

# Package ‘BMLgrid’

April 30, 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 Bihman-Middleton-Levin(BML) Traffic Model.

**License** GPL-2

**Suggests** testthat

## R topics documented:

BMLgrid-package . . . . .	1
createBMLGrid . . . . .	2
move . . . . .	3
plot.BML . . . . .	3
print.summary.BML . . . . .	4
runBMLGrid . . . . .	4
summary.BML . . . . .	5
<b>Index</b>	<b>6</b>

---

BMLgrid-package	<i>Bihman-Middleton-Levin Traffic Model</i>
-----------------	---

---

## Description

This is a package for create and simulate Bihman-Middleton-Levin(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	<i>Create a BMLGrid</i>
---------------	-------------------------

---

**Description**

Creat a grid with blue cars and red cars

**Usage**

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

**Arguments**

<code>r</code>	Integer. The number of row of the grid
<code>c</code>	Integer. The number of column of the grid
<code>ncars</code>	Two dimentional vector or numeric. The number of cars, it should be like <code>c(red = 10, blue = 10)</code> , or a proportion of the cars in the grid which should be between 0 and 1
<code>prop</code>	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))
```

---

move	<i>Move a grid at a exact time step</i>
------	---

---

**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 velocity, 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
```

---

plot.BML	<i>Plot a BML grid</i>
----------	------------------------

---

**Description**

It will plot a BML grid.

**Usage**

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

**Arguments**

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

---

**Description**

It can run a BML grid in certain steps.

**Usage**

```
runBMLGrid(g, numSteps)
```

**Arguments**

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

**Value**

It is a LIST, use

grid	The final grid after the number of steps
vbystep	The velocity 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	<i>Summary for BML grid</i>
-------------	-----------------------------

---

**Description**

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

**Usage**

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

**Arguments**

grid	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 \textasciitildekw1

- createBMLGrid, [2](#)
- move, [3](#)
- plot.BML, [3](#)
- print.summary.BML, [4](#)
- runBMLGrid, [4](#)
- summary.BML, [5](#)

## \*Topic \textasciitildekw2

- createBMLGrid, [2](#)
- move, [3](#)
- plot.BML, [3](#)
- print.summary.BML, [4](#)
- runBMLGrid, [4](#)
- summary.BML, [5](#)

## \*Topic **package**

- BMLgrid-package, [1](#)

BMLgrid (BMLgrid-package), [1](#)

BMLgrid-package, [1](#)

createBMLGrid, [2](#)

move, [3](#)

plot.BML, [3](#)

print.summary.BML, [4](#)

runBMLGrid, [4](#)

summary.BML, [5](#)