

# Package ‘LifemapR’

August 24, 2023

**Title** Data Visualisation on 'Lifemap' Tree

**Version** 1.0.4

**Imports** leaflet, shiny, jsonlite, dplyr, leaflet.minicharts, purrr,  
htmltools, rlang

**Description** Allow to visualise data on the NCBI phylogenetic tree as presented in Lifemap '<http://lifemap.univ-lyon1.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** 7.2.3

**Depends** R (>= 3.5.0)

**Suggests** knitr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Cassandra Bompard [aut],  
Damien M. de Vienne [aut],  
Aur lie Siberchicot [ctb, cre]

**Maintainer** Aur lie Siberchicot <aurelie.siberchicot@univ-lyon1.fr>

**Repository** CRAN

**Date/Publication** 2023-08-24 08:20:06 UTC

## R topics documented:

+.lifemap_obj . . . . .	2
add_lm_branches . . . . .	3
add_lm_markers . . . . .	4
add_lm_piecharts . . . . .	4
build_Lifemap . . . . .	5
create_matrix . . . . .	6
create_value_range . . . . .	6

display_map . . . . .	7
display_option . . . . .	7
draw_Lifemap . . . . .	8
eukaryotes_1000 . . . . .	9
eukaryotes_80 . . . . .	10
gen_res . . . . .	11
get_direct_ancestor . . . . .	11
is.lifemap_obj . . . . .	12
is.lm_branches . . . . .	12
is.lm_markers . . . . .	13
is.lm_piecharts . . . . .	13
kraken_res . . . . .	14
lifemap . . . . .	14
lm_branches . . . . .	15
LM_eukaryotes . . . . .	16
lm_markers . . . . .	17
lm_piecharts . . . . .	19
make_newick . . . . .	20
pass_infos . . . . .	21
pass_infos_discret . . . . .	21
print.lifemap_obj . . . . .	22
request_database . . . . .	23
url_verification . . . . .	23

## Index 24

---

+.lifemap_obj	<i>Add a graphical element to a tree visualisation.</i>
---------------	---

---

### Description

Add a graphical element to a tree visualisation.

### Usage

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

### Arguments

- |    |   |
|----|---|
| e1 | An object of class <code>lifemap_obj</code> that contains at least <code>\$df</code> , a dataframe, and <code>\$basemap</code> , 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	<i>Compute the aesthetics for a subtree visualisation.</i>
-----------------	--

---

## 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	<i>A function to construct a LifemapR object, usable by the other functions of the package.</i>
---------------	---

---

## Description

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

## Usage

```
build_Lifemap(df, basemap = c("fr", "ncbi", "base", "virus"), verbose = TRUE)
```

## Arguments

df	A dataframe containing at least a column named "taxid" including NCBI format TaxIDs. The dataframe can also contain characteristics associated with those TaxIDs in separated columns.
basemap	The basemap wanted ("fr", "ncbi", "base" or "virus").
verbose	If TRUE, will write details on the status of the operation in the terminal.

## Value

A lifemap object with :

- df : a dataframe containing at least for each TaxID (taxid) :
  - The longitude (long)
  - The latitude (lat)
  - The scientific name (sci\_name)
  - The zoom level at which the taxa can be seen (zoom)
  - Its ascendants (ascend)
  - Its type ("requested" or "ancestor")
- basemap : the basemap used to get taxa's details

## Examples

```
data(eukaryotes_80)
LM <- build_Lifemap(eukaryotes_80, "fr")
```

---

create_matrix	<i>Create a dataframe for the ancestry.</i>
---------------	---

---

**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	<i>Compute a new scale for a value</i>
--------------------	--

---

**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	<i>Create a Lifemap base.</i>
-------------	-------------------------------

---

**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 = c("fr", "ncbi", "base", "virus"))
```

**Arguments**

df	A dataframe. If given, its columns can be easily accessed with "~" (eg. ~GC).
basemap	The basemap choosen to be displayed, it can be either : <ul style="list-style-type: none"> <li>• 'fr' for <a href="https://lifemap-fr.univ-lyon1.fr/">https://lifemap-fr.univ-lyon1.fr/</a></li> <li>• 'ncbi' for <a href="https://lifemap-ncbi.univ-lyon1.fr/">https://lifemap-ncbi.univ-lyon1.fr/</a></li> <li>• 'base' for <a href="https://lifemap.univ-lyon1.fr/">https://lifemap.univ-lyon1.fr/</a></li> <li>• 'virus' for <a href="http://virusmap.univ-lyon1.fr/">http://virusmap.univ-lyon1.fr/</a></li> </ul>

**Value**

An HTML widget object with graphics layers.

**Examples**

```
display_map()
```

---

display_option	<i>Compute the different display options.</i>
----------------	---

---

**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	<i>Represent data on a Lifemap basemap.</i>
--------------	---

---

**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

---

`get_direct_ancestor`*Add the direct ancestor for each taxa of the dataframe.*

---

**Description**

Add the direct ancestor for each taxa of the dataframe.

**Usage**

```
get_direct_ancestor(df)
```

**Arguments**

`df` A dataframe with requested taxid and the full ascendance information.

**Value**

A dataframe with the direct ancestor for each taxa.

---

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	<i>Reports whether x is a lm_markers object.</i>
---------------	--

---

**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	<i>Reports whether x is a lm_branches object.</i>
-----------------	---

---

**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	<i>Kraken results</i>
------------	-----------------------

---

**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	<i>Initialise a new Lifemap visualisation.</i>
---------	--

---

**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	<i>Add a "branches" layer to a lifemap_obj object.</i>
-------------	--

---

### 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	<i>Transformation in a LifemapR format of NCBI information for 1000 eukaryotes</i>
---------------	--

---

**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 of taxids 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 fill-Color 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 : <ul style="list-style-type: none"> <li>• "auto" : the markers are displayed depending on the zoom level, by default, allow to have a lot of markers</li> <li>• "requested" : only displays the requested taxa, but all at the same time</li> <li>• "all" : displays all the taxa including all the ancestors to the root</li> <li>• "leaves" : displays only the latest (most recent) taxa</li> </ul> (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	<i>Add a "piecharts" layer to a lifemap_obj object.</i>
--------------	---

---

**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 indicating 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 : <ul style="list-style-type: none"> <li>• "auto" : the markers are displayed depending on the zoom level, by default, allow to have a lot of charts</li> <li>• "requested" : only displays the requested taxa, but all at the same time</li> <li>• "all" : displays all the taxa including all the ancestors to the root</li> <li>• "leaves" : displays only the latest (most recent) taxa</li> </ul> (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	<i>Create a newick for the given dataset.</i>
-------------	---

---

**Description**

Create a newick for the given dataset.

**Usage**

```
make_newick(df)
```

**Arguments**

df	A dataframe either enriched with by the <code>build_Lifemap</code> function or containing at least two columns : <ul style="list-style-type: none"> <li>• "taxid" containing NCBI format taxids</li> <li>• "ancestor" containing for each taxid, it's direct ancestor in the phylogeny</li> </ul>
----	---

**Value**

A character string.

---

pass_infos	<i>Infer numerical values to nodes.</i>
------------	---

---

**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	<i>Infer discret values to nodes for lm_piecharts function.</i>
--------------------	---

---

**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)
}
```

---

request_database	<i>Request one of the core of the solr database corresponding to the basemap chosen with the TaxIDs wanted.</i>
------------------	---

---

**Description**

Request one of the core of the solr database corresponding to the basemap chosen with the TaxIDs wanted.

**Usage**

```
request_database(taxids, basemap, core)
```

**Arguments**

taxids	A vector of TaxIDs to be requested.
basemap	The name of the basemap on which the data is represented (can be "ncbi", "fr", "base" or "virus").
core	The core to be requested either "taxo" (for position details) or "addi" (for ascendance details).

**Value**

A dataframe containing all the pieces of information requested from the database.

---

url_verification	<i>Check if the URL is working</i>
------------------	------------------------------------

---

**Description**

Check if the URL is working

**Usage**

```
url_verification(basemap_url, t = 20)
```

**Arguments**

basemap_url	The url corresponding to the basemap interrogated.
t	The time before timeout.

**Value**

A logical.

# Index

## \* datasets

- eukaryotes\_1000, 9
- eukaryotes\_80, 10
- gen\_res, 11
- kraken\_res, 14
- LM\_eukaryotes, 16

+ .lifemap\_obj, 2

- add\_lm\_branches, 3
- add\_lm\_markers, 4
- add\_lm\_piecharts, 4

- build\_Lifemap, 5

- create\_matrix, 6
- create\_value\_range, 6

- display\_map, 7
- display\_option, 7
- draw\_Lifemap, 8

- eukaryotes\_1000, 9
- eukaryotes\_80, 10

- gen\_res, 11
- get\_direct\_ancestor, 11

- is.lifemap\_obj, 12
- is.lm\_branches, 12
- is.lm\_markers, 13
- is.lm\_piecharts, 13

- kraken\_res, 14

- lifemap, 14
- lm\_branches, 15
- LM\_eukaryotes, 16
- lm\_markers, 17
- lm\_piecharts, 19

- make\_newick, 20

- pass\_infos, 21

- pass\_infos\_discret, 21
- print.lifemap\_obj, 22

- request\_database, 23

- url\_verification, 23