Package ‘SMITIDstruct’

June 14, 2019

Type Package
Encoding UTF-8
Title Data Structure and Manipulations Tool for Host and Viral Population
Version 0.0.5
Date 2019-06-14
Author Jean-Francois Rey [aut, cre]
Maintainer Jean-Francois Rey <jean-francois.rey@inra.fr>
Description Statistical Methods for Inferring Transmissions of Infectious Diseases from deep sequencing data (SMITID).
   It allow sequence-space-time host and viral population data storage, indexation and querying.
License GPL (>= 2) | file LICENSE
LazyData true
BuildVignettes true
NeedsCompilation no
Biarch true
URL https://informatique-mia.inra.fr/biosp/anr-smitid-project,
 https://gitlab.paca.inra.fr/SMITID/structR
BugReports https://gitlab.paca.inra.fr/SMITID/structR/issues
Depends methods, utils, grDevices (>= 3.0.0), graphics (>= 3.0.0), R
(>= 3.3.0)
DependsNote BioC (>= 3.0)
Imports ggplot2, sf (>= 0.6.3), stats (>= 3.0.2), Biostrings (>= 2.0.0)
ImportsNote BioC (>= 3.0), Recommended: Biostrings
Suggests testthat (>= 2.0)
Collate 'Class-Host.R' 'Class-ViralPop.R' 'Methods-Host.R'
 'Methods-ViralPop.R' 'Methods-time.R' 'SMITIDstruct.R' 'demo.R'
 'diversity.R' 'index.R'
RoxygenNote 6.1.1
Repository CRAN
Date/Publication 2019-06-14 11:30:11 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMITIDstruct-package</td>
<td>3</td>
</tr>
<tr>
<td>addcode</td>
<td>4</td>
</tr>
<tr>
<td>addHost</td>
<td>4</td>
</tr>
<tr>
<td>addIndex</td>
<td>5</td>
</tr>
<tr>
<td>addViralObs</td>
<td>5</td>
</tr>
<tr>
<td>alleleCount</td>
<td>6</td>
</tr>
<tr>
<td>concatViralPop</td>
<td>6</td>
</tr>
<tr>
<td>createAViralPop</td>
<td>7</td>
</tr>
<tr>
<td>createHost</td>
<td>7</td>
</tr>
<tr>
<td>createIndex</td>
<td>8</td>
</tr>
<tr>
<td>demo.SMITIDstruct.run</td>
<td>8</td>
</tr>
<tr>
<td>diversity.pDistance</td>
<td>8</td>
</tr>
<tr>
<td>diversity.sfs</td>
<td>9</td>
</tr>
<tr>
<td>getCov</td>
<td>9</td>
</tr>
<tr>
<td>getDate</td>
<td>10</td>
</tr>
<tr>
<td>getDiversity.pDistance</td>
<td>10</td>
</tr>
<tr>
<td>getDiversity.sfs</td>
<td>11</td>
</tr>
<tr>
<td>getInfosByHostAndTime</td>
<td>11</td>
</tr>
<tr>
<td>getStates</td>
<td>12</td>
</tr>
<tr>
<td>getTimeLine</td>
<td>12</td>
</tr>
<tr>
<td>getTimestamp</td>
<td>13</td>
</tr>
<tr>
<td>getTransmissionTree</td>
<td>13</td>
</tr>
<tr>
<td>Host</td>
<td>14</td>
</tr>
<tr>
<td>is.juliendate</td>
<td>14</td>
</tr>
<tr>
<td>is.StringDate</td>
<td>15</td>
</tr>
<tr>
<td>is.timestamp</td>
<td>15</td>
</tr>
<tr>
<td>isInCode</td>
<td>16</td>
</tr>
<tr>
<td>loadCoords</td>
<td>16</td>
</tr>
<tr>
<td>loadCovs</td>
<td>17</td>
</tr>
<tr>
<td>loadHost</td>
<td>17</td>
</tr>
<tr>
<td>loadStates</td>
<td>18</td>
</tr>
<tr>
<td>loadTree</td>
<td>18</td>
</tr>
<tr>
<td>loadTreeDF</td>
<td>19</td>
</tr>
<tr>
<td>loadViralObs</td>
<td>20</td>
</tr>
<tr>
<td>loadViralPop</td>
<td>20</td>
</tr>
<tr>
<td>loadViralPopSet</td>
<td>21</td>
</tr>
<tr>
<td>mergeCode</td>
<td>21</td>
</tr>
<tr>
<td>plotDiversity.pDistance</td>
<td>22</td>
</tr>
<tr>
<td>plotDiversity.sfs</td>
<td>22</td>
</tr>
<tr>
<td>setStates</td>
<td>23</td>
</tr>
<tr>
<td>simulateStates</td>
<td>23</td>
</tr>
</tbody>
</table>
SMITIDstruct-package

ViralPop .................................................. 24

Index 25

SMITIDstruct-package  Data Structure and Manipulation Tool for Host and Viral Population

Description

Statistical Methods for Inferring Transmissions of Infectious Diseases from deep sequencing data (SMITID). It allow sequence-space-time host and viral population data storage, indexation and querying.

Details

Package: SMITIDstruct
Type: Package
Version: 0.0.5
Date: 2019-06-14
License: GPL (>=2)

The SMITIDstruct package contains functions and methods for manipulating Host and Viral population genotico-space-time data.

Author(s)

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

See Also
demo.SMITIDstruct.run

Examples

## Run a simulation
library("SMITIDstruct")
demo.SMITIDstruct.run()
addcode

Description
add a code event to another

Usage
addcode(code, code.add)

Arguments
code an existing code
code.add the code to add

Value
merge of the two code

addHost

Description
add an Host to a HostSet

Usage
addHost(lhost, id)

Arguments
lhost a hostSet Object
id a character of host ID

Value
a HostSet of host object with there ID

Examples
lhost <- list()
lhost <- addHost(lhost, "42")
addIndex

Description
add to an index a new eventcode

Usage
addIndex(index, id_host, time, code)

Arguments
- index: an index
- id_host: an host index in HostSet
- time: a time
- code: an event code

Value
the index updated (add a row or update one)

addViralObs

Description
load Viral pop observation in Host object

Usage
addViralObs(lhost, lvpop)

Arguments
- lhost: a HostSet
- lvpop: a ViralPopSet

Value
lhost update with viral population observed
### alleleCount

**Description**

count allele at each position

**Usage**

```r
alleleCount(mat, seq.char = c("A", "T", "G", "C"))
```

**Arguments**

- `mat`: a genomique seq list as matrix by row
- `seq.char`: allele alphabet

**Value**

a matrix, each row as a unique seq and col as allele count by position

---

### concatViralPop

**Description**

concat several Viral population in one ViralPop object

**Usage**

```r
concatViralPop(lvpop, lid)
```

**Arguments**

- `lvpop`: a ViralPop Set
- `lid`: vector of viralpop id to concat

**Value**

a ViralPop object with ID concatenation from all IDs and time at 0.
**createAViralPop**

**Description**
Create a new ViralPop object

**Usage**
```
createAViralPop(host_id, obs_time, seq, id_seq = "seq_ID", seq_value = "seq", prop = "prop", compact = FALSE)
```

**Arguments**
- `host_id`: host ID which viral pop is observed
- `obs_time`: time of the observation (numeric or date)
- `seq`: a data.frame of sequences ID, sequences and counts
- `id_seq`: column name containing the sequences ID
- `seq_value`: column name containing the sequences
- `prop`: column name containing the count of each sequences
- `compact`: boolean, default FALSE, if TRUE will try group identicals sequences (not implemented yet)

---

**createHost**

**Description**
create a list of Host class object

**Usage**
```
createHost(list_host)
```

**Arguments**
- `list_host`: a character vector of host ID

**Value**
a HostSet of host object with there ID

**Examples**
```
lh <- seq(1,30,1)
1host <- createHost(lh)
```
createIndex

**Description**
create an index of time id_host and event code

**Usage**
createIndex(hostlist)

**Arguments**
hostlist a Hostset

**Value**
a data.frame with TIME, ID_HOST and EVENTCODE as columns

demo.SMITIDstruct.run
demo.SMITIDstruct.run

**Description**
run a demo to load HostSet, ViralPopSet and index

**Usage**
demo.SMITIDstruct.run()

diversity.pDistance

**Description**
diversity calculation using Mean Pairwise Distance

**Usage**
diversity.pDistance(vpop)

**Arguments**
vpop a ViralPop object

**Value**
result
diversity.sfs

**Description**

Allele frequency spectrum or Site frequency spectra: the distribution of alternative allele frequencies across all sites of genetic sequences

**Usage**

`diversity.sfs(vpop)`

**Arguments**

`vpop` a viralPop class

**Value**

the site frequency spectra

---

getcov
getcov

**Description**

get Host(s) covariates

**Usage**

`getcov(lhost, id = NA)`

**Arguments**

`lhost` a HostSet

`id` a vector of host id (default NA: all lhost)

**Value**

a data.frame
**getDescription**

Converts a timestamp to a Date (string).

**Usage**

```
getDescription(time, format = "%Y-%m-%dT%H:%M:%S")
```

**Arguments**

- `time`: a timestamp or vector of
- `format`: Date format output (default `%Y-%m-%dT%H:%M:%S`)

**Value**

- `time` as a string date

---

**getDescription.pDistance**

Gets the pairwise distance of a host over the viral population observed.

**Usage**

```
getDescription.pDistance(host, lvpop)
```

**Arguments**

- `host`: an Host object
- `lvpop`: a ViralPopSet object

**Value**

A data.frame with columns as time of observation and `p_distance`.
getDiversity.sfs

**Description**

get Allele Frequency Spectrum or Site Frequency spectra for observed viral pop of an host

**Usage**

getDiversity.sfs(host, lvpop)

**Arguments**

- **host** an Host object
- **lvpop** an ViralPopSet object

**Value**

a list indexed by time that contains allele.time and count

getInfosByHostAndTime

**Description**

get hosts informations, status, infectedby, coordinates and time

**Usage**

getInfosByHostAndTime(index, lhost)

**Arguments**

- **index** an index
- **lhost** a hosts list

**Value**

a data.frame with colnames (id, time, infectedby, status, probabilities, X, Y)
### `getStates`

**Description**
get Host(s) states

**Usage**
```r
getStates(lhost, id = NA)
```

**Arguments**
- `lhost` a HostSet
- `id` a vector of host id (default NA : all lhost)

**Value**
a data.frame

### `getTimeLine`

**Description**
get the time line of an host

**Usage**
```r
getTimeLine(lhost, id)
```

**Arguments**
- `lhost` a hostSet
- `id` a host ID

**Value**
a data.frame
**getTimestamp**

Description

Get the timestamp of Date

Usage

```r
getTimestamp(date, format = "%Y-%m-%d %H:%M:%S")
```

Arguments

- **date**: a date (as string) or vector of
- **format**: the date format (default %Y-%m-%d %H:%M:%S)

Value

timestamp of the date(s)

---

**getTransmissionTree**

Description

get a transmission tree as a data.frame

Usage

```r
getTransmissionTree(lhost, id = NA)
```

Arguments

- **lhost**: a hostSet
- **id**: a vector of hosts ids (default NA: all host)

Value

a data.frame as source|target|time in columns

Examples

```r
path = system.file("extdata", "data-simul/", package="SMITIDstruct")
lhost <- list()
lhost <- loadTree(lhost,paste(path,"/tree.txt",sep=''))
print(getTransmissionTree(lhost))
```
### Description

Spatio-temporal information about Host.

### Details

Object can be created by calling ...  
\texttt{rdname Host-class}

### Slots

\begin{itemize}
  \item \texttt{id} Host identifier  
  \item \texttt{coordinates} Host coordinates in time (as sf)  
  \item \texttt{states} Host States/Status (dob, Inf...)  
  \item \texttt{sources} data.frame of time and host id who infected this host  
  \item \texttt{offsprings} data.frame of time and host id who has been contaminated by this host  
  \item \texttt{ID_V_POP} data.frame of time and index of Viral population Observation  
  \item \texttt{covariates} data.frame of time, cavariate and value of this host.
\end{itemize}

### is.juliendate

\textbf{Description}

Check if a numeric is not a timestamp

\textbf{Usage}

\texttt{is.juliendate(time)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{time} a numeric
\end{itemize}

\textbf{Value}

TRUE if time is a julien day, otherwise FALSE
is.StringDate

Description

Check if a string represent a date

Usage

is.StringDate(date)

Arguments

date a string or a vector of string (without NA)

Value

TRUE if date contains date format

is.timestamp

Description

Check if a numeric represent a timestamp

Usage

is.timestamp(time)

Arguments

time a numeric

Value

TRUE if time >= 1971
**isInCode**

**Description**
check a code contains a specific code

**Usage**
isInCode(code, thecode)

**Arguments**
- code: list of code to test
- thecode: the real code

**Value**
TRUE if code contain thecode otherwise FALSE

---

**loadCoords**

**Description**
Load Hosts states

**Usage**
loadCoords(lhost, dfCoords, id = "ID")

**Arguments**
- lhost: a HostSet
- dfCoords: a data.frame with host ID, time and longitude latitude values
- id: colname for host ID

**Value**
lhost updated

**Examples**
```r
path = system.file("extdata", "data-simul/", package="SMITIDstruct")
lhost <- list()
lhost <- loadTree(lhost, paste(path,"/tree.txt",sep=''))
coords <- read.table(file=paste(path,"/hosts_coords.txt",sep=''), header=TRUE, check.names=FALSE)
lhost <- loadCoords(lhost,coords)
```
### loadCovs

**Description**

Load Hosts covariates

**Usage**

```r
loadCovs(lhost, dfCovs, id = "ID", colCovs)
```

**Arguments**

- `lhost`: a HostSet
- `dfCovs`: a data.frame with host ID in rows and covariates in columns
- `id`: colname for host ID
- `colCovs`: colnames of covariates columns

**Value**

`lhost` updated with covariates

### loadHost

**Description**

load host object from a file

**Usage**

```r
loadHost(file = "host.txt")
```

**Arguments**

- `file`: a file containing hosts data

**Value**

a list of Host object (HostSet) include Class-Host.R
loadStates

Description
Load Hosts states

Usage
loadStates(lhost, dfStates, id = "ID", colStates)

Arguments
- lhost: a HostSet
- dfStates: a data.frame with host ID and states in columns and time as value
- id: colname for host ID
- colStates: colnames of States columns

Value
lhost updated

Examples
path = system.file("extdata", "data-simul/", package="SMITIDstruct")
lhost <- list()
class(lhost) <- "hostSet"
lhost <- loadTree(lhost, paste(path,"/tree.txt",sep=''))
obs <- read.table(paste(path,"/obs.txt",sep=''),header=TRUE, check.names=FALSE)
obs.states <- c(colnames(obs[-grep("ID|Tobs.x",colnames(obs))]))
lhost <- loadStates(lhost, obs, colStates=obs.states)

loadTree

Description
load sources and offsprings from file

Usage
loadTree(lhost = list(), file = "tree.txt", source = "ID-source", receptor = "ID-receptor", tinf = "Tinf", weight = "Weight")
loadTreeDF

Arguments

lhost  a HostSet
file  a file containing tree data
source  column name for source ID
receptor  column name for receptor ID
tinf  column name for infection Time
weight  column name of infection weight

Value

the lhost param update with sources and offsprings

Examples

path = system.file("extdata", "data-simul/", package="SMTPDirect")
lhost <- list()
class(lhost) <- "hostSet"
lhost <- loadTree(lhost, paste(path,"/tree.txt",sep=''))

Description

load sources and offsprings from a data.frame

Usage

loadTreeDF(lhost = list(), df = data.frame(), source = "ID-source", receptor = "ID-receptor", tinf = "Tinf", weight = "Weight")

Arguments

lhost  a HostSet
df  a data.frame containing tree data
source  column name for source ID
receptor  column name for receptor ID
tinf  column name for infection Time
weight  infection links probability

Value

the lhost param update with sources and offsprings
loadViralObs

Description

load a ViralPop object

Usage

loadViralObs(id, time, file)

Arguments

id host pathogen ID
time time of the observation (numeric or Date)
file a fasta file

Value

a new ViralPop object

loadViralPop

Description

Load all ViralPop observated in the file.obs

Usage

loadViralPop(directory, listFiles, listCol = list(id = "id", timeObs = "time", filename = "filename"), file.extension = "fasta")

Arguments

directory path where is data
listFiles a dataframe with host ID, time observation and file name (filename.fasta)
listCol a list of listFiles columns names ("id", "timeObs", "filename")
file.extension genotype file extension

Value

a vector of ViralPop object
Examples

```r
path = system.file("extdata", "data-simul/", package="SMITIDstruct")
files <- list.files(path, pattern = ".*.fasta", full.names=FALSE)
lfinfo <- sapply(files, function(x) return(substr(x, 1, nchar(x) - 6))))
splitFiles <- strsplit(lfinfoinfo, ";")
listF <- cbind(data.frame(matrix(unlist(splitFiles), nrow=length(splitFiles), byrow=TRUE),
stringsAsFactors = FALSE), names(splitFiles))

colnames(listF) <- c("id", "time", "filename")

lvpop <- loadviralPop(path, listF)
```

Description

load a list of viral populations

Usage

```r
loadviralpopset(lvpop = list(), list)
```

Arguments

- `lvpop`: a viralPopSet (default new one)
- `list`: a list (see details)

Details

The list have to be on this format: list$HOST_ID$TIME$seq_id$seq$prop A list indexed by host ID, follow by a list indexed by time (of observation). The last list contains an array of seq_ID (sequence ID), an array of seq (sequence as characters), and an array of the count of seq. example: `S'HOST_42'S'2014-01-01T00:00:00$seq_ID ["SEQ_1","SEQ_2"] S'HOST_42'S'2014-01-01T00:00:00$seq ["ACGT","TGCA"] S'HOST_42'S'2014-01-01T00:00:00$seq_ID ["46","6"]`

mergeCode

Description

merge a list of event code

Usage

```r
mergeCode(listcode)
```
plotDiversity.sfs

Arguments

  listcode  a list of event code*

Value

  a code

---

plotDiversity.pDistance  

Description

plot Mean Pairwise Distance for an host viralpop over time

Usage

  plotDiversity.pDistance(host, lvpop)

Arguments

  host  an Host object
  lvpop  a ViralPopSet object

---

plotDiversity.sfs  

Description

plot Allele frequency spectum for an host viralpop over time

Usage

  plotDiversity.sfs(host, lvpop)

Arguments

  host  an Host object
  lvpop  an ViralPopSet object
**setStates**

**Description**

set hosts states from a data.frame

**Usage**

```r
setStates(lhost, dfStates, colStates = c(id = "ID", time = "time", states = "value"))
```

**Arguments**

- `lhost`: a HostSet
- `dfStates`: a data.frame with host ID and states and time in columns
- `colStates`: vector of the columns name, id, time and states

**Value**

the HostSet updated

---

**simulateStates**

**Description**

simulate states from sources infection

**Usage**

```r
simulateStates(lhost)
```

**Arguments**

- `lhost`: a HostSet

**Value**

lhost update with states from sources time ~
### Class ViralPop

**Description**

Viral population data containing genotypes

**Slots**

- **ID**: Host identifier
- **time**: Observation time as numeric since 1970/01/01
- **size**: Qt of variants
- **names**: list of variants id with same sequence
- **genotypes**: all variants genotypes (as DNAStringSet)
- **proportions**: proportions of each variants
Index

_PACKAGE (SMITIDstruct-package), 3

addcode, 4
addHost, 4
addIndex, 5
addViralObs, 5
alleleCount, 6

concatViralPop, 6
createAViralPop, 7
createHost, 7
createIndex, 8
demo.SMITIDstruct.run, 3, 8
diversity.pDistance, 8
diversity.sfs, 9
getCov, 9
date, 10
diversity.pDistance, 10
diversity.sfs, 11
getInfosByHostAndTime, 11
getState, 12
getTimeLine, 12
getTimestamp, 13
getTransmissionTree, 13

Host, 14
Host-class (Host), 14

is.juliendate, 14
is.StringDate, 15
is.timestamp, 15
isInCode, 16

loadCoords, 16
loadCovs, 17
loadHost, 17
loadStates, 18
loadTree, 18
loadTreeDF, 19

loadViralObs, 20
loadViralPop, 20
loadViralPopSet, 21
mergeCode, 21
plotDiversity.pDistance, 22
plotDiversity.sfs, 22
setStates, 23
simulateStates, 23
SMITIDstruct (SMITIDstruct-package), 3
SMITIDstruct-package, 3

ViralPop, 24
ViralPop-class (ViralPop), 24