

# Package ‘Jsmm’

February 20, 2026

**Title** Joint Species Movement Modelling with the R-package Jsmm

**Version** 0.1.0

**Description** Software for simulating and predicting movement data based on diffusion-advection-reaction models.

**License** CC BY-NC-ND 4.0 + file LICENSE

**Encoding** UTF-8

**Author** Luisa F. Rodriguez [aut, cre](ORCID: <<https://orcid.org/0000-0002-8022-6522>>), Otso Ovaskainen [aut] (ORCID: <<https://orcid.org/0000-0001-9750-4421>>)

**Maintainer** Luisa F. Rodriguez <rpackagejsmm@gmail.com>

**URL** <https://github.com/lufrodriguezca/Jsmm>

**Imports** MASS,  
MCMCpack,  
Matrix,  
RTriangle,  
Rcpp,  
colorspace,  
ggplot2,  
coda,  
sf

**LinkingTo** Rcpp

**RoxygenNote** 7.3.3

## Contents

compute_likelihood . . . . .	2
convert_to_coda_object . . . . .	2
creating_domain_from_locations . . . . .	3
jsmm . . . . .	4
jsmm_add_triangulation . . . . .	5
mean_captures . . . . .	6
mean_distance_between_captures . . . . .	6
mean_time_to_capture . . . . .	7
merge_rc_data . . . . .	7
model_summary . . . . .	8
number_of_captures . . . . .	8

plot_domain . . . . .	9
plot_observation_effort . . . . .	9
plot_trajectories . . . . .	10
poolMcmcChains . . . . .	10
sampleMcmc . . . . .	11
simulate_captures . . . . .	12
simulate_from_prior . . . . .	12
simulate_trajectories . . . . .	13

**Index****14**


---

compute_likelihood	<i>Compute the likelihood</i>
--------------------	-------------------------------

---

**Description**

Compute the likelihood in log form for the Joint species movement model

**Usage**

```
compute_likelihood(m, Beta, max_dt = NULL)
```

**Arguments**

- m            Named list that contains all the model components for running Joint species movement modelling (jsmm) analyses.
- Beta        A named list with the movement and observation parameters included in the Jsmm model.
- max\_dt      Optional positive integer value representing the number of time partitions in which the PDE should be solved.

**Value**

A named list with the loglikelihood computations for each species.

---

convert_to_coda_object	<i>Convert to Coda object</i>
------------------------	-------------------------------

---

**Description**

Converts the posterior data generated after fitting the jsmm model into an object to the class mcmc from the Coda R-package.

**Usage**

```
convert_to_coda_object(
  m,
  start = 1,
  Beta = TRUE,
  Gamma = TRUE,
  rho = TRUE,
  SI = TRUE,
  li = TRUE
)
```

**Arguments**

<b>m</b>	Named list that contains all the model components for running Joint species movement modelling (jsmm) analyses.
<b>start</b>	Start of the MCMC
<b>Beta</b>	Optional logical data type. Set to TRUE to include Beta parameters, FALSE otherwise.
<b>Gamma</b>	Optional logical data type. Set to TRUE to include Gamma parameters, FALSE otherwise.
<b>rho</b>	Optional logical data type. Set to TRUE to include rho parameters, FALSE otherwise.
<b>SI</b>	Optional logical data type. Set to TRUE to include Sigma parameters, FALSE otherwise.
<b>li</b>	Optional logical data type. Set to TRUE to include Markov chain Monte Carlo chains, FALSE otherwise.

**Value**

A named list with objects from the class mcmc from coda.

**creating\_domain\_from\_locations**

*Create a simulated domain*

**Description**

This function allows the creation of a customized rectangular domain, capture, and release sites.

**Usage**

```
creating_domain_from_locations(
  lc,
  lc_label = NA,
  buffer_x = c(0, 0),
  buffer_y = c(0, 0),
  nnod_side_x,
  nnod_side_y,
  rad_capture_site,
```

```

    n_sides_capture_site,
    angle_capture_site,
    int_buffer = FALSE
)

```

### Arguments

lc	matrix that provides the positions of the capture and release sites. Columns correspond to the x and y coordinates, respectively.
lc_label	character vector with capture/release sites labels (optional).
buffer_x	real positive number representing the distance from the center of the outer capture/release location to the domain boundary in the x-axis.
buffer_y	real positive number representing the distance from the center of the outer capture/release location to the domain boundary in the y-axis.
nnod_side_x	integer positive representing the number of nodes in x axes.
nnod_side_y	integer positive representing the number of nodes in y axes.
rad_capture_site	real positive value representing the capture/release site radius.
n_sides_capture_site	Number of sides for the capture/release polygon.
angle_capture_site	input parameter in radians measuring the counterclockwise rotation angle for the capture/release polygon.
int_buffer	boolean parameter for inserting an intermediate buffer between the border of the captures/releases array and the landscape boundary.

### Value

Polygonal object from the class sf.

jsmm

*Define the Jsmm model*

### Description

Creates a named list with all the necessary components for running the Jsmm analyses.

### Usage

```

jsmm(
  domain,
  observation_effort,
  releases = NULL,
  captures = NULL,
  cov_data,
  ns,
  sp_names = NULL,
  C = NULL,
  tr_data = NULL,
  model_formula
)

```

**Arguments**

domain	named list with the elements domain and triangulation. The former is a polygonal object from the class sf, and the latter provides information about the domain triangulation.
observation_effort	named list with spatial and temporal information about releases and captures in the experiment.
releases	named list with information about where and when releases occurred.
captures	named list with information about where and when captures occurred.
cov_data	named list with (some combination of) spatial, temporal or/and spatio-temporal data.
ns	integer representing the number of species included in the model.
sp_names	string vector parameter assigning the species names in the model (optional).
C	phylogenetic relationships matrix between species.
tr_data	dataframe with trait data.
model_formula	named list that consists of (some combination of) the objects diffusion, advection, mortality, habitat_preference, observation, and traits. These follow the R-notation for the model formulae.

**Value**

A named list with all the information for running Jsmm analyses.

**jsmm\_add\_triangulation**

*Add triangulation*

**Description**

Create a domain triangulation from a polygonal object from the class sf.

**Usage**

```
jsmm_add_triangulation(domain, max_t_area, min_t_angle)
```

**Arguments**

domain	polygonal object from the class sf.
max_t_area	Maximum triangle area.
min_t_angle	Minimum triangle angle in degrees.

**Value**

A named list with the triangulation and its corresponding polygonal sf object.

mean_captures	<i>Compute the captured individuals mean</i>
---------------	--

### Description

This function computes the mean of the captured individual for each species. This function is used for evaluating the model fit (posterior predictive data).

### Usage

```
mean_captures(m, captures)
```

### Arguments

- |                       |   |
|-----------------------|---|
| <code>m</code>        | Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses. |
| <code>captures</code> | Matrix with the capture data.   |

### Value

A vector with the estimates for each species.

mean_distance_between_captures	<i>Compute the mean distance between captures</i>
--------------------------------	---

### Description

This function computes the mean euclidean distance between captures for each species. This function is used for evaluating the model fit (posterior predictive data).

### Usage

```
mean_distance_between_captures(m, captures)
```

### Arguments

- |                       |   |
|-----------------------|---|
| <code>m</code>        | Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses. |
| <code>captures</code> | Matrix with the capture data.   |

### Value

A vector with the computations for each species.

---

mean\_time\_to\_capture    *Compute the mean time between captures*

---

## Description

This function computes the mean time between captures for each species. This function is used for evaluating the model fit (posterior predictive data).

## Usage

```
mean_time_to_capture(m, captures)
```

## Arguments

- |          |   |
|----------|---|
| m        | Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses. |
| captures | Matrix with the capture data.   |

## Value

A vector with the computations for each species.

---

merge\_rc\_data                  *Merge releases and capture data*

---

## Description

Sets releases and capture data.

## Usage

```
merge_rc_data(m)
```

## Arguments

- |   |   |
|---|---|
| m | Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses. |
|---|---|

## Value

A modified Jsmm model m.

`model_summary`      *Show summary of the Jsmm model.*

## Description

Generates a summary of all the Jsmm components.

## Usage

```
model_summary(m)
```

## Arguments

`m`      named list corresponding to the Jsmm model.

## Value

Produces a visualization on screen of the Jsmm main components.

`number_of_captures`      *Calculate the number of captures*

## Description

This function computes captured individuals for each species. This function is used for evaluating the model fit (posterior predictive data).

## Usage

```
number_of_captures(m, captures)
```

## Arguments

`m`      Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses.

`captures`      Matrix with the capture data.

## Value

A vector with the computations for each species.

`plot_domain`      *Plot domain*

## Description

Create a plot from the polygonal triangulation.

## Usage

```
plot_domain(domain, customize_plot = NULL)
```

## Arguments

domain polygonal object from the class sf.

`customize_plot` list with optional presetting for visualization.

## Value

A customized domain plot.

## plot\_observation\_effort

## Description

Visualize where and when captures and releases occurred.

## Usage

```
plot_observation_effort(  
  observation_effort,  
  xlab,  
  ylab,  
  by_x,  
  cex.axisx,  
  cex.axisy,  
  lasy,  
  tcky  
)
```

## Arguments

## observation\_effort

Named list with releases and captures information.

**xlab** Customized labels for the x-axis.

**ylab** Customized labels for the y-axis.

by\_x Ticks in x frequency.

cex.axisx      Size text x.  
 cex.axisy      Size text y.  
 lasy            Rotation label y.  
 tcky            Control tick marks.

**Value**

A plot with releases and captures information.

**plot\_trajectories**      *Plot movement trajectories.*

**Description**

The function allows visualization of the movement trajectories generated by the function simulate\_trajectories().

**Usage**

```
plot_trajectories(m = m, plot_sp = NULL, customize_plot = NULL)
```

**Arguments**

m                Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses.  
 plot\_sp          vector composed of integers representing the species index to visualize.  
 customize\_plot   list with predefined settings for visualization.

**Value**

NULL. Visualization function.

**poolMcmcChains**      *Pool MCMC chains*

**Description**

This function sets up the posterior data in suitable format. Used for showing posterior vs priors.

**Usage**

```
poolMcmcChains(postList, chainIndex = 1:length(postList), start = 1, thin = 1)
```

**Arguments**

postList        List with the model posterior data  
 chainIndex      Index of the MCMC chain.  
 start           Starting point MCMC chain.  
 thin            Thin MCMC chain

**Value**

A list with the formatted posterior data.

sampleMcmc

*Sample MCMC***Description**

Performs all the computations for the parameter estimation for the the Joint species movement model (Jsmm) via Markov Chain Monte Carlo (MCMC).

**Usage**

```
sampleMcmc(
  m,
  samples,
  transient,
  thin,
  max_dt = NULL,
  init_pars = NULL,
  nChains = 1,
  no.data = FALSE
)
```

**Arguments**

<code>m</code>	Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses.
<code>samples</code>	Integer parameter representing the number of samples drawn from the MArkov Chain.
<code>transient</code>	Integer parameter representing the initial part of the chain considered as transient.
<code>thin</code>	Integer parameter representing the thinning.
<code>max_dt</code>	Optional positive integer value representing the number of time partitions in which the PDE should be solved.
<code>init_pars</code>	Initial parameters for fitting the model.
<code>nChains</code>	Number of MCMC chains to simulate.
<code>no.data</code>	Boolean parameter indicating presence or absence of data.

**Value**

This functions adds to the named list `m`, the posterior data, `thin`, `samples`, and `transient` information.

`simulate_captures`      *Simulate movement data.*

### Description

The function returns the model object m, to which it has added the objects captures and releases.

### Usage

```
simulate_captures(m, pars, releases, max_dt = NULL, secondary_release)
```

### Arguments

<code>m</code>	named list created by the function jsmm().
<code>pars</code>	named list with the objects Beta, rho, SI, and Gamma defining the model parameters.
<code>releases</code>	matrix whose first and second columns provide information about the species and release event, respectively.
<code>max_dt</code>	real number bigger than zero that describes the maximal time resolution used in numerical computations.
<code>secondary_release</code>	boolean parameter that describes whether the captured individuals are released after capture.

### Value

The function returns the model object m, to which it has added the objects captures and releases.

`simulate_from_prior`      *Simulate movement and observation parameters from priors.*

### Description

This function generates the movement and observation parameters from the priors rho (phylogenetic signal parameter), parameter Sigma for generating the variance-covariance matrix, and vector Gamma.

### Usage

```
simulate_from_prior(m)
```

### Arguments

<code>m</code>	Named list that contains all the model components for running Joint species movement modelling (Jsmm) analyses.
----------------	---

### Value

A named list with the movement and observation parameters, and the corresponding rho, Sigma, and Gamma priors.

---

```
simulate_trajectories Simulate movement trajectories.
```

---

## Description

The function generates trajectories for a movement model in which secondary releases are not possible.

## Usage

```
simulate_trajectories(m, pars, releases, max_dt = NULL)
```

## Arguments

m	named list created by the function jsmm().
pars	named list with the objects Beta, rho, SI, and Gamma defining the model parameters.
releases	matrix whose first and second columns provide information about the species and release event, respectively.
max_dt	real number bigger than zero that describes the maximal time resolution used in numerical computations.

## Value

#' The function returns the model object m, to which it has been added a matrix called trajectories.

# Index

compute\_likelihood, 2  
convert\_to\_coda\_object, 2  
creating\_domain\_from\_locations, 3  
  
jsmm, 4  
jsmm\_add\_triangulation, 5  
  
mean\_captures, 6  
mean\_distance\_between\_captures, 6  
mean\_time\_to\_capture, 7  
merge\_rc\_data, 7  
model\_summary, 8  
  
number\_of\_captures, 8  
  
plot\_domain, 9  
plot\_observation\_effort, 9  
plot\_trajectories, 10  
poolMcmcChains, 10  
  
sampleMcmc, 11  
simulate\_captures, 12  
simulate\_from\_prior, 12  
simulate\_trajectories, 13