

Package ‘SMITIDvisu’

March 5, 2019

Type Package

Title Visualize Data for Host and Viral Population from 'SMITIDstruct' using HTMLwidgets

Version 0.0.4

Description Visualisation tools for 'SMITIDstruct' package.
Allow to visualize host timeline, transmission tree, index diversities and variant graph using HTMLwidgets. It mainly using D3JS javascript framework.

Date 2019-03-05

Depends R (>= 3.4.0), utils

LinkingTo Rcpp

NeedsCompilation yes

SystemRequirements C++11

Biarch true

License GPL (>= 3) | file LICENSE

URL <https://informatique-mia.inra.fr/biosp/anr-smitid-project>,
<https://gitlab.paca.inra.fr/SMITID/visu>

Encoding UTF-8

LazyData true

Imports Rcpp (>= 0.11.0), htmlwidgets (>= 0.3.2), yaml (>= 2.1.16),
jsonlite (>= 1.5.0), magrittr

Suggests SMITIDstruct, knitr, shiny

RoxygenNote 6.1.0

Author Jean-Francois Rey [aut, cre]

Maintainer Jean-Francois Rey <jean-francois.rey@inra.fr>

Repository CRAN

Date/Publication 2019-03-05 13:30:03 UTC

R topics documented:

SMITIDvisu-package	2
demo.SMITIDvisu.run	3
hostline	3
mstCompute	4
mstVariant	4
mstVariantProxy	5
SMITIDvisu-shiny	6
st	6
st.dist113_2	7
st.dist113_all	7
st.listTimeProp113	8
st.prop113_2	8
st.prop113_all	9
timeLine	9
timeLineProxy	10
transmissionTree	11
tt.edges	11
tt.nodes	12
updatemstVariant	13
updateTimeLine	13

Index	15
--------------	-----------

SMITIDvisu-package	<i>Visualize Data for Host and Viral Population from SMITIDstruct using HTMLwidgets</i>
--------------------	---

Description

Visualisation tools for SMITIDstruct package. Allow to visualize host timeline, transmission tree, index diversities and variant graph using HTMLwidgets. It mainly using D3JS, noUiSlider and FileSaver javascript libraries.

Details

Package: SMITIDvisu
 Type: Package
 Version: 0.0.4
 Date: 2019-03-05
 License: GPL (>=3)

Author(s)

Jean-Francois Rey <jean-francois.rey@inra.fr>

Examples

```
library(SMITIDvisu)
demo.SMITIDvisu.run()
```

demo.SMITIDvisu.run *demo.SMITIDvisu.run*

Description

run a demo to visualize data

Usage

```
demo.SMITIDvisu.run()
```

hostline *A host infomation over time*

Description

kind of host time line

Usage

```
data("hostline")
```

Format

A data frame with 8 observations on the following 5 variables.

- level a character vector
- label a character vector
- ID a character vector
- timestart a character vector
- timeend a character vector

Examples

```
data(hostline)
print(hostline)
```

mstCompute	<i>compute the minimum spanning tree</i>
------------	--

Description

compute the minimum spanning tree of a matrix representing edges between nodes (of a graph)

Usage

```
mstCompute(mat)
```

Arguments

mat	weighted matrix representing nodes connection (edges weight)
-----	--

Value

a matrix with 1 if nodes are linked, 0 otherwise.

mstVariant	<i>mstVariant</i>
------------	-------------------

Description

Draw Variants genotypes distances as a graph using Minimum Spanning Tree algorithm.

Usage

```
mstVariant(mat, prop, node.prop = NULL, width = NULL, height = NULL,
  elementId = NULL)
```

Arguments

mat	a distance matrix between sequence of variants (interger distance no floating values)
prop	a data.frame for variants sequences proportions and count (see details)
node.prop	list of variants with proportions and time (default NULL)
width	numeric width for the area in pixels.
height	numeric hieght for the area in pixels.
elementId	the element ID where is draw

Details

mat is a simple distance matrix with interger values, row and lines contain a unique identifier of each variant sequences. **prop** is a data.frame where each row is a variant sequence, it have to contain in columns factor "ID", "proportion" and "count". "ID" is a unique identifier matching matrix value identifier, "proportion" is the proportions of the variant sequence and "count" the number of variant sequence in a varions set. **node.prop** is a list with name that matching **mat** identifier and **prop** "ID". Each list element contains a subvector time (Julian or timestamp) and value (proportions). That allow to draw variants proportions over time.

Examples

```
library(SMITIDvisu)
data(st)
mstV <- mstVariant(st.dist113_all,st.prop113_all, st.listTimeProp113)

## export as standalone html file
htmlwidgets::saveWidget(mstV, "mstVariant.html")
browseURL("mstVariant.html")
```

mstVariantProxy

mstVariantProxy

Description

```
get mstVariantProxy
```

Usage

```
mstVariantProxy(mstVid, session = shiny::getDefaultReactiveDomain())
```

Arguments

mstVid	widget instance identifier
session	shiny session

Examples

```
## Not run:
library(SMITIDvisu)
## server.R
mstVariantProxy <- mstVaraintProxy("mstvariantoutput")

## End(Not run)
```

 SMITIDvisu-shiny

Shiny bindings for visualisation widgets

Description

Output and render functions for using visualisation widgets within Shiny applications and interactive Rmd documents.

Usage

```
mstVariantOutput(outputId, width = "100%", height = "600px")
```

```
rendermstVariant(expr, env = parent.frame(), quoted = FALSE)
```

```
timeLineOutput(outputId, width = "100%", height = "400px")
```

```
renderTimeLine(expr, env = parent.frame(), quoted = FALSE)
```

```
transmissionTreeOutput(outputId, width = "100%", height = "500px")
```

```
renderTransmissionTree(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended.
expr	An expression that generates a networkD3 graph
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

 st

A SMITIDstruct package variable.

Description

A SMITIDstruct package variable from simul-chain as a list. The list is a set of HostSet, ViralPopset and an index

Usage

```
data("st")
```

Format

The format is: List of 3 \$:List of 21- attr(*, "class")= chr "HostSet" \$:List of 20- attr(*, "class")= chr "ViralPopSet" \$:'data.frame': 79 obs. of 3 variables: ..\$ TIME : chr [1:79] "0" "0" "1.26" "1.35"\$ ID_HOST : chr [1:79] "1" "2" "2" "2"\$ EVENTCODE: chr [1:79] "000011" "000110" "001000" "001000" ...

Examples

```
data(st)
## maybe str(st) ; plot(st) ...
```

st.dist113_2	<i>Distance matrix of observed variants sequences of a host 113 at time 2 from simulation.</i>
--------------	--

Description

Levenshtein Distance matrix with rows and cols label as sequences ID.

Usage

```
data("st")
```

Format

The format is: num [1:23, 1:23] 0 1 1 1 2 1 1 1 1 2 ...

Examples

```
data(st)
```

st.dist113_all	<i>Distance matrix of observed variants sequences of a host 113 at time 2, 3 and 4 from simulation.</i>
----------------	---

Description

Levenshtein Distance matrix with rows and cols label as sequences ID. Unique sequence variants observed on host 113 at time 2, 3 and 4 from a simulation.

Usage

```
data("st")
```

Format

The format is: num [1:51, 1:51] 0 1 1 1 1 1 3 1 1 1 ...

Examples

```
data(st)
```

```
st.listTimeProp113
```

List of variants ID with subvector for time and value.

Description

A list indexed by variants sequences ID. Each element contain a time and value vector for time of observation and proportions observed at this time.

Usage

```
data("st")
```

Examples

```
data(st)
```

```
st.prop113_2
```

Variants proportions and count for host 113 at time 2 from simulation.

Description

A data.frame with label "ID", "proportion" and "count" for an host 113 at time 2 from simulation. Each row is a sequence.

Usage

```
data("st")
```

Format

A data frame with 23 observations on the following 3 variables.

ID a character vector

proportion a numeric vector

count a numeric vector

Examples

```
data(st)
```

st.prop113_all	<i>Variants proportions and count for an host 113 at time 2, 3 and 4 from simulation.</i>
----------------	---

Description

A data.frame with label "ID", "proportion" and "count" for an host 113 at time 2, 3 and 4 from simulation. Each row is a sequence.

Usage

```
data("st")
```

Format

A data frame with 51 observations on the following 3 variables.

ID a character vector

proportion a numeric vector

count a numeric vector

Examples

```
data(st)
```

timeLine	<i>timeLine</i>
----------	-----------------

Description

Draw a host time line. Time use timestamp or Date in ISO format.

Usage

```
timeLine(data, title, color = list(default = "black"), width = NULL,
  height = NULL, elementId = NULL)
```

Arguments

data	a data.frame that represent hosts status in time with ID, status and time in columns
title	a title as character
color	list of color for timeline elements
width	numeric width for the area in pixels.
height	numeric height for the area in pixels.
elementId	the element ID where is draw

Examples

```

library(SMITIDvisu)
data(hostline)
tl <- timeLine(hostline,
               title="Example host 113",
               color=list("infected"="red","offspring"="green",
                          "alive"="blue","inf"="orange",
                          "dead"="black","Obs"="purple"))

## export as standalone html file
htmlwidgets::saveWidget(tl, "timeline.html")
browseURL("timeline.html")

```

timeLineProxy

timeLineProxy get an instance of a timeline

Description

timeLineProxy get an instance of a timeline

Usage

```
timeLineProxy(tlid, session = shiny::getDefaultReactiveDomain())
```

Arguments

tlid	a timeline instance id
session	shiny session

Value

an object of class timeline_proxy

Examples

```

## Not run:
## server.R
## output server variable
output$timeline <- renderTimeLine({
  timeLine(data.frame(), "")
})
## ui.R
timeLineOutput("timeline")
## server.R
tlproxy <- timeLineProxy("timeline")

## End(Not run)

```

transmissionTree	<i>transmissionTree</i>
------------------	-------------------------

Description

Draw a transmission tree over the time. Time use timestamp or Date in ISO format ("

Usage

```
transmissionTree(nodes, edges, nodes.color = list(default = "black"),
  width = NULL, height = NULL, elementId = NULL)
```

Arguments

nodes	a data.frame that represent hosts status in time with ID, status and time in columns
edges	a data.frame that represent transmission link between hosts (pathogens) with ID, source, target and time in columns
nodes.color	a list of color for nodes status "status"="color"
width	numeric width for the area in pixels.
height	numeric height for the area in pixels.
elementId	the element ID where is draw

Examples

```
library(SMITIDvisu)
data(transmissiontree)
tt <- transmissionTree(tt.nodes,tt.edges, nodes.color = list("default"="black","Inf"="red"))

## export as standalone html file
htmlwidgets::saveWidget(tt, "transTree.html")
browseURL("transTree.html")
```

tt.edges	<i>Pathogen link over the time</i>
----------	------------------------------------

Description

A data.frame of all transmission links between hosts (pathogens). Four columns ID, source, target and time.

Usage

```
data("transmissiontree")
```

Format

A data frame with 46 observations on the following 4 variables.

ID a factor with levels 1 10 11 12 13 14 15 16 17 18 19 2 20 21 22 23 24 25 26 27 28 29 3 30 31
32 33 34 35 36 37 38 39 4 40 41 42 43 44 45 46 5 6 7 8 9

source a character vector

target a character vector

time a character vector

Examples

```
data(transmissiontree)
print(tt.edges)
```

tt.nodes

Host list with there status over the time.

Description

a data.frame of all the hosts identify by there ID. Three columns is use ID, status and time

Usage

```
data("transmissiontree")
```

Format

A data frame with 101 observations on the following 3 variables.

ID a character vector

time a character vector

status a character vector

Examples

```
data(transmissiontree)
print(tt.nodes)
```

updateMstVariant	<i>updateMstVariant</i>
------------------	-------------------------

Description

update (redraw) an instance on mstVariant

Usage

```
updateMstVariant(mstVProxy, mat, prop, propTime = NULL)
```

Arguments

mstVProxy	mstVaraintProxy instance
mat	distance matrix
prop	proportions data.frame
propTime	list of each variant by time and proportions

See Also

[mstVariant](#)

Examples

```
## Not run:  
library(SMITIDvisu)  
data(mstVariant)  
## server.R  
mstVaraintProxy("mstvariantoutput") %>% updateMstVariant(st.dist,st.prop)  
  
## End(Not run)
```

updateTimeLine	<i>updateTimeLine</i>
----------------	-----------------------

Description

updateTimeLine

Usage

```
updateTimeLine(tlProxy, data, title)
```

Arguments

tlProxy	a timeline proxy instance
data	new data
title	new title

See Also

[timeLine](#)

Examples

```
## Not run:
## server.R
## output server variable
output$timeline <- renderTimeLine({
  timeLine(data.frame(), "")
})
## ui.R
timeLineOutput("timeline")
## server.R
timeLineProxy("timeline") %>% updateTimeLine(newtimeline, "newId")

## End(Not run)
```

Index

*Topic **datasets**

- hostline, [3](#)
 - st, [6](#)
 - st.dist113_2, [7](#)
 - st.dist113_all, [7](#)
 - st.listTimeProp113, [8](#)
 - st.prop113_2, [8](#)
 - st.prop113_all, [9](#)
 - tt.edges, [11](#)
 - tt.nodes, [12](#)
- demo.SMITIDvisu.run, [3](#)
- hostline, [3](#)
- mstCompute, [4](#)
- mstVariant, [4](#), [13](#)
- mstVariantOutput (SMITIDvisu-shiny), [6](#)
- mstVariantProxy, [5](#)
- rendermstVariant (SMITIDvisu-shiny), [6](#)
- renderTimeLine (SMITIDvisu-shiny), [6](#)
- renderTransmissionTree
(SMITIDvisu-shiny), [6](#)
- SMITIDvisu (SMITIDvisu-package), [2](#)
- SMITIDvisu-package, [2](#)
- SMITIDvisu-shiny, [6](#)
- st, [6](#)
- st.dist113_2, [7](#)
- st.dist113_all, [7](#)
- st.listTimeProp113, [8](#)
- st.prop113_2, [8](#)
- st.prop113_all, [9](#)
- timeLine, [9](#), [14](#)
- timeLineOutput (SMITIDvisu-shiny), [6](#)
- timeLineProxy, [10](#)
- transmissionTree, [11](#)
- transmissionTreeOutput
(SMITIDvisu-shiny), [6](#)
- tt.edges, [11](#)
- tt.nodes, [12](#)
- updatemstVariant, [13](#)
- updateTimeLine, [13](#)