Package: morphology (via r-universe)

October 24, 2024

Version 0.0.0.9000

Title Morphological description of 3D categorical arrays

Description The {morphology} package implements a flexible API for the calculation of nearest-neighbour based morphological quantities for one-, two- or three-dimensional categorical arrays.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Depends R (>= 2.10)

Imports tibble, std, mmand, matrixStats, nabor, reshape2, ggplot2, igraph

Suggests knitr, rmarkdown, lobstr

Remotes rogiersbart/std

LazyData true

VignetteBuilder knitr

Repository https://rogiersbart.r-universe.dev

RemoteUrl https://github.com/rogiersbart/morphology

RemoteRef HEAD

RemoteSha 70a0ae3570941c9a46da2dbd9bca431b4d61c6e6

Contents

`describe` *Describe morphology using summary functions*

Description

Describe morphology using summary functions

Usage

```
describe(
  .list,
  what = mean,
  of = "neighbours",
  in_function_of = switch(of, neighbours = "distance", distance = "neighbours",
    pathlength = "distance"),
  at = NULL,
  connected = NULL,
  cumulative = FALSE,
  name = NULL
)
```

Arguments

`.list`	List with the `distance` matrix and `within`, and potentially a `connected` matrix, typically output from `look_for()`. Alternatively, this can be a data frame resulting from a `describe()` including a `name` argument, to add additional described summaries.
`what`	Summary function.
`of`	Quantity to summarise, and return as `y`. Can be `"neighbours"` (default), `"distance"` or `"pathlength"`.
`in_function_of`	Quantity to return as `x`. The default is to use `"distance"` in case `of` is `"neighbours"` or `"pathlength"`, and `"neighbours"` otherwise.
`at`	Distances at which to describe the results. Defaults to the integer sequence from `0L` to the maximum search distance, if available, and otherwise the maximum distance found.
`connected`	Logical. Count connected or disconnected neighbours only. Defaults to `NULL`, for which all neighbours are considered.

cumulative	Logical. Provide cumulative results (distances <= threshold) or binned results (distances at center of provided breaks; default).
name	Name to use for the current described dataset in an extra name column. If provided, the input list is included as attribute to the output, and describe() can be repeated multiple times in a pipechain.

Value

Data frame with distance and neighbours columns. If name is provided, an additional name column is included for identifying different described summaries. In case a name column is there, the object has an extra morphology.list attribute, to enable piping into additional describe() calls.

---

finalise *Remove all morphology object attributes to allow garbage collection*

Description

Remove all morphology object attributes to allow garbage collection

Usage

```
finalise(.x)
```

Arguments

.x	Object to remove the attributes from.

Value

Same object, but without the morphology attributes.

---

gen_fully_penetrable_spheres *Fully penetrable spheres generator*

Description

Fully penetrable spheres generator

Usage

```
gen_fully_penetrable_spheres(
  dimensions = rep(200, 3),
  proportion = 0.5,
  radius = 15
)
```

Arguments

dimensions	Dimensions of the generated array.
proportion	Proportion of the spheres category in the generated array.
radius	Radius of the spheres.

Value

Array with fully penetrable spheres

---

look_at *Define what category to look at*

Description

Define what category to look at

Usage

```
look_at(
  .array,
  what = "voxels",
  of_category,
  in_relation_to = NULL,
  kernel_width = 3,
  kernel_shape = "diamond"
)
```

Arguments

.array	Array to analyze.
what	Either "voxels" (default) or "components" to look at individual voxels or connected components.
of_category	Vector of categories to consider. If negative, these are omitted.
in_relation_to	Vector of categories to consider for cross-category morphology. If negative, these are omitted.
kernel_width	Width of the kernel for determining connectivity. Can be a vector (row, column, layer). Not used when looking at voxels.
kernel_shape	Shape of the kernel for determining connectivity. Can be "diamond" (default), "disc" or "box". Not used when looking at voxels.

Value

Data frame (tibble) with x, y, z and value or component columns.

look_for *Define what neighbours to look for*

Description

Define what neighbours to look for

Usage

```
look_for(.df, neighbours = 1, within = Inf, from_border = FALSE, error = 0)
```

Arguments

.df	Data frame with x, y, and/or z, and value or component columns, typically output from look_at() or look_in().
neighbours	Number of nearest neighbours to look for. Can be Inf if within is finite, in which case all neighbours within that distance are looked for.
within	Maximum search distance.
from_border	Look for neighbours of border voxels. Defaults to TRUE. If FALSE, minus sampling is performed.
error	Approximate error bound, for approximate nearest neighbour search.

Value

List with the distance matrix and xyz data frame. When looking at components, additionally a connected matrix.

look_in *Define what direction to look in*

Description

Define what direction to look in

Usage

```
look_in(.df, direction = "xyz", every = 1)
```

Arguments

`.df`	Data frame with `x`, `y`, `z` and `value` or `component` columns, typically output from `look_at()`.
`direction`	Direction to consider. Either 1D (`"x"`, `"y"`, `"z"`), 2D (`"xy"`, `"xz"`, `"yz"`) or 3D (`"xyz"`).
`every`	Value indicating how many voxels to consider, e.g. one of two (`1/2`), one every three (`1/3`), etc. Defaults to every voxel (`1/1`). This can be a vector (row, column, layer) to thin the different dimensions differently. Alternatively it can be a list of two vectors, where the first one is used for the category of interest (`of_category` in `look_at()`) and the second one is for the neighbour category (`in_relation_to` in `look_at()`, or again `of_category` for a self-query).

Value

Data frame (tibble) with `x`, `y`, and/or `z`, and `value` or `component` columns.

`q05` *Calculate the 0.05 quantile*

Description

Calculate the 0.05 quantile

Usage

```
q05(...)
```

Arguments

`...`	Arguments passed to the `quantile()` function.

Value

Sample quantile.

q25 *Calculate the 0.25 quantile*

Description

Calculate the 0.25 quantile

Usage

```
q25(...)
```

Arguments

...	Arguments passed to the `quantile()` function.

Value

Sample quantile.

q50 *Calculate the 0.50 quantile*

Description

Calculate the 0.50 quantile

Usage

```
q50(...)
```

Arguments

...	Arguments passed to the `quantile()` function.

Value

Sample quantile.

q75 *Calculate the 0.75 quantile*

Description

Calculate the 0.75 quantile

Usage

```
q75(...)
```

Arguments

...	Arguments passed to the `quantile()` function.

Value

Sample quantile.

q95 *Calculate the 0.95 quantile*

Description

Calculate the 0.95 quantile

Usage

```
q95(...)
```

Arguments

...	Arguments passed to the `quantile()` function.

Value

Sample quantile.

scale_by *Scale the morphological description*

Description

Scale the morphological description

Usage

```
scale_by(df, ...)
```

Arguments

df	Data frame
...	Things to scale the description by. Can be "neighbourhood", which automatically chooses between alternative options "volume", "area" or "length", or "proportion", "inverse proportion", another description data frame with the same distances, ...

Value

Data frame with scaled neighbours column.

visualise *Visualise the morphological description*

Description

Visualise the morphological description

Usage

```
visualise(df, ...)
```

Arguments

df	Data frame resulting from a describe() call, or just a numeric vector.
...	Arguments passed to ggplot2::labs() for labelling.

Value

A ggplot2 plot.

Index