Package ‘LifemapR’

March 6, 2025

Title Data Visualisation on 'Lifemap' Tree

Version 1.1.5

Imports leaflet, shiny, jsonlite, dplyr, tidyr, leaflet.minicharts,
htmltools, rlang, RCurl, fastmatch, arrow

Description Allow to visualise data on the NCBI phylogenetic tree as presented in Lifemap <https://lifemap.cnrs.fr/>. It takes as input a dataframe with at least a ``taxid'' column containing NCBI format TaxIds and allows to draw multiple layers with different visualisation tools.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Suggests knitr

VignetteBuilder knitr

URL https://lifemap-tol.github.io/LifemapR/,

https://github.com/Lifemap-ToL/LifemapR

BugReports https://github.com/Lifemap-ToL/LifemapR/issues

NeedsCompilation no

Author Cassandra Bompard [aut],
Damien M. de Vienne [aut] (<https://orcid.org/0000-0001-9532-5251>),
Julien Barnier [aut] (<https://orcid.org/0000-0002-8249-0909>),
Aurélie Siberchicot [ctb, cre]
(<https://orcid.org/0000-0002-7638-8318>)

Maintainer Aurélie Siberchicot <aurelie.siberchicot@univ-lyon1.fr>

Repository CRAN

Date/Publication 2025-03-06 10:40:12 UTC

Contents

`+.lifemap_obj` *Add a graphical element to a tree visualisation.*

Description

Add a graphical element to a tree visualisation.

Usage

```
## S3 method for class 'lifemap_obj'
e1 + e2
```

Arguments

`e1`	An object of class lifemap_obj that contains at least $df, a dataframe, and $basemap, the map used to get the coordinates.
`e2`	A description of the graphical features wanted for a set of points (eg. markers, subtree, piecharts, ...).

Value

A lifemap_obj object.

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
data(LM_eukaryotes)
LM_obj <- lifemap(LM_eukaryotes) + lm_markers() + lm_branches()
}
```

`add_lm_branches` *Compute the aesthetics for a subtree visualisation.*

Description

Compute the aesthetics for a subtree visualisation.

Usage

```
add_lm_branches(
  proxy,
  aes,
  df,
  df_visible,
  df_descendants,
  group_info,
  all_taxids
)
```

Arguments

`proxy`	The map to be modified.
`aes`	The dataframe containing the aesthetics details (must be of lm_branches class).
`df`	The full dataframe.
`df_visible`	The dataframe containing visible taxa.
`df_descendants`	The dataframe containing all the information on the descendants of visible taxa.
`group_info`	the ID of this group of lines.
`all_taxids`	A vector containing all the visible taxids and their direct descendants.

Value

An updated map with the new layer added.

`add_lm_markers` *Compute the aesthetics for markers visualisation.*

Description

Compute the aesthetics for markers visualisation.

Usage

```
add_lm_markers(proxy, aes, df, df_visible, group_info)
```

Arguments

`proxy`	The map to be modified.
`aes`	The dataframe containing the aesthetics information (must be of lm_markers class).
`df`	The full dataframe.
`df_visible`	The dataframe containing visible taxa.
`group_info`	The ID of this group of markers.

Value

An updated map with the new layer added.

`add_lm_piecharts` *Compute the aesthetics for discret values visualisation.*

Description

Compute the aesthetics for discret values visualisation.

Usage

```
add_lm_piecharts(proxy, aes, df, df_visible, layer)
```

Arguments

`proxy`	The map to be modified.
`aes`	The dataframe containing the aesthetics details (must be of lm_piecharts class).
`df`	The full dataframe.
`df_visible`	The dataframe containing visible taxa.
`layer`	The ID of this group of charts

Value

An updated map with the new layer added.

`build_Lifemap` *A function to construct a LifemapR object, usable by the other functions of the package.*

Description

A function to construct a LifemapR object, usable by the other functions of the package.

Usage

```
build_Lifemap(df, basemap = NULL, verbose = TRUE)
```

Arguments

`df`	A dataframe containing at least one column named "taxid" that contains NCBI Taxonomy Identifiers (taxid). The dataframe can contain any number of additional columns defining traits/characters/values associated to each taxid.
`basemap`	Deprecated argument.
`verbose`	If TRUE (the default), the function will print detailed information to the console. If FALSE, it will run silently.

Value

A list of class lifemap_obj containing:

- df : a dataframe containing at least for each taxid :
  - The x coordinate (lon)
  - The y coordonate (lat)
  - The scientific name (sci_name)
  - The zoom level at which the taxa is visible (zoom)
  - A list of its ascendants (ascend)
  - Its type ("requested" or "ancestor")
  - Its direct ancestor
  - Its type (type), i.e. whether the taxid was requested by the user ("requested") or if it is the anecestor of a requested taxid ("ancestor")
- basemap : the basemap used to get taxa’s details

Examples

```
data(eukaryotes_80)
## Not run:
# make sure you have a good internet connection to load these very large files
LM <- build_Lifemap(eukaryotes_80)

## End(Not run)
```

`create_matrix` *Create a dataframe for the ancestry.*

Description

Create a dataframe with pairs of taxids, each taxid and their corresponding values are associated with every one of its ancestors.

Usage

```
create_matrix(df, cols)
```

Arguments

`df`	A dataframe containing taxids and values.
`cols`	The columns containing the values which needs to be inferred.

Value

A dataframe.

Examples

```
data(LM_eukaryotes)

create_matrix(LM_eukaryotes$df, c("GC.", "Genes"))
```

`create_value_range` *Compute a new scale for a value*

Description

Compute a new scale for a value

Usage

```
create_value_range(value, df, df2, min, max)
```

Arguments

`value`	A vector of values.
`df`	The full dataframe.
`df2`	The dataframe containing visibles taxa.
`min`	The new minimum of the range.
`max`	The new maximum of the range.

Value

A vector of values.

`display_map` *Create a Lifemap base.*

Description

This function create a blank Leaflet map using the leaflet function from the leaflet package. If a dataframe is provided, it will be used for the creation of the map.

Usage

```
display_map(df = NULL, basemap = NULL)
```

Arguments

`df`	A dataframe. If given, its columns can be easily accessed with "~" (eg. ~GC.).
`basemap`	Deprecated argument.

Value

An HTML widget object with graphics layers.

Examples

```
display_map()
```

`display_option` *Compute the different display options.*

Description

Compute the different display options.

Usage

```
display_option(m, aes, df, type, leaves, i)
```

Arguments

`m`	The map to be modified.
`aes`	The dataframe containing the aesthetics details
`df`	The full dataframe.
`type`	A string indicating the type of representation, either "markers" or "discret"
`leaves`	The Vector of all the terminal taxids.
`i`	The index of the aesthetics.

Value

An updated map.

`draw_Lifemap` *Represent data on a Lifemap basemap.*

Description

Draw a map and all the aesthetics in the order you put them in, the last one will be on top of the others.

Usage

```
draw_Lifemap(lm_obj)
```

Arguments

`lm_obj`	A Lifemap object filled with aesthetics.

Value

A shiny application

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
  data(LM_eukaryotes)
  lifemap(LM_eukaryotes) + lm_markers() + lm_branches()
}
```

`eukaryotes_1000` *NCBI information for 1000 eukaryotes*

Description

A dataset containing NCBI information of 1000 eukaryotes. The variables are as follows:

Usage

```
data(eukaryotes_1000)
```

Format

A data frame with 1000 rows and 19 variables:

X.Organism.Name Organism name at the species level

taxid NCBI taxid

BioProject.Accession BioProject Accession number (from BioProject database)

BioProject.ID BioProject ID

Group Commonly used organism groups: Animals, Fungi, Plants, Protists

SubGroup NCBI Taxonomy level below group: Mammals, Birds, Fishes, Flatworms, Insects, Amphibians, Reptiles, Roundworms, Ascomycetes, Basidiomycetes, Land Plants, Green Algae, Apicomplexans, Kinetoplasts

Size..Mb. Total length of DNA submitted for the project

GC. Percent of nitrogenous bases (guanine or cytosine) in DNA submitted for the project

Assembly.Accession Name of the genome assembly (from NCBI Assembly database)

Replicons Number of replicons in the assembly

WGS Four-letter Accession prefix followed by version as defined in WGS division of GenBank/INSDC

Scaffolds Number of scaffolds in the assembly

Genes Number of Genes annotated in the assembly

Proteins Number of Proteins annotated in the assembly

Release.Date First public sequence release for the project

Modify.Date Sequence modification date for the project

Status Highest level of assembly: Chromosomes: one or more chromosomes are assembled Scaffolds or contigs: sequence assembled but no chromosomes

Center Origin of the sample

BioSample.Accession BioSample Accession number

`eukaryotes_80` *NCBI information for 80 eukaryotes*

Description

A dataset containing NCBI information of 80 eukaryotes. The variables are as follows:

Usage

```
data(eukaryotes_80)
```

Format

A data frame with 80 rows and 19 variables:

X.Organism.Name	Organism name at the species level
taxid	NCBI taxid
BioProject.Accession	BioProject Accession number (from BioProject database)
BioProject.ID	BioProject ID
Group	Commonly used organism groups: Animals, Fungi, Plants, Protists
SubGroup	NCBI Taxonomy level below group: Mammals, Birds, Fishes, Flatworms, Insects, Amphibians, Reptiles, Roundworms, Ascomycetes, Basidiomycetes, Land Plants, Green Algae, Apicomplexans, Kinetoplasts
Size..Mb.	Total length of DNA submitted for the project
GC.	Percent of nitrogenous bases (guanine or cytosine) in DNA submitted for the project
Assembly.Accession	Name of the genome assembly (from NCBI Assembly database)
Replicons	Number of replicons in the assembly
WGS	Four-letter Accession prefix followed by version as defined in WGS division of GenBank/INSDC
Scaffolds	Number of scaffolds in the assembly
Genes	Number of Genes annotated in the assembly
Proteins	Number of Proteins annotated in the assembly
Release.Date	First public sequence release for the project
Modify.Date	Sequence modification date for the project
Status	Highest level of assembly: Chromosomes: one or more chromosomes are assembled Scaffolds or contigs: sequence assembled but no chromosomes
Center	Origin of the sample
BioSample.Accession	BioSample Accession number

`gen_res` *Genomic results*

Description

A dataset containing information on Genome size and TE content for 808 taxids

Usage

```
data(gen_res)
```

Format

A data frame with 808 rows and 3 variables:

taxid	NCBI taxid
Genome_size	the Genome size in pb
TEcontent_bp	the transposable element content in pb

`is.lifemap_obj` *Reports whether x is a lifemap_obj object.*

Description

Reports whether x is a lifemap_obj object.

Usage

```
is.lifemap_obj(x)
```

Arguments

`x`	The object to test.

Value

A boolean indicating whether or not the object is of lifemap_obj type.

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
data(LM_eukaryotes)
is.lifemap_obj(LM_eukaryotes)
}
```

`is.lm_branches` *Reports whether x is a lm_branches object.*

Description

Reports whether x is a lm_branches object.

Usage

```
is.lm_branches(x)
```

Arguments

`x`	The object to test.

Value

A boolean indicating whether or not the object is of lm_branches type.

`is.lm_markers` *Reports whether x is a lm_markers object.*

Description

Reports whether x is a lm_markers object.

Usage

```
is.lm_markers(x)
```

Arguments

`x`	The object to test.

Value

A boolean indicating whether or not the object is of lm_markers type.

`is.lm_piecharts` *Reports whether x is a lm_branches object.*

Description

Reports whether x is a lm_branches object.

Usage

```
is.lm_piecharts(x)
```

Arguments

`x`	The object to test.

Value

A boolean indicating whether or not the object is of lm_piecharts type.

`kraken_res` *Kraken results*

Description

A dataset containing NCBI information of 1000 eukaryotes. The variables are as follows:

Usage

```
data(kraken_res)
```

Format

A data frame with 4427 rows and 6 variables:

coverage_percent Percentage of fragments covered by the clade rooted at this taxon

coverage_number Number of fragments covered by the clade rooted at this taxon

fragment_number Number of fragments assigned directly to this taxon

rank A rank code, indicating (U)nclassified, (R)oot, (D)omain, (K)ingdom, (P)hylum, (C)lass, (O)rder, (F)amily, (G)enus, or (S)pecies. Taxa that are not at any of these 10 ranks have a rank code that is formed by using the rank code of the closest ancestor rank with a number indicating the distance from that rank. E.g., "G2" is a rank code indicating a taxon is between genus and species and the grandparent taxon is at the genus rank.

taxid NCBI taxonomic ID number

name Indented scientific name

`lifemap` *Initialise a new Lifemap visualisation.*

Description

Initialise a lifemap_obj object. It can be used to describe aesthetics like the zoom level at which taxids becomes visible...

Usage

```
lifemap(lm_obj, zoom = 4)
```

Arguments

`lm_obj`	lifemap_obj object used for data visualisation.
`zoom`	The level of zoom for which values are displayed (if zoom = 0 only the nodes at the current zoom level or lower can have their values displayed).

Value

A lifemap_obj object.

`lm_branches` *Add a "branches" layer to a lifemap_obj object.*

Description

Add a "branches" layer to a lifemap_obj object.

Usage

```
lm_branches(
  data = NULL,
  color = NULL,
  var_color = NULL,
  size = 5,
  min = 2,
  max = 20,
  opacity = 0.5,
  FUN = NULL,
  legend = TRUE,
  legendPosition = c("topright", "bottomright", "bottomleft", "topleft")
)
```

Arguments

`data`	A sub dataset to use, if NULL then all of the taxids from the lifemap object given to lifemap() will be used.
`color`	Either a color for the branches or a palette if a variable is used to represent branches' color.
`var_color`	A column name of the original dataframe to represent this variable by the color of branches.
`size`	Either a numeric for the branche's thickness or a variable to be represented by the branche's thickness.
`min`	An integer indicating the minimal thickness of the branches if the size is a column name.
`max`	An integer indicating the maximal thickness of the branches if the size is a column name.
`opacity`	An integer indicating branche's opacity.
`FUN`	The function to be applied to infer values. If NULL values won't be inferred
`legend`	A logical indicating whether or not to display the legend.
`legendPosition`	c("topright", "bottomright", "bottomleft", "topleft"), the position of the legend.

Value

An lm_branches object containing all aesthetics details for one layer of branches

Examples

```
data(LM_eukaryotes)

lm_branches(var_color = "GC.", color = "Accent")

lm_branches(data = LM_eukaryotes$df[LM_eukaryotes$df$Group %in% "Plants",])
```

`LM_eukaryotes` *Transformation in a LifemapR format of NCBI information for 1000 eukaryotes*

Description

A dataset containing NCBI information of 1000 eukaryotes. The variables are as follows:

Usage

```
data(LM_eukaryotes)
```

Format

A lifemap object - a list containing the basemap used to fetch data and df, a data frame with 2760 rows and 26 variables:

X.Organism.Name Organism name at the species level

taxid NCBI taxid

BioProject.Accession BioProject Accession number (from BioProject database)

BioProject.ID BioProject ID

Group Commonly used organism groups: Animals, Fungi, Plants, Protists

SubGroup NCBI Taxonomy level below group: Mammals, Birds, Fishes, Flatworms, Insects, Amphibians, Reptiles, Roundworms, Ascomycetes, Basidiomycetes, Land Plants, Green Algae, Apicomplexans, Kinetoplasts

Size..Mb. Total length of DNA submitted for the project

GC. Percent of nitrogenous bases (guanine or cytosine) in DNA submitted for the project

Assembly.Accession Name of the genome assembly (from NCBI Assembly database)

Replicons Number of replicons in the assembly

WGS Four-letter Accession prefix followed by version as defined in WGS division of GenBank/INSDC

Scaffolds Number of scaffolds in the assembly

Genes Number of Genes annotated in the assembly

Proteins Number of Proteins annotated in the assembly

Release.Date First public sequence release for the project

Modify.Date Sequence modification date for the project

Status Highest level of assembly: Chromosomes: one or more chromosomes are assembled Scaffolds or contigs: sequence assembled but no chromosomes

Center Origin of the sample

BioSample.Accession BioSample Accession number

lon longitude of taxids on a specific basemap

lat latitude of taxids on a specific basemap

sci_name scientific name of taxids

zoom zoom of taxids on a specific basemap

ascend the list of all ancestors of taxids on a specific basemap

type either "requested" if the taxid was given, "ancestor" if gotten from the database

ancestor the direct ancestor oftaxids on a specific basemap

`lm_markers` *add a "markers" layer to a lifemap_obj object.*

Description

Adds a layer with circles that can represent data. The main parameters that can be used to represent data are the following:

- radius
- var_fillColor
- var_color (stroke's color)

Usage

```
lm_markers(
  data = NULL,
  radius = 10,
  min = 10,
  max = 40,
  fillColor = NULL,
  var_fillColor = NULL,
  fillOpacity = 0.8,
  stroke = FALSE,
  color = NULL,
  var_color = NULL,
  weight = 1,
  opacity = 0.8,
  legend = TRUE,
  legendPosition = c("topright", "bottomright", "bottomleft", "topleft"),
  legendOrientation = c("vertical", "horizontal"),
  legendOpacity = 0.5,
  FUN = NULL,
```

```
  display = c("auto", "requested", "all", "leaves"),
  popup = NULL,
  label = NULL
)
```

Arguments

`data`	A sub dataset to use, if NULL then all of the taxids from the lifemap object given to lifemap() will be used.
`radius`	Either a numerical value or a column name of the original dataframe to represent this variable by the size of markers.
`min`	An integer indicating the minimal size of the markers if radius is a column name.
`max`	An integer indicating the maximal size of the markers if radius is a column name.
`fillColor`	Either a color or a palette if the fillColor is represented by a variable.
`var_fillColor`	A column name of the original dataframe to represent this variable by the fillColor of markers.
`fillOpacity`	A numeric indicating the fill opacity.
`stroke`	A logical indicating whether to draw a border for markers.
`color`	The stroke color. Either a color or a palette if the stroke color is represented by a variable.
`var_color`	A column name of the original dataframe to represent this variable by the stroke color.
`weight`	The stroke width in pixels.
`opacity`	The stroke opacity.
`legend`	Whether to display the legend or not.
`legendPosition`	c("topright", "bottomright", "bottomleft", "topleft").
`legendOrientation`	c("vertical", "horizontal").
`legendOpacity`	Legend opacity (applies on the shapes in the legend, not the background itself).
`FUN`	The function to be applied to infer values. If NULL then the information missing from the parent nodes won't be inferred.
`display`	c("auto", "requested", "all", "leaves"), a string indicating how to display markers : • "auto" : the markers are displayed depending on the zoom level, by default, allow to have a lot of markers • "requested" : only displays the requested taxa, but all at the same time • "all" : displays all the taxa including all the ancestors to the root • "leaves" : displays only the latest (most recent) taxa (WARNING : "requested", "leaves" and "auto" shouldn't be used to display more than 2000 markers as it may result in long computing time).
`popup`	A column name indicating what to display when clicking on a node.
`label`	A column name indicating what to display when hovering on a node.

Value

An lm_markers object containing all aesthetics details for one layer of markers.

Examples

```
data(LM_eukaryotes)

lm_markers(data = LM_eukaryotes$df[LM_eukaryotes$df$Group %in% "Plants", ])

lm_markers(radius = "GC.", var_fillColor = "Genes")
```

`lm_piecharts` *Add a "piecharts" layer to a lifemap_obj object.*

Description

Add a "piecharts" layer to a lifemap_obj object.

Usage

```
lm_piecharts(
  data = NULL,
  param,
  type = c("pie", "bar", "polar-area", "polar-radius", "auto"),
  width = 30,
  height = 30,
  opacity = 1,
  showLabels = FALSE,
  pal = "Accent",
  legend = TRUE,
  legendPosition = c("topright", "bottomright", "bottomleft", "topleft"),
  display = c("auto", "requested", "all", "leaves")
)
```

Arguments

`data`	A sub dataset to use, if NULL then all of the taxids from the lifemap object given to lifemap() will be used.
`param`	A column name indicating the discret variable to be represented.
`type`	The type of chart to draw.
`width`	The maximal width of the charts.
`height`	The maximal height of the charts.
`opacity`	The chart's opacity.
`showLabels`	A boolean indicating whether to display the values directly on the chart or not.

`pal`	The palette to be used for the charts.
`legend`	A boolean indiacting whether to draw the legend or not.
`legendPosition`	c("topright", "bottomright", "bottomleft", "topleft"). Where should the legend be placed.
`display`	c("auto", "requested", "all", "leaves"), a string indicating how to display charts : • "auto" : the markers are displayed depending on the zoom level, by default, allow to have a lot of charts • "requested" : only displays the requested taxa, but all at the same time • "all" : displays all the taxa including all the ancestors to the root • "leaves" : displays only the latest (most recent) taxa (WARNING : "requested", "leaves" and "auto" shouldn't be used to display more than 2000 charts as it may result in long computing time)

Value

An lm_piecharts object containing all aesthetics details for one layer of charts

Examples

```
data(LM_eukaryotes)

lm_piecharts(param = "Status")

lm_piecharts(data = LM_eukaryotes$df[LM_eukaryotes$df$Group %in% "Plants", ], param = "Status")
```

`make_newick` *Create a newick for the given dataset.*

Description

Create a newick for the given dataset.

Usage

```
make_newick(df)
```

Arguments

`df`	A dataframe either enriched with by the build_Lifemap function or containing at least two columns : • "taxid" containing NCBI format taxids • "ancestor" containing for each taxid, it's direct ancestor in the phylogeny

Value

A character string.

`pass_infos` *Infer numerical values to nodes.*

Description

Infer numerical values to nodes.

Usage

```
pass_infos(M, FUN, value)
```

Arguments

`M`	The dataframe returned by create_matrix.
`FUN`	The function to be applied when inferring the values.
`value`	The column name to which the function applies.

Value

An array of values.

Examples

```
data(LM_eukaryotes)

infos <- create_matrix(LM_eukaryotes$df, c("GC.", "Genes"))

inferred_values <- pass_infos(M = infos, FUN = mean, value = "GC.")
```

`pass_infos_discret` *Infer discret values to nodes for lm_piecharts function.*

Description

Infer discret values to nodes for lm_piecharts function.

Usage

```
pass_infos_discret(M, value)
```

Arguments

`M`	The dataframe returned by create_matrix.
`value`	The column name represented as piecharts.

Value

A dataframe containing the TaxIDs and as many columns as there are distinct values.

Examples

```
data(LM_eukaryotes)

infos <- create_matrix(LM_eukaryotes$df, "Status")

inferred_values <- pass_infos_discret(M = infos, value = "Status")
```

`print.lifemap_obj` *Method to print lifemap_obj objects.*

Description

Method to print lifemap_obj objects.

Usage

```
## S3 method for class 'lifemap_obj'
print(x, ...)
```

Arguments

`x`	An lifemap_obj.
`...`	Further arguments passed to or from other methods.

Value

Either a description of the dataframe and basemap used for the lm_obj object, or a shiny application if aesthetics are furnished

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
data(LM_eukaryotes)
print(LM_eukaryotes)
}
```

Index