Package: ibmcraftr (via r-universe)

September 7, 2024

Type Package

Title Toolkits to Develop Individual-Based Models in Infectious Disease

Version 1.0.0

Date 2016-11-16

Description It provides a generic set of tools for initializing a synthetic population with each individual in specific disease states, and making transitions between those disease states according to the rates calculated on each timestep. The new version 1.0.0 has C++ code integration to make the functions run faster. It has also a higher level function to actually run the transitions for the number of timesteps that users specify. Additional functions will follow for changing attributes on demographic, health belief and movement.

License MIT + file LICENSE

LazyData TRUE

RoxygenNote 5.0.1

Suggests testthat

LinkingTo Rcpp

Imports Rcpp

Repository https://saitheinthantun.r-universe.dev

RemoteUrl https://github.com/saitheinthantun/ibmcraftr

RemoteRef HEAD

RemoteSha f71715fe9207cfa063f3194f66141f7ac7d4ebaa

Contents

`cumprob` *Calculate cumulative probabilities for state transitions.*

Description

This function takes in a vector of probabilities of states transitions and calculate the probability of staying in the original state and output the cumulative probabilities for all possibilities.

Usage

```
cumprob(probs, actual = FALSE)
```

Arguments

`probs`	A numeric vector of the probabilities of transition to states.
`actual`	A logical value, if TRUE, will calculate actual cumulative probabilities which may surpass 1!.

Value

A numeric vector of cumulative probabilites inclusive of the probability of having the same state in the next timestep.

Examples

```
cumprob(c(.2,.2,.9))
cumprob(c(.2,.2,.9), actual=TRUE)
cumprob(c(.2,.2,.2))
```

`rate2prob` *Miscellaneous functions to support the* `ibmcraftr` *packare are here.*

Description

Miscellaneous functions to support the `ibmcraftr` packare are here.

Usage

```
rate2prob(rates)
```

Arguments

`rates`	A numeric scalar or vector to be transformed into rates.

Value

A numeric scalar or vector in terms of probabilities.

Examples

```
rate2prob(c(.1, .5))
```

`run_state_trans` *Run state_trans function over a given number of timesteps.*

Description

Organize population data and transition parameters to run state_trans function over the given number of timesteps.

Usage

```
run_state_trans(timesteps, param, pop, transient = "", useC = TRUE)
```

Arguments

`timesteps`	A numeric scalar based on which the state_trans function will run for that specific no. of `timesteps` and accumulate the results.
`param`	A list of lists. Each low-level list must contain transition parameters required by the `state_trans` function.
`pop`	A state matrix created from `syn_pop` function. This matrix represents the states of the population.
`transient`	A character vector. Each element must include formula(e)/expression(s) to evaluate dynamic parameters after each timestep.
`useC`	A logical value, which is TRUE by default, will run `state_transition` function written in RCPP, `stRCPP`.

Value

A summary matrix of the states all individuals in the population are in.

Examples

```
pop <- syn_pop(c(19,1,0,0,0)) #synthesizing population
b <- 2 #effective contact rate
param <- list(
list(1,c(2,5),c(NA,.1)), #transition from state 1 to 2 using FOI lambda
list(2,3,100), #transition from state 2 to 3,
list(3,4,100)  #the 3rd term ensures the transition to the next stage
)
```

```
timesteps <- 10
transient <- c("param[[1]][[3]][1] <- rate2prob(b*sum(pop[,2],pop[,3])/sum(pop))")
eval(parse(text=transient))

run_state_trans(timesteps, param, pop, transient)
run_state_trans(timesteps, param, pop, transient, useC = FALSE)
```

`state_trans` *Make state transitions.*

Description

Take in the matrix of the states of synthetic population (created by `syn_pop` function) and calculate the transitions from one state to other state(s) using the transition rate(s).

Usage

```
state_trans(origin, new.states, params, s.matrix)
```

Arguments

`origin`	A number which represents the column index `s.matrix` you want to do the transition from
`new.states`	A numeric vector or a number which represents the column index `s.matrix` you want as the destination(s) for the transition
`params`	A numeric vector of similar length to `new.states` which serves as the transition rate(s)
`s.matrix`	A state matrix created from `syn_pop` function

Value

A transition matrix of the same dimension as `s.matrix`. -1 indicates that the individual has left the corresponding state. +1 indicates that the individual has become the corresponding state.

Examples

```
pop <- syn_pop(c(19,1,0,0))
state_trans(1,2,.1,pop)
state_trans(1,4,100,pop)
```

stRCPP *Make state transitions using Rcpp.*

Description

Take in the matrix of the states of synthetic population (created by syn_pop function) and calculate the transitions from one state to other state(s) using the transition probabilities [not rate(s)]. The major difference from the R alone version was that instead of having the transition rate(s), transition probabilities are used. These probabilities will thus be calculated with another function.

Usage

```
stRCPP(origin, new.states, params, s.matrix)
```

Arguments

origin	A number which represents the column index s.matrix you want to do the transition from
new.states	A numeric vector or a number which represents the column index s.matrix you want as the destination(s) for the transition
params	A numeric vector of similar length to new.states which serves as the transition rate(s)
s.matrix	A state matrix created from syn_pop function

Value

A transition matrix of the same dimension as s.matrix. -1 indicates that the individual has left the corresponding state. +1 indicates that the individual has become the corresponding state.

Examples

```
pop <- syn_pop(c(19,1,0,0))
stRCPP(1,2,.1,pop)
```

syn_pop *Create a synthetic population having several states.*

Description

Populate a matrix in which columns represent the states of the individuals and rows represent the individuals.

Usage

```
syn_pop(states, shuffle = FALSE)
```

Arguments

states	A numeric vector with each element representing the number of individuals in a particular state its index corresponds to.
shuffle	A logical value to enable shuffling of the individuals (rows) in the resulting matrix.

Value

A matrix of 0s, and 1s. The rows representing the individuals and the columns representing the states the individuals are in

Examples

```
syn_pop(c(3,2,1))
syn_pop(c(0,0,1,5), shuffle=TRUE)
```

Index