

# Package ‘SMITIDstruct’

March 5, 2019

**Type** Package

**Encoding** UTF-8

**Title** Data Structure and Manipulations Tool for Host and Viral Population

**Version** 0.0.4

**Date** 2019-03-05

**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** 6.1.0

**Repository** CRAN

**Date/Publication** 2019-03-05 13:20:03 UTC

## R topics documented:

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

---

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: 0.0.4  
Date: 2019-02-28  
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	<i>addcode</i>
---------	----------------

---

**Description**

add a code event to an another

**Usage**

```
addcode(code, code.add)
```

**Arguments**

code	an existing code
code.add	the code to add

**Value**

merge of the two code

---

addHost	<i>addHost</i>
---------	----------------

---

**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	<i>addIndex</i>
----------	-----------------

---

**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	<i>addViralObs</i>
-------------	--------------------

---

**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

*alleleCount*


---

**Description**

count allele at each position

**Usage**

```
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

*concatViralPop*


---

**Description**

concat several Viral population in one ViralPop object

**Usage**

```
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	<i>createAViralPop</i>
-----------------	------------------------

---

**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	<i>createHost</i>
------------	-------------------

---

**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)
lhost <- createHost(lh)
```

---

createIndex	<i>createIndex</i>
-------------	--------------------

---

**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	<i>demo.SMITIDstruct.run</i>
-----------------------	------------------------------

---

**Description**

run a demo to load HostSet, ViralPopSet and index

**Usage**

```
demo.SMITIDstruct.run()
```

---

diversity.pDistance	<i>diversity.pDistance</i>
---------------------	----------------------------

---

**Description**

diversity calculation using Mean Pairwise Distance

**Usage**

```
diversity.pDistance(vpop)
```

**Arguments**

vpop	a ViralPop object
------	-------------------

**Value**

result



---

diversity.sfs	<i>diversity.sfs</i>
---------------	----------------------

---

**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	<i>getCov</i>
--------	---------------

---

**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

---

<code>getDate</code>	<i>getDate</i>
----------------------	----------------

---

**Description**

Converte timestamp to Date (string)

**Usage**

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

**Arguments**

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

**Value**

time as string date

---

<code>getDiversity.pDistance</code>	<i>getDiversity.pDistance</i>
-------------------------------------	-------------------------------

---

**Description**

get pairwise distance of an host over viral population observed

**Usage**

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

**Arguments**

<code>host</code>	an Host object
<code>lvpop</code>	a ViralPopSet object

**Value**

a data.frame with col as time of observation and `p_distance`

---

*getDiversity.sfs*      *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

---

*getStates*            *getStates*

---

**Description**

get Host(s) states

**Usage**

`getStates(lhost, id = NA)`

**Arguments**

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

**Value**

a data.frame

---

`getTimeLine`*getTimeLine*

---

**Description**

get the time line of an host

**Usage**

```
getTimeLine(lhost, id)
```

**Arguments**

lhost            a hostSet

id                a host ID

**Value**

a data.frame

---

`getTimestamp`*getTimestamp*

---

**Description**

Get the timestamp of Date

**Usage**

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

**Arguments**

date            a date (as string) or vector of

format          the date format (default %Y-%m-%dT%H:%M:%S)

**Value**

timestamp of the date(s)

---

getTransmissionTree    *getTransmissionTree*

---

**Description**

get a transmission tree as a data.frame

**Usage**

```
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**

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

---

Host                            *Class Host*

---

**Description**

Spatio-temporal information about Host.

**Details**

Object can be created by calling ...

rdname Host-class

**Slots**

ID Host identifier  
 coordinates Host coordinates in time (as sf)  
 states Host States/Status (dob, Inf...)  
 sources data.frame of time and host id who infected this host  
 offsprings data.frame of time and host id who has been contaminated by this host  
 ID\_V\_POP data.frame of time and index of Viral population Observation  
 covariates data.frame of time, cavariate and value of this host.

---

<code>is.juliendate</code>	<i>is.juliendate</i>
----------------------------	----------------------

---

**Description**

Check if a numeric is not a timestamp

**Usage**

```
is.juliendate(time)
```

**Arguments**

time            a numeric

**Value**

TRUE if time is a julien day, otherwise FALSE

---

<code>is.StringDate</code>	<i>is.StringDate</i>
----------------------------	----------------------

---

**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	<i>is.timestamp</i>
--------------	---------------------

---

**Description**

Check if a numeric represent a timestamp

**Usage**

```
is.timestamp(time)
```

**Arguments**

time            a numeric

**Value**

TRUE if time >= 1971

---

loadCoords	<i>loadCoords</i>
------------	-------------------

---

**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**

```
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	<i>loadCovs</i>
----------	-----------------

---

**Description**

Load Hosts covariates

**Usage**

```
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	<i>loadHost</i>
----------	-----------------

---

**Description**

load host object from a file

**Usage**

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

**Arguments**

file	a file containing hosts data
------	------------------------------

**Value**

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



---

loadStates	<i>loadStates</i>
------------	-------------------

---

**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.*", colnames(obs))]))
lhost <- loadStates(lhost, obs, colStates=obs.states)
```

---

loadTree	<i>loadTree</i>
----------	-----------------

---

**Description**

load sources and offsprings from file

**Usage**

```
loadTree(lhost = list(), file = "tree.txt", source = "ID-source",
receptor = "ID-receptor", tinf = "Tinf")
```

**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

**Value**

the lhost param update with sources and offsprings

**Examples**

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

---

loadTreeDF

*loadTreeDF*


---

**Description**

load sources and offsprings from file

**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	epidemiological links probability

**Value**

the lhost param update with sources and offsprings

---

loadViralObs	<i>loadViralObs</i>
--------------	---------------------

---

**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	<i>loadViralPop</i>
--------------	---------------------

---

**Description**

Load all ViralPop observed 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**

```

path = system.file("extdata", "data-simul/", package="SMITIDstruct")
files <- list.files(path, pattern = ".*.fasta" ,full.names=FALSE)
lfileinfo <- sapply(files,function(x){return(substr(x,1,nchar(x)-6))})
splitFiles <- strsplit(lfileinfo, "_");
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)

```

---

loadViralPopSet	<i>loadViralPopSet</i>
-----------------	------------------------

---

**Description**

load a list of viral populations

**Usage**

```
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\$list\$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 : \$'HOST\_42'\$'2014-01-01T00:00:00'\$seq\_ID ["SEQ\_1","SEQ\_2"] \$'HOST\_42'\$'2014-01-01T00:00:00'\$seq ["ACGT","TGCA"] \$'HOST\_42'\$'2014-01-01T00:00:00'\$seq\_ID ["46","6"]

---

mergeCode	<i>mergeCode</i>
-----------	------------------

---

**Description**

merge a list of event code

**Usage**

```
mergeCode(listcode)
```

**Arguments**

listcode          a list of event code\*

**Value**

a code

---

plotDiversity.pDistance  
*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      *plotDiversity.sfs*

---

**Description**

plot Allele frequency spetrum for an host viralpop over time

**Usage**

plotDiversity.sfs(host, lvpop)

**Arguments**

host              an Host object  
lvpop             an ViralPopSet object

---

 setStates

*setStates*


---

**Description**

set hosts states from a data.frame

**Usage**

```
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

*simulateStates*


---

**Description**

simulate states from sources infection

**Usage**

```
simulateStates(lhost)
```

**Arguments**

lhost            a HostSet

**Value**

lhost update with states from sources time ~

---

ViralPop

*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 DNABStringSet)

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  
`getDate`, 10  
`getDiversity.pDistance`, 10  
`getDiversity.sfs`, 11  
`getStates`, 11  
`getTimeLine`, 12  
`getTimestamp`, 12  
`getTransmissionTree`, 13

`Host`, 13  
`Host-class` (Host), 13

`is.juliendate`, 14  
`is.StringDate`, 14  
`is.timestamp`, 15

`loadCoords`, 15  
`loadCovs`, 16  
`loadHost`, 16  
`loadStates`, 17  
`loadTree`, 17  
`loadTreeDF`, 18  
`loadViralObs`, 19  
`loadViralPop`, 19

`loadViralPopSet`, 20

`mergeCode`, 20

`plotDiversity.pDistance`, 21  
`plotDiversity.sfs`, 21

`setStates`, 22  
`simulateStates`, 22  
`SMITIDstruct` (SMITIDstruct-package), 3  
`SMITIDstruct-package`, 3

`ViralPop`, 23  
`ViralPop-class` (ViralPop), 23