.rec.events = TRUE,
  show.metadata.label = TRUE,
  taxon.metadata.label.cex = 0.85,
  taxon.metadata.delimeter = "\t",
  ref.genome.length = NULL,
  show.rec.freq.per.base = FALSE,
  show.rec.freq.per.genome = FALSE,
  rec.events.per.base.as.barplot = FALSE,
  ladderize.tree.right = NULL,
  midpoint.root = FALSE,
  show.rec.plot.bg = TRUE,
  show.genome.annot = TRUE,
  show.rec.plot.tracks = FALSE,
  show.rec.plot.border = FALSE,
  ace.model.name = "ARD",
  trait.for.ancestral.reconstr = NULL,
  save.to.this.file = NULL,
  plot.width = 12,
  plot.height = 9.5,
  show.tip.label = FALSE,
  align.tip.label = FALSE,
  show.fig.legend = TRUE,
  genome.start = NULL,
  genome.end = NULL,
  color.tree.tips.by.column = NULL,
  tree.tip.node.cex = 0.35,
  tree.tip.label.cex = 0.35,
  tree.node.cex = 0.6
)
```

Arguments

`tree.file.name`	File name or ape phylo object containing the phylogenetic tree in Newick format.
`taxon.metadata.file`	File name or data frame containing taxon metadata.
`taxon.metadata.columns`	Vector containing metadata columns to plotted.
`gubbins.gff.file`	Gubbins output recombination file in GFF format.
`recom.input.type`	Type of input recombination data, either "Gubbins" GFF or "BRATNextGen" tabular data.
`ref.genome.name`	Reference genome file name in GFF format.
`metadata.column.label.angle`	Angle for the metadata column labels.

show.gene.label
: A Boolean value indicating whether or not to show gene labels in the reference genome.

gene.label.angle
: Angle for the gene labels.

show.metadata.columns
: A Boolean indicating whether or not to show metadata in the figure.

subtree.tips
: A vector containing a subset of taxons/taxa used to generate a subtree from the main phylogenetic tree.

color.pallette
: A vector containing names of the viridis colour palletes for visualisation. Choose from "plasma", "cividis", "viridis", "magma" and "inferno".

taxon.id.column
: Character or string for the column name containing the strain/taxon name in the metadata file.

show.genome.ticks
: A Boolean indicating whether to show the xticks for the recombination events diagram/heatmap.

show.genome.axis
: A Boolean indicating whether to show the axis for the recombination events diagram/heatmap.

rec.heatmap.color
: A two-value vector containing colour names to use for the recombination event diagram/heatmap.

tree.scale.length
: A positive number showing the length of the phylogenetic tree branches.

show.rec.events
: A Boolean indicating whether to show the recombination event diagram/diagram.

show.metadata.label
: A Boolean indicating whether to show labels for the selected metadata columns.

taxon.metadata.label.cex
: A number for the size of the labels for the selected matadata columns

taxon.metadata.delimeter
: A delimeter separating metadata columns.

ref.genome.length
: An optional reference genome length, otherwise it's read from the reference genome GFF file or data frame.

show.rec.freq.per.base
: A Boolean indicating whether to show the frequency of recombination per genomic position/base.

show.rec.freq.per.genome
: A Boolean indicating whether to show the frequency of recombination events per genome/taxon.

rec.events.per.base.as.barplot
: A Boolean indicating whether to show the frequency of recombination events per genome/taxon as a barchart or colour scale (heatmap).

ladderize.tree.right
: A Boolean indicating whether to ladderize the phylogenetic tree to the right.

midpoint.root
: A Boolean indicating whether to root the phylogenetic tree at midpoint.

show.rec.plot.bg
: A Boolean indicating whether to show background for the recombination events diagram/heatmap.

show.genome.annot
: A Boolean indicating whether to show genome annotation above the recombination events diagram/heatmap.

show.rec.plot.tracks
: A Boolean indicating whether to plot genome tracks for each taxa.

show.rec.plot.border
: A Boolean indicating whether to show the border for the recombination events diagram/heatmap.

ace.model.name
: A character or string for the model name used for the discrete ancestral character reconstruction. Choose from "ARD", "ER" and "SYM".

trait.for.ancestral.reconstr
: A character or string for the column in the metadata file or data frame used for discrete ancestral character reconstruction.

save.to.this.file
: If speficified save the plot to this filename, otherwise show the plot in R.

plot.width
: Width of the figure

plot.height
: Height of the figure

show.tip.label
: A Boolean indicating whether to show the phylogenetic tip labels.

align.tip.label
: A Boolean indicating whether to align the phylogenetic tip labels.

show.fig.legend
: A Boolean indicating whether to show the legend for the selected metadata columns.

genome.start
: A positive number indicating start position in the genome to zoom in.

genome.end
: A positive number indicating end position in the genome to zoom in.

color.tree.tips.by.column
: Character or string for the column name in the metadata file for colouring the phylogenetic tree tips or terminal nodes.

tree.tip.node.cex
: A number for the terminal node or tip size in the phylogenetic tree.

tree.tip.label.cex
: A number for the tip label size in the phylogenetic tree.

tree.node.cex
: A number for the size of the nodes phylogenetic tree.

Value

None

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Read phylogenetic tree in Newick format, reference genome in GFF formatted file,
generated usign readseq, and Gubbins GFF file to plot the genomic features

metadata.file<-system.file("extdata", "ST320.tsv", package = "RCandy",
mustWork = TRUE)
tree.file<-system.file("extdata", "ST320.final_tree.tre", package = "RCandy",
mustWork = TRUE)
gubbins.gff<-system.file("extdata", "ST320.recombination_predictions.gff",
package = "RCandy",mustWork = TRUE)
ref.genome.gff<-system.file("extdata", "Hungary19A-6.gff", package = "RCandy",
mustWork = TRUE)

RCandyVis(tree.file.name = tree.file, taxon.metadata.file = metadata.file,
taxon.metadata.columns = c("Source","Country"),ref.genome.name = ref.genome.gff,
gubbins.gff.file = gubbins.gff,color.tree.tips.by.column = "Country",
show.rec.freq.per.base = FALSE,show.gene.label = FALSE,ladderize.tree.right = TRUE,
midpoint.root = TRUE)

RCandyVis(tree.file.name = tree.file, taxon.metadata.file = metadata.file,
taxon.metadata.columns = c("Source","Country"),ref.genome.name = ref.genome.gff,
gubbins.gff.file = gubbins.gff,color.tree.tips.by.column = "Country",
show.rec.freq.per.base = FALSE,show.gene.label = FALSE,ladderize.tree.right = TRUE,
midpoint.root = TRUE,genome.start = 30000, genome.end = 60000,show.gene.label=TRUE,
save.to.this.file = "RCandy.output.pdf",)

## End(Not run)
```

---

read.tree.file *Function to read phylogenetic tree in Newick file*

Description

This function reads and process a user specified phylogenetic tree file in Newick format.

Usage

```
read.tree.file(tree.file.name)
```

Arguments

tree.file.name	Path to the input phylogenetic tree file in Newick format.

Value

A data frame containing selected metadata columns and strain names in the phylogenetic tree.

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Load phylogenetic tree file.

tree.file<-system.file("extdata", "ST320.final_tree.tre", package = "RCandy",mustWork = TRUE)

read.tree.file(tree.file.name=tree.file)

## End(Not run)
```

---

show.blank.plot *Generate a blank plot*

Description

This function generates a blank plot. The plot can be used as a filler in when creating multipanel diagrams.

Usage

```
show.blank.plot()
```

---

show.genome.annotation.plot *Draw genome annotation features*

Description

This function reads a reference genome in GFF format and then plots the genetic features (coding sequences) on both forward and reverse strands.

Usage

```
show.genome.annotation.plot(
  genome.name,
  genome.start = NULL,
  genome.end = NULL,
  genome.start.upstream = 0,
  genome.end.downstream = 0,
  show.gene.label = FALSE,
  gene.feature.width = 1.5
)
```

Arguments

genome.name	Path to the input file name.
genome.start	Start position of the genome to show in the plot.
genome.end	End position of the genome to show in the plot.
genome.start.upstream	Start drawing the genome plot from the specified bases upstream of the genome.
genome.end.downstream	End drawing the genome plot from the specified bases upstream of the genome.
show.gene.label	Show genetic feature label.
gene.feature.width	Width of the genetic features.

Value

None

Author(s)

Chrispin Chaguza, <Chrispin.Chaguza@gmail.com>

References

https://github.com/ChrispinChaguza/RCandy

Examples

```
## Not run:
Read genome in GFF formatted file (generated usign readseq) and plot the genomic features

ref.genome.gff<-system.file("extdata", "Hungary19A-6.gff", package = "RCandy",mustWork = TRUE)

show.genome.annotation.plot(ref.genome.gff)

## End(Not run)
```

Index