

# Package ‘spdynmod’

February 4, 2022

**Type** Package

**Title** Spatio-Dynamic Wetland Plant Communities Model

**Description** A spatio-dynamic modelling package that focuses on three characteristic wetland plant communities in a semiarid Mediterranean wetland in response to hydrological pressures from the catchment. The package includes the data on watershed hydrological pressure and the initial raster maps of plant communities but also allows for random initial distribution of plant communities. For more detailed info see: Martinez-Lopez et al. (2015) <[doi:10.1016/j.ecolmodel.2014.11.024](https://doi.org/10.1016/j.ecolmodel.2014.11.024)>.

**Version** 1.1.6

**Date** 2022-01-04

**Author** Javier Martinez-Lopez <[javi.martinez.lopez@gmail.com](mailto:javi.martinez.lopez@gmail.com)>, Babak Naimi <[naimi.b@gmail.com](mailto:naimi.b@gmail.com)>, Julia Martinez-Fernandez <[juliamf@um.es](mailto:juliamf@um.es)>

**Maintainer** Javier Martinez-Lopez <[javi.martinez.lopez@gmail.com](mailto:javi.martinez.lopez@gmail.com)>

**License** GPL (>= 2)

**URL** <https://github.com/javimarlop/spdynmod>

**BugReports** <https://github.com/javimarlop/spdynmod/issues>

**LazyLoad** yes

**VignetteBuilder** knitr

**Imports** deSolve, animation

**Depends** raster, sp

**Suggests** knitr, roxygen2, igraph, testthat

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-02-04 22:20:02 UTC

**R topics documented:**

animate_bs . . . . .	2
animate_model . . . . .	2
animate_rb . . . . .	3
animate_sm . . . . .	3
animate_ss . . . . .	4
mrgf . . . . .	4
neigh_cell . . . . .	5
plot_maps . . . . .	5
run_model . . . . .	6
spdynmod . . . . .	6
<b>Index</b>	<b>8</b>

---

animate_bs	<i>Make an animation of the model output in relation to bare soil.</i>
------------	--

---

**Description**

It requires ImageMagick installed in the system to work.

**Usage**

```
animate_bs()
```

**Value**

a GIF animated file

**Examples**

```
## Not run animate_bs()
```

---

animate_model	<i>Make an animation of the model output containing all state variables.</i>
---------------	--

---

**Description**

It requires ImageMagick installed in the system to work.

**Usage**

```
animate_model()
```

**Value**

a GIF animated file

**Examples**

```
## Not run animate_model()
```

---

animate_rb	<i>Make an animation of the model output in relation to reed beds.</i>
------------	--

---

**Description**

It requires ImageMagick installed in the system to work.

**Usage**

```
animate_rb()
```

**Value**

a GIF animated file

**Examples**

```
## Not run animate_rb()
```

---

animate_sm	<i>Make an animation of the model output in relation to the salt marsh community.</i>
------------	---

---

**Description**

It requires ImageMagick installed in the system to work.

**Usage**

```
animate_sm()
```

**Value**

a GIF animated file

**Examples**

```
## Not run animate_sm()
```

---

animate_ss	<i>Make an animation of the model output in relation to the salt steppe community.</i>
------------	--

---

**Description**

It requires ImageMagick installed in the system to work.

**Usage**

```
animate_ss()
```

**Value**

a GIF animated file

**Examples**

```
## Not run animate_ss()
```

---

mrgf	<i>A function to perform Multiple Resolution Goodness of Fit.</i>
------	---

---

**Description**

Returns the results of a Multiple Resolution Goodness of Fit after the modified method of Kuhnert et al. 2005, originally by Costanza 1989. This function is computationally intensive and consumes a lot of RAM memory.

**Usage**

```
mrgf(year = "1992", w1 = 1, w2 = 27, k = 0)
```

**Arguments**

year	year validation year (by default 1992)
w1	w1 initial window size (by default 1)
w2	w2 final window size (by default 27; max. 113)
k	k parameter for weighting Ft with lower/larger windows resolutions

**Value**

Fw vector of fits for each window size  
 Ft weighted overall fit

**Examples**

```
## Not run mrgf(year='1992',w1=1,w2=113,k=0)
```

---

neigh_cell	<i>A function to identify neighboring cells.</i>
------------	--

---

**Description**

Returns cell numbers of neighboring pixels to a reference cell

**Usage**

```
neigh_cell(cell)
```

**Arguments**

cell                    cell number from which to compute neighbor cells

**Value**

returns cell numbers of neighboring pixels to the reference one

**Examples**

```
## Not run neigh_cell(n)
```

---

plot_maps	<i>Plot abundance maps of plant communities.</i>
-----------	--

---

**Description**

Plot abundance maps of plant communities in a given year.

**Usage**

```
plot_maps(year = 2008)
```

**Arguments**

year                    year to plot (from 1984 to 2008)

**Value**

by default plots the final plant communities map (year = 2008).

**Examples**

```
## Not run plot_maps()
```

---

run_model	<i>Run the model.</i>
-----------	-----------------------

---

### Description

Model solving function using ode.2D from the 'deSolve' package.

### Usage

```
run_model(pgr_rb = 0.005, pgr_sm = 0.2, rnd = FALSE, method = "euler",
          TS = 0.25)
```

### Arguments

pgr_rb	potential growth rate of reed beds
pgr_sm	potential growth rate of salt marsh
rnd	create random initial state variables map. It is calculated based on a script adapted from Murray Efford (University of Auckland, New Zealand) and Santiago Saura (Universidad Politecnica de Madrid, Spain). This option requires that the igraph package is available.
method	integration method: "lsodes", "euler", "rk4", "ode23", "ode45", "adams", "iteration"
TS	time step

### Value

the function outputs a matrix named "out" which contains the model simulated values for every pixel, time step and state variable.

### Examples

```
## Not run run_model()
```

---

spdynmod	<i>spdynmod model</i>
----------	-----------------------

---

### Description

spdynmod model  
The wetland model function.

### Usage

```
spdynmod(t, init, parameters, nr, nc)
```

**Arguments**

t	time
init	init
parameters	model parameters
nr	number of raster map rows
nc	number of raster map columns

**Value**

Model solver

**Examples**

```
## Not run spdynmod()
```

# Index

- \* **Fit**
  - [mrgf](#), [4](#)
- \* **Goodness**
  - [mrgf](#), [4](#)
- \* **Multiple**
  - [mrgf](#), [4](#)
- \* **Resolution**
  - [mrgf](#), [4](#)
- \* **animation**
  - [animate\\_bs](#), [2](#)
  - [animate\\_model](#), [2](#)
  - [animate\\_rb](#), [3](#)
  - [animate\\_sm](#), [3](#)
  - [animate\\_ss](#), [4](#)
- \* **keywords**
  - [run\\_model](#), [6](#)
- \* **neighborhood**
  - [neigh\\_cell](#), [5](#)
- \* **plot**
  - [plot\\_maps](#), [5](#)
- \* **solver**
  - [spdynmod](#), [6](#)

[animate\\_bs](#), [2](#)  
[animate\\_model](#), [2](#)  
[animate\\_rb](#), [3](#)  
[animate\\_sm](#), [3](#)  
[animate\\_ss](#), [4](#)

[mrgf](#), [4](#)

[neigh\\_cell](#), [5](#)

[plot\\_maps](#), [5](#)

[run\\_model](#), [6](#)

[spdynmod](#), [6](#)  
[spdynmod-package \(spdynmod\)](#), [6](#)