Package 'BMLgrid'

May 17, 2015

Type Package

Title Bihman-Middleton-Levin Traffic Model

Version 1.0

Date 2015-04-30

Author Yin Zhang

Maintainer Yin Zhang <yinzh@ucdavis.edu>

Description This is a package for create and simulate Biham-Middleton-Levine(BML) Traffic Model.

License GPL-2

Suggests testthat

R topics documented:

BMLgrid-package 1
createBMLGrid 2
crunBMLGrid 3
move 3
moveslow 4
plot.BML 5
print.summary.BML 5
runBMLGrid 6
summary.BML 6

Index **8**

`BMLgrid-package` *Bihman-Middleton-Levin Traffic Model*

Description

This is a package for create and simulate Biham-Middleton-Levine(BML) Traffic Model.

Details

Package:	BMLgrid
Type:	Package
Version:	1.0
Date:	2015-04-30
License:	GPL-2

There are two important function, creatBMLGrid() to creat a grid and runBMLGrid() to run a grid several steps

Author(s)

Yin Zhang

Maintainer: Yin Zhang <yinzh@ucdavis.edu>

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))

g_out=runBMLGrid(g, numSteps = 10000)
```

`createBMLGrid` *Create a BMLGrid*

Description

Creat a grid with blue cars and red cars

Usage

```
createBMLGrid(r, c, ncars, prop = 0.5)
```

Arguments

`r`	Integer. The number of row of the grid
`c`	Integer. The number of column of the grid
`ncars`	Two dimentional vector or numeric. The number of cars, it should be like c(red = 10, blue = 10), or a proportion of the cars in the grid which should be between 0 and 1
`prop`	Numeric. The proportion of red:blue cars. The default value is 0.5

Value

It returns a grid with red and blue cars.

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))
```

crunBMLGrid *Run a BML grid in steps in C*

Description

It can run a BML grid in certain steps.

Usage

```
crunBMLGrid(grid, numSteps)
```

Arguments

grid	BML grid, it is the initial status.
numSteps	interger, the number of steps to run the cars.

Value

grid	The final grid after the number of steps
vbystep	The velosity for every steps
movestep	The number of cars to move in every steps

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))
g.out=crunBMLGrid(g, numSteps = 10000)
```

move *Move a grid at a exact time step*

Description

move gives a result of a grid which move one step. The odd step should be move blue cars and the even step should be move red cars

Usage

```
move(grid, time)
```

Arguments

grid	BML grid, the initial grid should be move
time	interger, the time step to move the grid

Value

The result is a LIST,

grid	The final grid after one step
v	the velosity, moved cars/number of corresponding cars
carmove	number of moved car

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))

move(g, 5) # move g at 5 time step
```

---

moveslow *Move a grid in a slow way*

Description

moveslow gives a result of a grid which move one step. The odd step should be move blue cars and the even step should be move red cars

Usage

```
moveslow(grid, time)
```

Arguments

grid	BML grid, the initial grid should be move
time	interger, the time step to move the grid

Value

The result is a LIST,

grid	The final grid after one step
v	the velosity, moved cars/number of corresponding cars
carmove	number of moved car

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))
moveslow(g, 5) # move g at 5 time step
```

plot.BML *Plot a BML grid*

Description

It will plot a BML grid.

Usage

```
plot.BML(x, main = "BML plot", ...)
```

Arguments

x	The BML grid to plot.
main	The name of the plot. The default value is "BML plot"
...	Arguments to other features of the plot

Value

it is a plot

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))

plot(g)
```

print.summary.BML *Print the summary for BML grid*

Description

It will give a summary of initial BLM grid and the status of after run several time steps.

Usage

```
print.summary.BML(x, ...)
```

Arguments

x	The result of summary.grid
...	Additional arguments affecting the summary produced.

Value

It will give a plot of intial grid and final grid and a LIST,

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))

summary(g, numSteps = 1000)
```

runBMLGrid *Run a BML grid in steps in R*

Description

It can run a BML grid in certain steps.

Usage

```
runBMLGrid(g, numSteps, slow = FALSE)
```

Arguments

g	BML grid, it is the initial status.
numSteps	Interger, the number of steps to run the cars.
slow	Logical, means whether use the "moveslow"" function, the default value is FALSE, means not use the "moveslow"" function but "move" function

Value

It is a LIST,

grid	The final grid after the number of steps
vbystep	The velosity for every steps
movestep	The number of cars to move in every steps

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))

g.out=runBMLGrid(g, numSteps = 10000)
```

summary.BML *Summary for BML grid*

Description

It will give a summary of initial BLM grid and the status of after run several time steps.

Usage

```
summary.BML(object, numSteps, ...)
```

Arguments

object	BML grid, the initial grid
numSteps	Interger, the steps to run the initial grid
...	Additional arguments affecting the summary produced

Value

It is a LIST,

`initial`	The initial grid
`final`	The final grid
`prop`	The proportion of cars in the grid
`num_red`	The number of red cars
`num_blue`	The number of blue cars
`ave_velocity`	The average of velocity in the number of steps
`car_move`	The average number of moved cars
`block`	The average of blocked cars

Examples

```
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100,blue = 100))

summary(g, numSteps = 1000)
```

Index

∗Topic **\textasciitildekwd1**
createBMLGrid, 2
crunBMLGrid, 3
move, 3
moveslow, 4
plot.BML, 5
print.summary.BML, 5
runBMLGrid, 6
summary.BML, 6
∗Topic **\textasciitildekwd2**
createBMLGrid, 2
crunBMLGrid, 3
move, 3
moveslow, 4
plot.BML, 5
print.summary.BML, 5
runBMLGrid, 6
summary.BML, 6
∗Topic **package**
BMLgrid-package, 1

BMLgrid *(*BMLgrid-package*)*, 1
BMLgrid-package, 1

createBMLGrid, 2
crunBMLGrid, 3

move, 3
moveslow, 4

plot.BML, 5
print.summary.BML, 5

runBMLGrid, 6

summary.BML, 6