

Package: MRFtools (via r-universe)

October 23, 2026

Version 0.0-3

Title Tools for Constructing and Plotting Markov Random Fields in R
for Graphical Data

Maintainer Eric J. Pedersen <eric.j.pedersen@gmail.com>

Depends R (>= 4.1.0)

Imports gratia, stats, generics, assertthat, sf, ape, colorspace,
tidygraph, ggraph, ggplot2, tidyr, dplyr, rlang

Suggests testthat, vdiffr, mgcv, gamm4, sp, quarto, withr

Description Utility functions for using Markov Random Field smooths in
Generalized Additive Models fitted with the 'mgcv' package.

License GPL-3

LazyData true

URL <https://github.com/eric-pedersen/MRFtools>

BugReports <https://github.com/eric-pedersen/MRFtools/issues>

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Encoding UTF-8

Config/testthat/edition 3

VignetteBuilder quarto

Config/pak/sysreqs

libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libgdal-dev gdal-bin libgeos-dev libglpk-
dev libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://gavinsimpson.r-universe.dev>

Date/Publication 2026-03-26 07:37:48 UTC

RemoteUrl <https://github.com/gavinsimpson/MRFtools>

RemoteRef HEAD

RemoteSha 1193d58ed7886fabf4fe0996d798d11679a48e0b

Contents

as.matrix.mrf_penalty	2
as_mrf_penalty	3
get_config	3
get_labels	4
get_model	4
get_mrf	4
get_penalty	5
get_type	6
mrf_config	6
mrf_penalty	7
mrf_penalty.dendrogram	7
mrf_penalty.factor	8
mrf_penalty.hclust	9
mrf_penalty.numeric	9
mrf_penalty.phylo	11
mrf_penalty.sf	12
mrf_penalty.SpatialPolygons	12
mrf_penalty.SpatialPolygonsDataFrame	13
vis	14
visualize.cyclic_mrf_penalty	14
visualize.dendrogram_mrf_penalty	15
visualize.fully_connected_graph_mrf_penalty	16
visualize.sequential_mrf_penalty	17

Index	19
--------------	-----------

as.matrix.mrf_penalty *Convert a MRF penalty object to a matrix*

Description

Convert a MRF penalty object to a matrix

Usage

```
## S3 method for class 'mrf_penalty'
as.matrix(x, ...)
```

Arguments

x	an object inheriting from class "mrf_penalty"
...	arguments passed to other methods

Examples

```
p <- mrf_penalty(1:10)
as.matrix(p)
```

as_mrf_penalty	Create a "mrf_penalty" object from a matrix and a config
----------------	--

Description

Create a "mrf_penalty" object from a matrix and a config

Usage

```
as_mrf_penalty(penalty, config)
```

Arguments

penalty	matrix
config	list

get_config	Extract configuration details of an MRF penalty
------------	---

Description

Extract configuration details of an MRF penalty

Usage

```
get_config(penalty)
```

Arguments

penalty	an object of class "mrf_penalty"
---------	----------------------------------

Value

An object of class "mrf_config", a list.

get_labels	<i>Extract MRF node labels from an MRF penalty</i>
------------	--

Description

Extract MRF node labels from an MRF penalty

Usage

```
get_labels(penalty)
```

Arguments

penalty	an object of class "mrf_penalty"
---------	----------------------------------

get_model	<i>Extract the model type and parameters from an MRF penalty</i>
-----------	--

Description

Extract the model type and parameters from an MRF penalty

Usage

```
get_model(penalty)
```

Arguments

penalty	an object of class "mrf_penalty"
---------	----------------------------------

get_mrf	<i>Extract a fitted MRF</i>
---------	-----------------------------

Description

Extract a fitted MRF

Usage

```
get_mrf(object, ...)

## S3 method for class 'bam'
get_mrf(object, ...)

## S3 method for class 'gamm'
get_mrf(object, ...)

## S3 method for class 'list'
get_mrf(object, ...)

## S3 method for class 'gam'
get_mrf(object, term, ...)
```

Arguments

object	An object from which to extract the fitted MRF. Currently only for objects of classes gam, bam, and gamm, and GAMMs fitted by <code>gamm4::gamm4()</code> .
...	Arguments passed to other methods.
term	character; the MRF term to extract. Can be a partial match to a term, which is matched against the smooth label.

Value

A object representing the fitted MRF

get_penalty	<i>Extract a MRF penalty matrix</i>
-------------	-------------------------------------

Description

Extract a MRF penalty matrix

Usage

```
get_penalty(penalty, ...)

## S3 method for class 'mrf_penalty'
get_penalty(penalty, ...)
```

Arguments

penalty	an R object from which to extract the MRF penalty matrix.
...	arguments passed to other methods.

Value

A penalty matrix of class "matrix".

get_type	<i>Extract the type of MRF from the penalty</i>
----------	---

Description

Extract the type of MRF from the penalty

Usage

```
get_type(object)

## S3 method for class 'mrf_penalty'
get_type(object)

## S3 method for class 'mrf_config'
get_type(object)
```

Arguments

object an object of class "mrf_penalty" or "mrf_config".

Value

A length 1 character vector containing the type of MRF penalty.

mrf_config	<i>MRF penalty configuration data</i>
------------	---------------------------------------

Description

MRF penalty configuration data

Usage

```
mrf_config(
  type = NULL,
  model = NULL,
  params = NULL,
  node_labels = NULL,
  delta = NULL,
  obj = NULL
)
```

Arguments

type	character
model	character
params	list?
node_labels	character
delta	numeric
obj	character

mrf_penalty	<i>Markov Random Field Penalty</i>
-------------	------------------------------------

Description

Markov Random Field Penalty

Usage

```
mrf_penalty(object, ...)
```

Arguments

object	an R object to create the MRF penalty from.
...	arguments passed to other methods.

mrf_penalty.dendrogram	<i>MRF penalty from a dendrogram</i>
------------------------	--------------------------------------

Description

MRF penalty from a dendrogram

Usage

```
## S3 method for class 'dendrogram'  
mrf_penalty(object, model = NULL, node_labels = NULL, delta = FALSE, ...)
```

Arguments

object	an R object to create the MRF penalty from.
model	character; one of "full" or "individual" indicating if a fully connected graph ("full") or a random effect (random intercepts; "individual") penalty is created.
node_labels	character; a vector of alternative labels for the levels of the factor.
delta	numeric or logical; either the numeric value to add to the diagonal of the MRF penalty matrix, or a logical value indicating if such an adjustment should be made. The default is to not alter the diagonal of the penalty matrix.
...	arguments passed to other methods.

mrf_penalty.factor	<i>Fully connected graph and random effect MRF penalties from a factor</i>
--------------------	--

Description

Fully connected graph and random effect MRF penalties from a factor

Usage

```
## S3 method for class 'factor'
mrf_penalty(
  object,
  model = c("full", "individual"),
  node_labels = NULL,
  delta = FALSE,
  ...,
  type
)
```

Arguments

object	an R object to create the MRF penalty from.
model	character; one of "full" or "individual" indicating if a fully connected graph ("full") or a random effect (random intercepts; "individual") penalty is created.
node_labels	character; a vector of alternative labels for the levels of the factor.
delta	numeric or logical; either the numeric value to add to the diagonal of the MRF penalty matrix, or a logical value indicating if such an adjustment should be made. The default is to not alter the diagonal of the penalty matrix.
...	arguments passed to other methods.
type	character; deprecated. Use, model instead.

Examples

```
# a factor
fv <- factor(letters[1:10])

# create the MRF penalty for a fully connected graph
p <- mrf_penalty(fv, model = "full")
p
as.matrix(p)

# create the MRF penalty equivalent of random effects
p <- mrf_penalty(fv, model = "individual")
p
as.matrix(p)
```

mrf_penalty.hclust	<i>MRF penalty from a hclust object</i>
--------------------	---

Description

MRF penalty from a hclust object

Usage

```
## S3 method for class 'hclust'
mrf_penalty(object, ...)
```

Arguments

object	an R object to create the MRF penalty from.
...	arguments passed to other methods.

mrf_penalty.numeric	<i>Continuous-time random walk MRF penalty from a numeric vector</i>
---------------------	--

Description

Models one-dimensional numeric vectors as random-walk models.

Usage

```
## S3 method for class 'numeric'
mrf_penalty(
  object,
  model = c("rw1", "rw2", "ar1", "ou"),
  cyclic = FALSE,
  rho = NULL,
  at_nodes = NULL,
  node_labels = NULL,
  add_missing = NULL,
  end_points = NULL,
  end_dist = NULL,
  delta = FALSE,
  ...,
  type
)
```

Arguments

<code>object</code>	an R object to create the MRF penalty from.
<code>model</code>	character; one of "rw1", "rw2", "ar1", or "ou". See Description for details on the models
<code>cyclic</code>	logical; If TRUE, the end points are treated as neighbouring each other. See Description for details
<code>rho</code>	numeric;
<code>at_nodes</code>	numeric;
<code>node_labels</code>	character; a vector of alternative labels for the levels of the factor.
<code>add_missing</code>	logical;
<code>end_points</code>	numeric; an optional vector of length 2 providing the end points of the period of cycle.
<code>end_dist</code>	numeric;
<code>delta</code>	numeric or logical; either the numeric value to add to the diagonal of the MRF penalty matrix, or a logical value indicating if such an adjustment should be made. The default is to not alter the diagonal of the penalty matrix.
<code>...</code>	arguments passed to other methods.
<code>type</code>	character; deprecated. Use, <code>model</code> instead.

Examples

```
# linear rw1: 1st order continuous-time random walk
p <- mrf_penalty(1:10)
as.matrix(p)

# cyclic rw1:
p <- mrf_penalty(1:10, model = "rw1", cyclic = TRUE)
```

```
as.matrix(p)

# cyclic with user-specified end points
p <- mrf_penalty(1:10, model = "rw1", cyclic = TRUE, end_points = c(0,11))
as.matrix(p)
```

mrf_penalty.phylo	<i>MRF penalty from a phylogeny</i>
-------------------	-------------------------------------

Description

MRF penalty from a phylogeny

Usage

```
## S3 method for class 'phylo'
mrf_penalty(
  object,
  model = c("rw1", "Brownian"),
  node_labels = NULL,
  delta = FALSE,
  eps = 0,
  ...
)
```

Arguments

object	an R object to create the MRF penalty from.
model	character; one of "full" or "individual" indicating if a fully connected graph ("full") or a random effect (random intercepts; "individual") penalty is created.
node_labels	character; a vector of alternative labels for the levels of the factor.
delta	numeric or logical; either the numeric value to add to the diagonal of the MRF penalty matrix, or a logical value indicating if such an adjustment should be made. The default is to not alter the diagonal of the penalty matrix.
eps	A value to add to the variance-covariance matrix diagonal to make it positive definite
...	arguments passed to other methods.

mrf_penalty.sf	<i>MRF penalty from polygon or multi-polygon simple features</i>
----------------	--

Description

MRF penalty from polygon or multi-polygon simple features

Usage

```
## S3 method for class 'sf'
mrf_penalty(
  object,
  model = "icar",
  node_labels = NULL,
  buffer = NULL,
  delta = FALSE,
  ...
)
```

Arguments

object	an R object to create the MRF penalty from.
model	character; one of "full" or "individual" indicating if a fully connected graph ("full") or a random effect (random intercepts; "individual") penalty is created.
node_labels	character; a vector of alternative labels for the levels of the factor.
buffer	numeric; buffer distance for all or for individual elements of the geometry. See argument dist in sf::st_buffer for details.
delta	numeric or logical; either the numeric value to add to the diagonal of the MRF penalty matrix, or a logical value indicating if such an adjustment should be made. The default is to not alter the diagonal of the penalty matrix.
...	arguments passed to other methods.

mrf_penalty.SpatialPolygons	<i>MRF penalty from a SpatialPolygons</i>
-----------------------------	---

Description

MRF penalty from a SpatialPolygons

Usage

```
## S3 method for class 'SpatialPolygons'
mrf_penalty(
  object,
  model = "icar",
  node_labels = NULL,
  buffer = NULL,
  delta = FALSE,
  ...
)
```

Arguments

object	an R object to create the MRF penalty from.
model	character; one of "full" or "individual" indicating if a fully connected graph ("full") or a random effect (random intercepts; "individual") penalty is created.
node_labels	character; a vector of alternative labels for the levels of the factor.
buffer	numeric; buffer distance for all or for individual elements of the geometry. See argument dist in sf::st_buffer for details.
delta	numeric or logical; either the numeric value to add to the diagonal of the MRF penalty matrix, or a logical value indicating if such an adjustment should be made. The default is to not alter the diagonal of the penalty matrix.
...	arguments passed to other methods.

mrf_penalty.SpatialPolygonsDataFrame

MRF penalty from a SpatialPolygonsDataFrame

Description

MRF penalty from a SpatialPolygonsDataFrame

Usage

```
## S3 method for class 'SpatialPolygonsDataFrame'
mrf_penalty(
  object,
  model = "icar",
  node_labels = NULL,
  buffer = NULL,
  delta = FALSE,
  ...
)
```

Arguments

object	an R object to create the MRF penalty from.
model	character; one of "full" or "individual" indicating if a fully connected graph ("full") or a random effect (random intercepts; "individual") penalty is created.
node_labels	character; a vector of alternative labels for the levels of the factor.
buffer	numeric; buffer distance for all or for individual elements of the geometry. See argument dist in sf::st_buffer for details.
delta	numeric or logical; either the numeric value to add to the diagonal of the MRF penalty matrix, or a logical value indicating if such an adjustment should be made. The default is to not alter the diagonal of the penalty matrix.
...	arguments passed to other methods.

vis	<i>Visualize a data set or object</i>
-----	---------------------------------------

Description

Synonyms of the [generics::visualize\(\)](#) method. Alternatives are `vis()` and `visualise()`.

Usage

```
vis(x, ...)
visualise(x, ...)
```

Arguments

x	A data frame or other object.
...	Other arguments passed to methods

visualize.cyclic_mrf_penalty	<i>Visualizing penalty matrix or graph object for a cyclic 1D MRF</i>
------------------------------	---

Description

Visualizing penalty matrix or graph object for a cyclic 1D MRF

Usage

```
## S3 method for class 'cyclic_mrf_penalty'
visualize(x, graph = TRUE, layout = "linear", circular = TRUE, ...)
```

Arguments

x	an object of class "cyclic_mrf_penalty"
graph	logical;
layout	character;
circular	logical;
...	arguments passed to other methods and ultimately on to ggraph::create_layout() if graph = TRUE.

Examples

```
# example code
mrf_penalty(1:10, type = "linear") |>
  visualize()
```

```
visualize.dendrogram_mrf_penalty
      Plot a thing
```

Description

Plot a thing

Usage

```
## S3 method for class 'dendrogram_mrf_penalty'
visualize(
  x,
  graph = TRUE,
  layout = "stress",
  circular = FALSE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  fill_scale = NULL,
  ...
)
```

Arguments

x	an object of class "dendrogram_mrf_penalty"
graph	logical;
layout	character;
circular	logical;

xlab, ylab, title, subtitle, caption	character; labels for plots. If xlab or ylab are not supplied, a suitable default is used.
fill_scale	a suitable fill scale to use if plotting the penalty matrix
...	arguments passed to other methods and ultimately on to ggraph::create_layout() if graph = TRUE.

Examples

```
# example code
hc <- hclust(dist(USArrests), "complete")
mrf_penalty(hc) |>
  visualize()
```

```
visualize.fully_connected_graph_mrf_penalty
      Plot a thing
```

Description

Plot a thing

Usage

```
## S3 method for class 'fully_connected_graph_mrf_penalty'
visualize(
  x,
  graph = TRUE,
  layout = "stress",
  circular = FALSE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  fill_scale = NULL,
  ...
)
```

Arguments

x	an object of class "fully_connected_graph_mrf_penalty"
graph	logical;
layout	character;
circular	logical;

xlab, ylab, title, subtitle, caption	character; labels for plots. If xlab or ylab are not supplied, a suitable default is used.
fill_scale	a suitable fill scale to use if plotting the penalty matrix
...	arguments passed to other methods and ultimately on to ggraph::create_layout() if graph = TRUE.

Examples

```
# example code
mrf_penalty(1:10, type = "linear") |>
  visualize()
```

```
visualize.sequential_mrf_penalty
      Plot a thing
```

Description

Plot a thing

Usage

```
## S3 method for class 'sequential_mrf_penalty'
visualize(
  x,
  graph = TRUE,
  layout = "linear",
  circular = FALSE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  fill_scale = NULL,
  ...
)
```

Arguments

x	an object of class "sequential_mrf_penalty"
graph	logical;
layout	character;
circular	logical;
xlab, ylab, title, subtitle, caption	character; labels for plots. If xlab or ylab are not supplied, a suitable default is used.

`fill_scale` a suitable fill scale to use if plotting the penalty matrix
`...` arguments passed to other methods and ultimately on to [ggraph::create_layout\(\)](#) if `graph = TRUE`.

Examples

```
# example code
mrf_penalty(1:10, type = "linear") |>
  visualize()
```

Index

`as.matrix.mrf_penalty`, 2
`as_mrf_penalty`, 3

`gamm4::gamm4()`, 5
`generics::visualize()`, 14
`get_config`, 3
`get_labels`, 4
`get_model`, 4
`get_mrf`, 4
`get_penalty`, 5
`get_type`, 6
`ggraph::create_layout()`, 15–18

`mrf_config`, 6
`mrf_penalty`, 7
`mrf_penalty.dendrogram`, 7
`mrf_penalty.factor`, 8
`mrf_penalty.hclust`, 9
`mrf_penalty.numeric`, 9
`mrf_penalty.phylo`, 11
`mrf_penalty.sf`, 12
`mrf_penalty.SpatialPolygons`, 12
`mrf_penalty.SpatialPolygonsDataFrame`,
13

`sf::st_buffer`, 12–14

`vis`, 14
`visualise(vis)`, 14
`visualize.cyclic_mrf_penalty`, 14
`visualize.dendrogram_mrf_penalty`, 15
`visualize.fully_connected_graph_mrf_penalty`,
16
`visualize.sequential_mrf_penalty`, 17