

Package ‘camtrapHawkes’

May 7, 2024

Title Analyze camera trap data with Hawkes processes

Version 1.1.0

Description Helper functions to infer a Hawkes process from camera trap data. This code relies heavily on the UnitEvents (https://sourcesup.renater.fr/frs/?group_id=3267) and ppstat (<https://github.com/nielsrhansen/ppstat>) packages for Hawkes process simulation and inference but implements helper functions for formatting and plotting.

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = FALSE)

RoxygenNote 7.2.3

Imports dplyr,
ggplot2,
ggraph,
ggtext,
grDevices,
grid,
gridExtra,
igraph,
lubridate,
magrittr,
pammtools,
patchwork,
ppstat,
rlang,
stats,
stringr,
tibble,
tidygraph,
tidyr,
tidyselect,
UnitEvents,
viridis

R topics documented:

add_stamps	3
compute_box	3
compute_intervals	4
compute_intervals_permute	4
compute_pos_neg	5
compute_pos_neg_list	5
compute_rate	6
compute_sensi	6
compute_speci	7
compute_TN_TP	7
create_funcshape	8
create_interactions	8
create_interactions_ppstat	9
create_spont_rates	9
df_to_matrix	10
filter_inactive_cameras	10
format_data_perf	11
get_bins_sum	11
get_bins_sum_list	12
get_ids	12
get_models_df	13
get_ppstat_coeffs	13
get_ppstat_interactions	14
get_sampling_info	14
infer	15
mlist_to_df	15
name_dataneur	16
plot_background_rate	17
plot_background_rate_simu	18
plot_bias	18
plot_graph	19
plot_interactions	20
plot_interactions_simu	21
plot_observed_rate	23
plot_perf	24
reinfer	25
require	25
shift_duplicates	26
ue_model_to_df	26
ue_to_df	27
ue_to_graph	27
unpack_inf	28
write_formula	28

add_stamps	<i>Add timestamps</i>
------------	-----------------------

Description

Add a timestamps column to df.

Usage

```
add_stamps(df, origin = NULL, unit = "days")
```

Arguments

df	a dataframe with datetime column of type date.
origin	optional origin (for which stamp is zero)
unit	time unit to use for the stamps (delay in ...)

Value

A dataframe with one more column (named stamp)

compute_box	<i>Compute box</i>
-------------	--------------------

Description

Compute time windows covering the whole dataset for a given dataset d.

Usage

```
compute_box(d, use_stamps = FALSE)
```

Arguments

d	dataframe with columns stamp, datetime and cameraID
use_stamps	use timestamps or dates?

Value

Returns the estimated box (matrix) line 1: Camera number line 2: start time line 3: stop time line 4: compartment ID each column corresponds to one camera.

compute_intervals	<i>Compute inter-event times</i>
-------------------	----------------------------------

Description

Compute the inter-event times between two species. Function adapted from Murphy et al 2021 (<http://onlinelibrary.wiley.com/doi/abs/10.1111/1365-2656.13548>)

Usage

```
compute_intervals(sp_from, sp_to)
```

Arguments

sp_from	species that is observed first
sp_to	following species

Value

Returns the median of the time interval between the 2 species occurrences

compute_intervals_permute	<i>Compute permuted inter-event times</i>
---------------------------	---

Description

Compute the permuted inter-event times between species. Function adapted from Murphy et al 2021 (<http://onlinelibrary.wiley.com/doi/abs/10.1111/1365-2656.13548>)

Usage

```
compute_intervals_permute(sp_from, sp_to)
```

Arguments

sp_from	species that is observed first
sp_to	following species

Value

Returns the median of the permuted time interval between the 2 species occurrences

compute_pos_neg	<i>Compute true/false positives/negatives</i>
-----------------	---

Description

Compute: TP (true positives) FP (false positives) TN (true negatives) FN (false negatives)

Usage

```
compute_pos_neg(Msimul, Mtrue)
```

Arguments

Msimul	a K*K*Ncomp array (K species Ncomp different compartments (often Ncomp =1))
Mtrue	true array (K*K*Ncomp array)

Value

Returns a named vector with four elements named TP, FP, TN, FN

compute_pos_neg_list	<i>Compute several true/false positives</i>
----------------------	---

Description

Evaluates several inferred models (in Msim_list) compared to the true model used to generate data (Mtrue). Computes: TP (true positives) FP (false positives) TN (true negatives) FN (false negatives)

Usage

```
compute_pos_neg_list(Msim_list, Mtrue, est = "BL")
```

Arguments

Msim_list	a list, each element has elements \$BL and \$BOL and in each there are \$\$ (spontaneous part) and \$I (interaction coefficients, lists.)
Mtrue	true model: array (K*K*Ncomp array)
est	estimator to use (BL, BVL or BOL)

Value

Returns a df with columns TP, FP, TN, FN and as many rows as there are models in Msim_list.

compute_rate	<i>Compute rates</i>
--------------	----------------------

Description

Compute the rate corresponding to data with a given model.

Usage

```
compute_rate(model, data, timestep = 0.01)
```

Arguments

model	a dataframe corresponding to the model estimated with UnitEvents. This dataframe can be obtained with the function "ue_model_to_df". Must have columns time, excitefunc, to, from and spont.
data	occurrence data. Must have one column stamp and one column species. Currently only one camera is supported.
timestep	timestep for the function discretisation (days). Should be smaller than delta used in the model for good results. Defaults to 0.01.

Value

A dataframe with columns time, lambda and species. time: the time. lambda: the intensity function at time t. species: the species for which the intensity is computed.

compute_sensi	<i>Computes sensitivity</i>
---------------	-----------------------------

Description

Computes sensitivity from true positives and false negatives

Usage

```
compute_sensi(TP, FN)
```

Arguments

TP	true positives vector
FN	false negatives vector

Value

The sensitivity (true positive rate)

compute_speci	<i>Computes specificity</i>
---------------	-----------------------------

Description

Computes specificity from true negatives and false positives

Usage

```
compute_speci(TN, FP)
```

Arguments

TN	true negatives vector
FP	false positives vector

Value

The specificity (true negative rate)

compute_TN_TP	<i>Compute TP/TN</i>
---------------	----------------------

Description

Compute true positive and true negative rates from p-values.

Usage

```
compute_TN_TP(pval, true, alpha)
```

Arguments

pval	a vector of p-values
true	the ground truth vector (must be in the same order as pval)
alpha	the significance threshold (defaults to 0.05)

Value

A named vector TP, FP, TN, FN with the values.

create_funcshape	<i>Create function shape</i>
------------------	------------------------------

Description

Create a function shape.

Usage

```
create_funcshape(funcshape, times, t = 0.5, strength)
```

Arguments

funcshape	exp, gamma or linear
times	times vector
t	half-life (time value for which the function reaches strength/2). only used when funcshape == "exp")
strength	function max strength coefficient

Value

A numeric vector of the same length as the times vector containing function values computed for each time.

create_interactions	<i>Create interactions</i>
---------------------	----------------------------

Description

Create an array of null interaction functions

Usage

```
create_interactions(spp_names, times)
```

Arguments

spp_names	names of species
times	times vector

Value

A (nspecies, nspecies, 1) array containing in each cell a (2, length(times)) matrix: top row is a vector of zero (null) interaction function), borrom row is the times vector.

```
create_interactions_ppstat
```

Create interactions

Description

Create a list of interactions as needed for a ppstat input filled with zeroes.

Usage

```
create_interactions_ppstat(spp_names, times)
```

Arguments

spp_names	Species names vector
times	Times vector

Value

A list of lists. Each list represents one species, and within each list there are the interactions with this species so that `flist[[i]][[j]]` represents the effect of $j \rightarrow i$.

Examples

```
create_interactions_ppstat(spp_names = letters[1:5], times = seq(0, 10, by = 1))
```

```
create_spont_rates
```

Create background rates

Description

Create a background rates vector.

Usage

```
create_spont_rates(spont, spp_names)
```

Arguments

spont	spontaneous coefficient if unique value will be repeated else will be in the order provided
spp_names	names of species

Value

Returns a matrix of dim (nspecies, 1) containing named spont rates

df_to_matrix	<i>Transform dataframe to matrix</i>
--------------	--------------------------------------

Description

Transforms the dataframe into a matrix ready for Lasso inference.

Usage

```
df_to_matrix(df)
```

Arguments

df	A dataframe with columns cameraID stamp species count
----	---

Value

an array (nspecies, ncameras, nmax+1) in the same format as the output of HawkesMulti from the UnitEvents package). nmax is the maximum number of occurrences for one species on one camera.

filter_inactive_cameras	<i>Filter out cameras</i>
-------------------------	---------------------------

Description

Filter out cameras that have not enough observations/not enough frequent observations

Usage

```
filter_inactive_cameras(df, thr_obs, thr_freq, plot = TRUE)
```

Arguments

df	dataframe to filter. Must have columns cameraID, snapshotName, count, stamp
thr_obs	minimal number of observations to keep camera
thr_freq	minimal frequency of observations to keep camera
plot	plot graphs?

Value

Returns the filtered dataframe. If plot = TRUE, ans plots camera sampling information.

format_data_perf	<i>Format performance dataframe</i>
------------------	-------------------------------------

Description

Format a dataframe measuring sensitivity and specificity

Usage

```
format_data_perf(d, add_quantiles = TRUE, level = 0.05)
```

Arguments

d	dataframe: must have columns TP, FP, TN, FN. additional columns possible, but they will affect quantiles computation if add_quantiles is TRUE.
add_quantiles	should quantiles for sensitivity and specificity be computed? Quantiles are computed on values grouped by all columns (except "TP", "FP", "TN", "FN").
level	level of the quantiles.

Value

Returns a cleaned form of d: additional column sensi computed as $TP/(TP+FN)$ additional column speci computed as $TN/(TN+FP)$ the columns are pivoted to longer format so that sensi and speci are grouped into "value" column and "type" describes sensi or speci. if add_quantiles, quantile values of value (sensi/speci) will be added (quantile computed grouping according to all other columns excluding "TP", "FP", "TN", "FN").

get_bins_sum	<i>Get bins sum</i>
--------------	---------------------

Description

Get the sum of the bins for a Hawkes model.

Usage

```
get_bins_sum(M)
```

Arguments

M	an interaction matrix($K \times K \times N_{comp}$ array) (K species and N_{comp} different compartments (often $N_{comp} = 1$))
---	--

Value

A $K \times K$ matrix containing 1 if the interaction was inferred, else 0

get_bins_sum_list	<i>Get bins sums</i>
-------------------	----------------------

Description

Get the sum of the bins for a list of Hawkes model.

Usage

```
get_bins_sum_list(Mlist)
```

Arguments

Mlist	Mlist: a list of interaction matrix($K \times K \times N_{comp}$ array) (K species and N_{comp} different compartments (often $N_{comp} = 1$))
-------	--

Value

A $K \times K$ matrix containing the number of times each value was inferred

get_ids	<i>Get IDs</i>
---------	----------------

Description

Get a unique row ID for each combination of species/camera.

Usage

```
get_ids(df, delete_missing_species = FALSE)
```

Arguments

df	a dataframe that must have columns species cameraID
delete_missing_species	delete species not seen at all cameras?

Value

a dataframe with columns cameraID species rowid (species_cameraID) The df is arranged with all species for the same site first

get_models_df	<i>Get models dataframe</i>
---------------	-----------------------------

Description

Get a list of all models in inferred_models plus the true model

Usage

```
get_models_df(inferred_models, models, est = "BL")
```

Arguments

inferred_models	a list with 2 components \$reinfer: list of models, each one inferred from a different dataset which is assumed to have been generated by the same true model. \$reinfer_parameters: metadata about the inferred models
models	a list of true models to match the true model that have been used to generate data from which models in \$reinfer were inferred.
est	estimator to use (\$BL or \$BOL)

Value

A dataframe of models

get_ppstat_coeffs	<i>Get ppstat coefficients</i>
-------------------	--------------------------------

Description

Returns the coefficients an inferred ppstat model.

Usage

```
get_ppstat_coeffs(model, term, alpha = 0.05)
```

Arguments

model	model object used for ppstat.
term	dimension to pick (ie species index)
alpha	wanted confidence level

Value

A dataframe with columns 'coef_id', 'species', 'coef', 'lower' and 'upper'.

```
get_ppstat_interactions
```

Get ppstat interactions

Description

Get toe interaction functions for ppstat

Usage

```
get_ppstat_interactions(model, alpha = 0.05, trans = NULL)
```

Arguments

model	ppstat model (assumed to be multivariate)
alpha	confidence level for the confidence intervals (see the documentation of ‘ppstat::termPlot’)
trans	transformation to apply (eg ‘exp’ for a log-intensity) (see the documentation of ‘ppstat::termPlot’)

Value

A dataframe with columns:

+ ‘x’: the time + ‘variable’: response variable (species) + ‘value’: interaction value + ‘cf.lower’/‘cf.upper’: confidence interval bounds + ‘response’: response variable

```
get_sampling_info
```

Get sampling info

Description

Get sampling info

Usage

```
get_sampling_info(df, return = TRUE)
```

Arguments

df	A camera trap dataframe. Must have columns cameraID stamp
return	return the summarized dataframe?

Value

prints a message summarizing sampling info for the cameras.

infer	<i>Infer a Hawkes model</i>
-------	-----------------------------

Description

Infer Hawkes model for data d.

Usage

```
infer(
  d,
  k = 12,
  delta = 2/24,
  Z = 0,
  gamma = 0.5,
  scale = 10000,
  use_stamps = FALSE
)
```

Arguments

d	dataset to infer from. Must have columns: snapshotName cameraID stamp date-time count
k	number of bins for the inference function
delta	binwidth
Z	cutoff
gamma	penalization LASSO parameter
scale	scale for precision of calculations (defaults to 10000)
use_stamps	use timestamps or dates?

Value

Returns the output in the same form as BoxLasso() function of UnitEvents A list with 2 (if n(species) = 1) or 3 elements (if n(species) > 1) \$BL \$BVL \$BOL

mlist_to_df	<i>List of models to dataframes</i>
-------------	-------------------------------------

Description

Transform a list of UnitEvent models to a dataframe

Usage

```
mlist_to_df(mlist, comp = 1)
```

Arguments

- mlist list of models of type UnitEvents each component of mlist must be a list with 2
 named elements \$S and \$I (corresponds to a model\$BL, BOL or \$BVL output
 of BoxLasso)
- comp the compartment number to extract

Value

A dataframe with 4 or 5 columns: time: time for the interaction functions excitefunc: value of the
excitation function from: from species to: to species spont: spontaneous rate of the "to" species rep
(if mlist is a named list): name of the i-th element of mlist

name_dataneur	<i>Name data</i>
---------------	------------------

Description

Attribute colnames and rownames to dataneur generated with Hawkesmulti.

Usage

```
name_dataneur(dn, species_names, cameras_names = NA)
```

Arguments

- dn an array (nspecies, ncameras, nmax+1) in the same format as the output of
 HawkesMulti from the UnitEvents package).
- species_names ordered vector of species names.
- cameras_names ordered vector of cameras names.

Value

The same matrix as the input (dn) but with named rows and columns.

plot_background_rate *Plot background rates*

Description

Plot the background rate for species.

Usage

```
plot_background_rate(
  ue_df,
  title = NA,
  textsize = 10,
  silhouettes = NA,
  confint = NULL,
  col_conf = "cornflowerblue",
  width_conf = 0.2,
  write_label = FALSE,
  nudge_label = 0.3
)
```

Arguments

ue_df	ue_df: dataframe with results of UnitEvents inference. Must have columns: + spont + to The function will group values of "spont" per "to" and get the unique values that should be unique for each species.
title	plot title
textsize	text minimal size (for x and y axes)
silhouettes	optional labels with animal silhouettes to replace default labels. If it exists, must be a named vector with each name corresponding to a species name in ue_df. The elements are markdown codes containing a element.
confint	If you want confidence intervals around the value of the background rate, use this argument. The first element is the name of the column containing the lowest bound of the confidence interval and the second element is the highest bound.
col_conf	Color of the error bars
width_conf	Width of error bars
write_label	write background rates values besides the points?
nudge_label	if the background rates are written, by how much should they be nudged on the x-axis?

Value

a ggplot object, representing background rates for each species.

plot_background_rate_simu	<i>Plot background rates for simulation</i>
---------------------------	---

Description

Plots the inferred background rates vs the real rates

Usage

```
plot_background_rate_simu(df, alpha = 0.05, title = NA)
```

Arguments

df	the dataframe with true and estimated intensities. It has columns: rep: repetition ID (s... for simul, true for true data) excitefunc: value of the excitation function from/to: ID of the species time: time in days
alpha	confidence level for plotting
title	optional plot title

Value

A ggplot of the background rates where the true rate is in red and the inferred rate in blue with a confidence interval (if there were several inferred models.)

plot_bias	<i>Plot bias</i>
-----------	------------------

Description

Plots the bias for each interaction from several inferences.

Usage

```
plot_bias(bias_df, fill = "valprop", textsize = 12)
```

Arguments

bias_df	a dataframe containing the summarized results of the inference for several repetitions. Must have columns: from and to (interacting species) the column specified with 'fill'
fill	name of the column to use for values to fill the plot.
textsize	size of the text

Value

a ggplot tile with colors corresponding to the proportion of times the corresponding interaction was inferred.

plot_graph

Plot graph

Description

Plots a graph g.

Usage

```
plot_graph(
  g,
  layout = c(),
  repel = FALSE,
  coledges = "grey",
  colnodes = "cadetblue3",
  coltext = "black",
  textsize = 5,
  s = 8,
  arrsize = 6,
  nudge_x = NA,
  parse_labels = FALSE,
  use_labels_column = FALSE
)
```

Arguments

g	The graph to plot (tbl_graph object)
layout	optional layout (defaults to layout_in_circle)
repel	repel node labels?
coledges	color of edges (a string color name)
cnodes	color of nodes (named vector named as species or a string color name). If it is NULL, species will be colored automatically.
coltext	color of text for the node labels. It can be a vector (will correspond to species alphabetical order),
textsize	node label size
s	size of nodes
arrsize	size of arrow
nudge_x	optional vector to move x labels for node labels.
parse_labels	parse text labels?
use_labels_column	use column 'label' for nodes labels?

Value

A ggplot object representing the graph.

plot_interactions	<i>Plot interaction functions</i>
-------------------	-----------------------------------

Description

Plot UnitEvents interaction functions

Usage

```
plot_interactions(  
  ue_df,  
  scale = "days",  
  title = NA,  
  relative = FALSE,  
  silhouettes = NA,  
  timestep = NA,  
  ystep = NA,  
  textsize = 10,  
  linesize = 0.5,  
  line_geom = c("step", "line"),  
  confint = NULL,  
  scales = "fixed",  
  col_conf = "cornflowerblue",  
  col_baseline = "black",  
  alpha_conf = 0.7,  
  h = 0.3,  
  baseline = 0,  
  separate_self = FALSE  
)
```

Arguments

ue_df	dataframe with results of UnitEvents inference. Must have columns: + time + excitefunc + from + to
scale	days or hours, following whether we want the time axis graduated in days or hours
title	plot title
relative	plot intensity absolute or relative (divided by spont) value?
silhouettes	optional labels with animal silhouettes to replace default labels. If it exists, must be a named vector with each name corresponding to a species name in ue_df. The elements are markdown codes containing a element.

timestep	optional timestep for x-axis.
ystep	optional step for y-axis (function values).
textsize	text minimal size (for x and y axes)
linesize	linewidth
line_geom	The 'geom' to use for the interaction function line: 'geom_step' (adapted for 'UnitEvents') or 'geom_line'.
confint	If you want confidence intervals around the value of the interaction function, use this argument. The first element is the name of the column containing the lowest bound of the confidence interval and the second element is the highest bound. If 'NULL' (the default) no confidence interval is plotted.
scales	Scale parameter for 'facet_grid'. Defaults to 'fixed' (fixed scales), see the documentation of 'ggplot2::facet_grid' for more information. Only used for the inter-species interactions (in case 'separate_self = TRUE').
col_conf	Color of the confidence interval
col_baseline	Color of the line used to show the baseline
alpha_conf	Transparency of the confidence interval
h	Horizontal spacing between plots (if separate self)
baseline	Baseline against which to compare interaction functions. Defaults to zero (adapted for UnitEvents model).
separate_self	whether to separate auto-interactions and plot them above

Value

ggplot object, a plot with the pairwise interaction functions between species.

plot_interactions_simu

Plot interaction function for simulation

Description

Plots the inferred interaction functions vs the real function

Usage

```
plot_interactions_simu(
  df,
  title = NA,
  textsize = 10,
  baseline = 0,
  timestep = NA,
  line_geom = c("step", "line"),
  col_true = "brown3",
```

```

col_infer = "darkblue",
col_conf = "cornflowerblue",
conf = c("sample", "confint"),
col_baseline = "black",
show.legend = FALSE,
parse = FALSE,
y_angle = 90,
alpha_conf = 0.5,
level = 0.05
)

```

Arguments

df	a dataframe containing values for the inferred function(s) and the true function. It has columns: + 'rep': repetition ID (s... for simul, true for true data) + 'excitefunc': value of the excitation function + 'from/to': ID of the species + 'time': time in days + if 'conf = confint': 'cf.lower' and 'cf.upper' with the pre-computed confidence intervals.
title	optional plot title
textsize	text minimal size (for x and y axes)
baseline	Baseline against which to compare interaction functions. Defaults to zero (adapted for UnitEvents model).
timestep	optional timestep for x-axis.
line_geom	The 'geom' to use for the interaction function line: 'geom_step' (adapted for 'UnitEvents') or 'geom_line'.
col_true	Color of the true function
col_infer	Color of the inferred function
col_conf	Color of the confidence interval around the inferred function
conf	The type of confidence interval to display: either the confidence interval computed as the quantiles of the inferred functions (if 'sample') or the median of the pre-computed confidence interval ('confint'). The second option is useful when the model was inferred with 'ppstat'. If 'conf = confint', 'df' must have two additional columns 'cf.lower' and 'cf.upper'.
col_baseline	color of the line for the baseline to compare the function to.
show.legend	Show the legend?
parse	parse the facet labels?
y_angle	Angle for the text in y in the panels
alpha_conf	Transparency of the confidence interval around the inferred function
level	confidence level to use around simulations

Value

A ggplot of the interaction functions where the true function is in red and the inferred function in blue with a confidence interval (if there were several inferred models.)

plot_observed_rate	<i>Plot rates</i>
--------------------	-------------------

Description

Plot the rate of an observed sequence of events.

Usage

```
plot_observed_rate(
  rates,
  data,
  timestep = 2,
  textsize = 20,
  ptsize = 1,
  lwd = 1,
  t1,
  t2,
  hlambda,
  hpoints,
  minor_spacing,
  major_spacing,
  max_lambda,
  ybreaks,
  cols,
  ylabel = TRUE,
  log = FALSE,
  line_geom = c("step", "line"),
  xlab = "Time (days)"
)
```

Arguments

rates	a dataframe with pre-computed rates. Must have columns: time lambda species
data	the corresponding occurrence data. Must have columns: stamp species
timestep	timestep to plot
textsize	base text size (axes text, axes labels are a 2 units bigger)
ptsize	point sizes for the events
lwd	linewidth for intensities
t1	minimal time bounds for subsetting data. If missing, then all data will be plotted.
t2	maximal time bounds for subsetting data. If missing, then all data will be plotted.
hlambda	plot relative heights for intensity panels
hpoints	plot relative heights for points panels

minor_spacing	space between the intensity function and the corresponding points.
major_spacing	space between points and the following intensity.
max_lambda	max value for y-axis for rates
ybreaks	breaks for y-axis (for lambda plots)
cols	named vector for colors: names are species names and contain colors
ylabel	display labels ?
log	if TRUE, the scales will be plotted on a log10 axis.
line_geom	The 'geom' to use for the interaction function line: 'geom_step' (adapted for 'UnitEvents') or 'geom_line'.
xlab	xlabel to display (optional)

Value

A ggplot object generated with patchwork. Multiple plots in the same column, where all plots are paired, the top plot representing the intensity and the bottom plot the actual occurrences for one species.

plot_perf

Plots performance

Description

Plots the performance of the inference.

Usage

```
plot_perf(d, xaxis = "trapping_days", thr, vline, psize = 1)
```

Arguments

d	the dataframe of observed sensitivity and specificity. It has columns: value (sensitivity or specificity value) type (sensitivity or specificity as "sensi" or "speci") "xaxis" (a measure of trapping days) possibly qinf, qsup (then quantiles are plotted around sensitivity and specificity values.)
xaxis	name of the x-axis to choose for plotting (must be present in d)
thr	optional threshold where to plot a horizontal line
vline	optional vline to plot to draw attention to a specific time
psize	point sizes

Value

a ggplot object with the sensitivity and specificity displayed along xaxis.

reinfer	<i>Reinfer Hawkes model</i>
---------	-----------------------------

Description

Reinfer Hawkes model from a dataset generated with a Hawkes model.

Usage

```
reinfer(M, Ntrial, k = 12, delta = 2/24, gamma = 0.5, scale = 10000)
```

Arguments

M	dataset to infer from, a DataNeur matrix (output of HawkesMulti from the UnitEvents package). It is an array (nspecies, ncameras, nmax+1) where nmax is the maximum number of occurrences for one species on one camera.
Ntrial	number of cameras
k	number of bins for the interaction functions
delta	binwidth
gamma	penalization LASSO parameter
scale	scale for precision of calculations (defaults to 10000)

Value

Returns the output in the same form as BoxLasso() function of UnitEvents A list with 2 (if n(species) = 1) or 3 elements (if n(species) > 1) \$BL \$BVL \$BOL

require	<i>Install and load a package (if not already loaded)</i>
---------	---

Description

Install and load a package (if not already loaded)

Usage

```
require(x)
```

Arguments

x	The package name
---	------------------

Value

Installs the package if it was not already installed

shift_duplicates	<i>Shift duplicates</i>
------------------	-------------------------

Description

Shift records occuring at the same timestamp on the same camera

Usage

```
shift_duplicates(d, precision = 10^6)
```

Arguments

d	a dataframe with columns row_ID (row id) cameraID (camera identifier) stamp (timestamp, numeric)
precision	precision to consider

Value

input dataframe with modified timestamps.

ue_model_to_df	<i>Model to dataframe</i>
----------------	---------------------------

Description

Transforms the specification of a UnitEvents model to a dataframe.

Usage

```
ue_model_to_df(mod, comp = 1)
```

Arguments

mod	a list of length 2 (and is the named output of the UnitEvent's function coeff2interac): \$S (matrix (n(species), Ncomp)). \$I (array (n(species), n(species), Ncomp)).
comp	the compartment to extract

Value

a dataframe with columns time: the time (support of interaction functions) excitefunc: values of the functions for a corresponding time t from: species "from" which the interaction is to: species "towards" which the interaction is

ue_to_df	<i>Simulation to dataframe</i>
----------	--------------------------------

Description

Convert output of simulation generated with HawkesMulti to a dataframe (rows must be named).

Usage

```
ue_to_df(ue)
```

Arguments

ue	an array (nspecies, ncameras, nmax+1) in the same format as the output of HawkesMulti from the UnitEvents package).
----	---

Value

Returns a dataframe with columns: species camera stamp

ue_to_graph	<i>Model to graph</i>
-------------	-----------------------

Description

Adaptation of plot_graph_Hawkes function in UnitEvents, that returns the igraph object instead of plotting it.

Usage

```
ue_to_graph(IS, neurnames)
```

Arguments

IS	a list of length 2 (the output of the UnitEvent's function coeff2interac): \$S (matrix (n(species), Ncomp)). \$I (array (n(species), n(species), Ncomp)).
neurnames	species_names

Value

a tbl_graph object representing the model. nodes: 1 column: names (with species names). edge: 3 columns: from, to, weight.

unpack_inf	<i>Format the inference output</i>
------------	------------------------------------

Description

Transform output of inference into a list of coefficient matrices.

Usage

```
unpack_inf(inf, species_names = NA, k, delta)
```

Arguments

inf	the output of the infer or BoxLasso. Must be a list with 3 elements: \$BL \$BVL \$BOL
species_names	a list of species names to rename rows and columns (optional)
k	number of bins
delta	binwidth

Value

A list of length 3: \$BL \$BVL \$BOL Each element is a list of length 2 (and is the named output of the UnitEvent's function `coeff2interac`): \$S (matrix (n(species), Ncomp)). The matrix's rows are named with species_names. \$I (array (n(species), n(species), Ncomp)). The array's first and second dimensions are named with species_names.

write_formula	<i>Write formula</i>
---------------	----------------------

Description

Write a formula expected by ppstat for the inference model

Usage

```
write_formula(
  spp,
  startknot_spp = -0.5,
  endknot_spp = 2.5,
  by_spp = 0.25,
  ord = 3,
  hourcov = FALSE,
  ord_hour = 3,
  startknot_hour = -pi/4,
  endknot_hour = 2 * pi + pi/4,
```

```

    by_hour = (3/24) * 2 * pi,
    trunc = NULL
  )

```

Arguments

spp	Vector of unique species names
startknot_spp	First knot for species interactions. It is expressed in the same unit as the stamps on the data to fit the model to (e.g. days).
endknot_spp	Last knot for species interactions. It is expressed in the same unit as the stamps on the data to fit the model to (e.g. days).
by_spp	Step for the species interactions knots. It is expressed in the same unit as the stamps on the data to fit the model to (e.g. days).
ord	Order of the splines of species interactions.
hourcov	Should the final formula include a hour covariate (TRUE or FALSE)?
ord_hour	Order of the splines of the hour covariate for background rates.
startknot_hour	First knot for the hour covariate spline for the background rate (expressed in radians).
endknot_hour	Last knot for the hour covariate spline for the background rate (expressed in radians).
by_hour	Step of the knots for the hour covariate spline for the background rate (expressed in radians).
trunc	Whether to truncate the species interactions. If it is NULL, species interactions splines will not be truncated. Else, it must be a numerical vectors and the spline basis will be truncated to the interval from trunc[1] to trunc[2] (see the documentation of <code>ppstat::bSpline</code>).

Value

A string that can be interpreted as a formula.