

# Package ‘neuromapr’

February 27, 2026

**Title** Spatial Null Models and Transforms for Brain Map Comparison

**Version** 0.2.1

**Description** Implements spatial null models and coordinate-space transformations for statistical comparison of brain maps, following the framework described in Markello et al. (2022) [doi:10.1038/s41592-022-01625-w](https://doi.org/10.1038/s41592-022-01625-w). Provides variogram-matching surrogates (Burt et al. 2020), Moran spectral randomization (Wagner & Dray 2015), and spin-based permutation tests (Alexander-Bloch et al. 2018). Includes an R interface to the 'neuromaps' annotation registry for browsing, downloading, and comparing brain map annotations from the Open Science Framework ('OSF'). Integrates with 'ciftiTools' for coordinate-space transforms.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Depends** R (>= 4.2.0)

**Imports** cli, dplyr, ggplot2, gifti, httr2, igraph, lifecycle, rlang, stats, tibble, tools, withr

**Suggests** ciftiTools, clue, freesurferformats, jsonlite, knitr, rmarkdown, RNifti, RSpectra, testthat (>= 3.0.0), vcr (>= 1.0.0)

**VignetteBuilder** knitr

**URL** <https://github.com/lcbc-uio/neuromapr>,  
<https://lcbc-uio.github.io/neuromapr/>,  
<https://doi.org/10.5281/zenodo.18648937>

**BugReports** <https://github.com/lcbc-uio/neuromapr/issues>

**NeedsCompilation** no

**Author** Athanasia Mo Mowinckel [aut, cre, cph] (ORCID:  
<https://orcid.org/0000-0002-5756-0223>)

**Maintainer** Athanasia Mo Mowinckel <a.m.mowinckel@psykologi.uio.no>

**Repository** CRAN

**Date/Publication** 2026-02-27 16:20:02 UTC

## Contents

annot_to_gifti . . . . .	2
check_wb_command . . . . .	3
clear_neuromaps_cache . . . . .	4
compare_maps . . . . .	4
fetch_neuromaps_annotation . . . . .	5
fsmorph_to_gifti . . . . .	6
generate_nulls . . . . .	7
get_parcel_centroids . . . . .	8
get_surface_distance . . . . .	9
make_surf_graph . . . . .	10
neuromaps_available . . . . .	11
null_alexander_bloch . . . . .	12
null_baum . . . . .	13
null_burt2018 . . . . .	14
null_burt2020 . . . . .	14
null_cornblath . . . . .	16
null_distribution . . . . .	17
null_moran . . . . .	18
null_spin_vasa . . . . .	19
parcellate . . . . .	20
parcels_to_vertices . . . . .	21
permtest_metric . . . . .	21
read_brain_map_values . . . . .	23
resample_images . . . . .	23
transform_to_space . . . . .	25
unparcellate . . . . .	26
vertex_areas . . . . .	27
vertices_to_parcels . . . . .	27

**Index** **29**

---

annot_to_gifti	<i>Convert FreeSurfer annotation to GIFTI</i>
----------------	---

---

### Description

Reads a FreeSurfer .annot file and writes a GIFTI label file.

### Usage

```
annot_to_gifti(annot_path, output_path = NULL)
```

**Arguments**

annot\_path      Path to FreeSurfer .annot file.  
output\_path     Output GIFTI path. If NULL, replaces the extension with .label.gii.

**Value**

The output file path (invisibly).

**References**

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

**Examples**

```
## Not run:  
annot_to_gifti("lh.aparc.annot")  
annot_to_gifti("lh.aparc.annot", "lh.aparc.label.gii")  
  
## End(Not run)
```

---

check\_wb\_command      *Check for Connectome Workbench*

---

**Description**

Verifies that wb\_command is available. If wb\_path is NULL, checks ciftiTools default and system PATH.

**Usage**

```
check_wb_command(wb_path = NULL)
```

**Arguments**

wb\_path            Optional explicit path to wb\_command.

**Value**

Path to wb\_command executable.

**Examples**

```
## Not run:  
check_wb_command()  
  
## End(Not run)
```

---

`clear_neuromaps_cache` *Clear cached neuromaps registry data*

---

### Description

Removes the session-level cache of the neuromaps annotation registry, forcing a fresh download on the next call to `neuromaps_available()` or `fetch_neuromaps_annotation()`.

### Usage

```
clear_neuromaps_cache()
```

### Value

NULL, invisibly.

### Examples

```
clear_neuromaps_cache()
```

---

`compare_maps`

*Compare brain maps with spatial null model significance testing*

---

### Description

Computes the correlation between two brain maps and optionally tests significance using a spatial null model to account for spatial autocorrelation.

### Usage

```
compare_maps(
  x,
  y,
  method = c("pearson", "spearman"),
  null_method = NULL,
  n_perm = 1000L,
  nulls = NULL,
  distmat = NULL,
  coords = NULL,
  seed = NULL,
  na.rm = TRUE,
  verbose = TRUE,
  ...
)

## S3 method for class 'neuromaps_enhanced_comparison'
print(x, ...)
```

## Arguments

x, y	Numeric vectors of brain map values, or file paths to GIFTI/NIfTI files.
method	Correlation method: "pearson" or "spearman".
null_method	Optional null model method for empirical p-values. One of "burt2020", "moran", "spin_vasa", "spin_hungarian", "alexander_bloch", "baum", "cornblath", "burt2018", or NULL for parametric only.
n_perm	Integer number of null permutations.
nulls	Pre-computed <a href="#">null_distribution</a> object for x.
distmat	Distance matrix (passed to null model if needed).
coords	Coordinate list (passed to spin null models if needed).
seed	Optional integer seed for reproducibility.
na.rm	Logical, remove NA values before computing correlation.
verbose	Logical, print progress messages.
...	Additional arguments passed to <a href="#">generate_nulls()</a> .

## Value

A `neuromaps_enhanced_comparison` object (inherits `neuromaps_comparison`) with additional fields `p_null`, `null_method`, `null_r`, and `n_perm`.

## References

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

## Examples

```
x <- rnorm(50)
y <- x + rnorm(50)
compare_maps(x, y, verbose = FALSE)
```

---

fetch\_neuromaps\_annotation

*Download a neuromaps annotation*

---

## Description

Download brain map annotation files from the neuromaps OSF repository. Surface annotations return two files (left and right hemispheres), volume annotations return one.

**Usage**

```

fetch_neuromaps_annotation(
  source,
  desc,
  space,
  density = NULL,
  resolution = NULL,
  hemisphere = NULL,
  data_dir = neuromaps_cache_dir(),
  overwrite = FALSE,
  verbose = TRUE
)

```

**Arguments**

source	Data source identifier (e.g. "abagen", "beliveau").
desc	Map descriptor key (e.g. "genepc1", "feobv").
space	Coordinate space (e.g. "fsaverage", "MNI152", "fsLR").
density	Surface vertex density (e.g. "10k", "164k"). Mutually exclusive with resolution.
resolution	Volume voxel resolution (e.g. "1mm", "2mm"). Mutually exclusive with density.
hemisphere	Hemisphere ("L" or "R").
data_dir	Directory for cached downloads. Defaults to neuromaps_cache_dir().
overwrite	Re-download even if cached file exists.
verbose	Print progress messages.

**Value**

Character vector of downloaded file path(s).

**Examples**

```

fetch_neuromaps_annotation("abagen", "genepc1", "fsaverage", density = "10k")

```

---

fsmorph\_to\_gifti

*Convert FreeSurfer morphometry to GIFTI*


---

**Description**

Reads a FreeSurfer morphometry file (.curv, .thickness, .sulc) and writes a GIFTI func file.

**Usage**

```

fsmorph_to_gifti(morph_path, output_path = NULL)

```

**Arguments**

morph\_path      Path to FreeSurfer morphometry file.  
output\_path      Output GIFTI path. If NULL, replaces the extension with .func.gii.

**Value**

The output file path (invisibly).

**References**

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

**Examples**

```
## Not run:  
fsmorph_to_gifti("lh.thickness")  
fsmorph_to_gifti("lh.thickness", "lh.thickness.func.gii")  
  
## End(Not run)
```

---

generate_nulls	<i>Generate null distributions for brain map data</i>
----------------	---

---

**Description**

Dispatches to the appropriate null model method for generating spatially-constrained surrogate brain maps.

**Usage**

```
generate_nulls(  
  data,  
  method = c("burt2020", "moran", "spin_vasa", "spin_hungarian", "alexander_bloch",  
             "baum", "cornblath", "burt2018"),  
  n_perm = 1000L,  
  distmat = NULL,  
  coords = NULL,  
  parcellation = NULL,  
  seed = NULL,  
  ...  
)
```

**Arguments**

data	Numeric vector of brain map values.
method	Character string specifying the null model method.
n_perm	Integer number of null permutations to generate.
distmat	Distance matrix (required for "burt2020", "burt2018", and "moran").
coords	List with \$lh and \$rh coordinate matrices (required for spin methods).
parcellation	Integer vector of parcel labels (required for "baum" and "cornblath").
seed	Optional integer seed for reproducibility.
...	Additional arguments passed to the specific null method (e.g. rotation for spin methods, kernel for moran/burt2020).

**Value**

A [null\\_distribution](#) object.

**References**

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

**Examples**

```
data <- rnorm(100)
distmat <- as.matrix(dist(seq_len(100)))
nd <- generate_nulls(data, method = "moran", distmat = distmat, n_perm = 10L)
```

---

get\_parcel\_centroids    *Compute parcel centroids*

---

**Description**

Finds the centroid of each parcel using one of three methods.

**Usage**

```
get_parcel_centroids(
  vertices,
  labels,
  method = c("average", "surface", "geodesic"),
  faces = NULL
)
```

**Arguments**

vertices	Numeric matrix (n x 3) of vertex coordinates.
labels	Integer vector of parcel labels. 0 and NA are medial wall.
method	Centroid method: "average" (coordinate means), "surface" (vertex closest to the average centroid), or "geodesic" (vertex minimizing sum of geodesic distances within parcel).
faces	Integer matrix (m x 3) of face indices. Required for "geodesic" method.

**Value**

Numeric matrix (n\_parcel x 3) with rownames set to parcel labels.

**References**

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

**Examples**

```
vertices <- matrix(rnorm(30), ncol = 3)
labels <- c(1L, 1L, 1L, 2L, 2L, 2L, 3L, 3L, 3L)
get_parcel_centroids(vertices, labels, method = "average")
```

---

get\_surface\_distance *Compute geodesic distances on a surface mesh*

---

**Description**

Builds a graph from a triangular mesh and computes shortest-path (Dijkstra) distances between vertices.

**Usage**

```
get_surface_distance(vertices, faces, source_vertices = NULL)
```

**Arguments**

vertices	Numeric matrix (n x 3) of vertex coordinates.
faces	Integer matrix (m x 3) of face indices (1-indexed).
source_vertices	Optional integer vector of source vertex indices. If NULL, computes the full n x n distance matrix.

**Value**

Numeric distance matrix. If source\_vertices is provided, returns a length(source\_vertices) x n matrix; otherwise an n x n matrix.

## References

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

## Examples

```
vertices <- matrix(  
  c(0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1),  
  nrow = 4, byrow = TRUE  
)  
faces <- matrix(c(1L, 2L, 3L, 2L, 3L, 4L), nrow = 2, byrow = TRUE)  
get_surface_distance(vertices, faces)
```

---

make\_surf\_graph

*Build an igraph from a triangular surface mesh*

---

## Description

Extracts unique edges from triangular faces, computes Euclidean edge weights, and returns an igraph graph object suitable for geodesic distance computation.

## Usage

```
make_surf_graph(vertices, faces)
```

## Arguments

vertices        Numeric matrix (n x 3) of vertex coordinates.  
faces            Integer matrix (m x 3) of face indices (1-indexed).

## Value

An igraph graph object with weighted edges.

## References

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

## Examples

```
vertices <- matrix(  
  c(0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1),  
  nrow = 4, byrow = TRUE  
)  
faces <- matrix(c(1L, 2L, 3L, 2L, 3L, 4L), nrow = 2, byrow = TRUE)  
g <- make_surf_graph(vertices, faces)
```

---

neuromaps\_available *List available neuromaps annotations*

---

### Description

Query the neuromaps registry to see which brain map annotations are available for download. Data is fetched from the [neuromaps](#) project's GitHub repository and cached for the session.

### Usage

```
neuromaps_available(
  source = NULL,
  desc = NULL,
  space = NULL,
  density = NULL,
  resolution = NULL,
  hemisphere = NULL,
  tags = NULL,
  format = NULL,
  refresh = FALSE,
  fixed = FALSE
)
```

### Arguments

source	Data source identifier (e.g. "abagen", "beliveau").
desc	Map descriptor key (e.g. "genepc1", "feobv").
space	Coordinate space (e.g. "fsaverage", "MNI152", "fsLR").
density	Surface vertex density (e.g. "10k", "164k"). Mutually exclusive with resolution.
resolution	Volume voxel resolution (e.g. "1mm", "2mm"). Mutually exclusive with density.
hemisphere	Hemisphere ("L" or "R").
tags	Character vector of tags. All must match (AND logic).
format	Filter by format ("surface" or "volume").
refresh	Logical. If TRUE, forces a fresh download of the registry data, ignoring any session cache.
fixed	Logical. If TRUE, filter strings are matched literally rather than as regular expressions.

### Details

All string filter parameters (source, desc, space, density, resolution, hemisphere, format) are treated as **R regular expressions** and matched with `grepl()`. For example, `source = "^beliveau$"` matches exactly, while `source = "bel"` matches any source containing "bel". Set `fixed = TRUE` for literal string matching. The `tags` parameter always uses exact matching (AND logic).

When used in `fetch_neuromaps_annotation()`, source, desc, and space are exact matches.

**Value**

A tibble of available annotations with columns: source, desc, space, den, res, hemi, format, fname, full\_desc, tags, N, age.

**Examples**

```
neuromaps_available()
neuromaps_available(source = "beliveau")
neuromaps_available(tags = "pet")
```

---

null\_alexander\_bloch *Alexander-Bloch spin test null model*

---

**Description**

Original vertex-level spin test: rotates coordinates and assigns each rotated vertex the value of its nearest original vertex (no optimal matching).

**Usage**

```
null_alexander_bloch(
  data,
  coords,
  n_perm = 1000L,
  seed = NULL,
  rotation = c("euler", "rodrigues")
)
```

**Arguments**

data	Numeric vector of brain map values.
coords	List with \$lh and \$rh matrices of spherical coordinates (n x 3 each).
n_perm	Integer number of null permutations to generate.
seed	Optional integer seed for reproducibility.
rotation	Rotation generation method: "euler" (ZYZ Euler angles, default, matches neuromaps Python) or "rodrigues" (Rodrigues axis-angle formula).

**Value**

A [null\\_distribution](#) object.

**References**

Alexander-Bloch AF et al. (2018) NeuroImage 175:111-120. doi:10.1016/j.neuroimage.2018.04.023

**Examples**

```
coords <- list(lh = matrix(rnorm(30), 10, 3), rh = matrix(rnorm(30), 10, 3))
data <- rnorm(20)
nd <- null_alexander_bloch(data, coords, n_perm = 10L, seed = 1L)
```

---

null\_baum

*Baum spin test null model*


---

**Description**

Spin-based null model with maximum-overlap parcel reassignment. After rotating vertex coordinates, each original parcel is assigned the value of the rotated parcel with the most vertex overlap.

**Usage**

```
null_baum(
  data,
  coords,
  parcellation,
  n_perm = 1000L,
  seed = NULL,
  rotation = c("euler", "rodrigues")
)
```

**Arguments**

data	Numeric vector of brain map values.
coords	List with \$lh and \$rh matrices of spherical coordinates (n x 3 each).
parcellation	Integer vector of parcel labels for all vertices. 0 and NA indicate medial wall.
n_perm	Integer number of null permutations to generate.
seed	Optional integer seed for reproducibility.
rotation	Rotation generation method: "euler" (ZYZ Euler angles, default, matches neuromaps Python) or "rodrigues" (Rodrigues axis-angle formula).

**Value**

A [null\\_distribution](#) object.

**References**

Baum GL et al. (2020) PNAS 117:21854-21861. doi:10.1073/pnas.2005518117

**Examples**

```
coords <- list(lh = matrix(rnorm(30), 10, 3), rh = matrix(rnorm(30), 10, 3))
parcellation <- c(rep(1L, 5), rep(2L, 5), rep(3L, 5), rep(4L, 5))
data <- c(1.0, 2.0, 3.0, 4.0)
nd <- null_baum(data, coords, parcellation, n_perm = 10L, seed = 1L)
```

---

null_burt2018	<i>Burt 2018 spatial autoregressive null model</i>
---------------	--

---

**Description**

Generates surrogate brain maps using a spatial autoregressive (SAR) model. Estimates spatial auto-correlation and distance decay parameters, then generates surrogates by solving the SAR equation and rank-matching to the original data.

**Usage**

```
null_burt2018(data, distmat, n_perm = 1000L, seed = NULL)
```

**Arguments**

data	Numeric vector of brain map values.
distmat	Distance matrix between parcels/vertices.
n_perm	Integer number of null permutations to generate.
seed	Optional integer seed for reproducibility.

**Value**

A [null\\_distribution](#) object.

**References**

Burt JB et al. (2018) Nature Neuroscience 21:1251-1259. doi:10.1038/s41593-018-0195-0

**Examples**

```
data <- rnorm(50)
distmat <- as.matrix(dist(matrix(rnorm(100), 50, 2)))
nd <- null_burt2018(data, distmat, n_perm = 10L, seed = 1L)
```

---

null_burt2020	<i>Variogram-matching null model</i>
---------------	--------------------------------------

---

**Description**

Generates spatially-constrained surrogate brain maps by matching the empirical variogram of the original data through smoothed random permutations.

**Usage**

```

null_burt2020(
  data,
  distmat,
  n_perm = 1000L,
  seed = NULL,
  ns = 500L,
  nh = 25L,
  pv = 25,
  knn = 1000L,
  deltas = seq(0.1, 0.9, by = 0.1),
  kernel = c("exponential", "gaussian", "uniform"),
  resample = FALSE
)

```

**Arguments**

data	Numeric vector of brain map values.
distmat	Distance matrix between parcels/vertices.
n_perm	Integer number of null permutations to generate.
seed	Optional integer seed for reproducibility.
ns	Integer, subsample size for variogram computation.
nh	Integer, number of distance bins for variogram.
pv	Numeric, percentile cutoff for maximum distance in variogram.
knn	Integer, number of nearest neighbors for smoothing.
deltas	Numeric vector of smoothing levels (fractions of knn).
kernel	Smoothing kernel function.
resample	Logical. If FALSE (default, matches brainsmash Python), variogram subsample indices are computed once and reused across all permutations. If TRUE, a fresh subsample is drawn each time.

**Value**

A [null\\_distribution](#) object.

**References**

Burt JB et al. (2020) NeuroImage 220:117038. doi:10.1016/j.neuroimage.2020.117038

**Examples**

```

data <- rnorm(50)
distmat <- as.matrix(dist(matrix(rnorm(100), 50, 2)))
nd <- null_burt2020(data, distmat, n_perm = 10L, seed = 1L)

```

---

null_cornblath	<i>Cornblath spin test null model</i>
----------------	---------------------------------------

---

### Description

Spin-based null model where each rotated vertex receives the label of its nearest non-medial-wall original vertex, then parcels are reassigned by majority vote among the resulting vertex labels.

### Usage

```

null_cornblath(
  data,
  coords,
  parcellation,
  n_perm = 1000L,
  seed = NULL,
  rotation = c("euler", "rodrigues")
)

```

### Arguments

data	Numeric vector of brain map values.
coords	List with \$lh and \$rh matrices of spherical coordinates (n x 3 each).
parcellation	Integer vector of parcel labels for all vertices. 0 and NA indicate medial wall.
n_perm	Integer number of null permutations to generate.
seed	Optional integer seed for reproducibility.
rotation	Rotation generation method: "euler" (ZYZ Euler angles, default, matches neuromaps Python) or "rodrigues" (Rodrigues axis-angle formula).

### Value

A [null\\_distribution](#) object.

### References

Cornblath EJ et al. (2020) Communications Biology 3:590. doi:10.1038/s42003-020-01296-5

### Examples

```

coords <- list(lh = matrix(rnorm(30), 10, 3), rh = matrix(rnorm(30), 10, 3))
parcellation <- c(rep(1L, 5), rep(2L, 5), rep(3L, 5), rep(4L, 5))
data <- c(1.0, 2.0, 3.0, 4.0)
nd <- null_cornblath(data, coords, parcellation, n_perm = 10L, seed = 1L)

```

---

null\_distribution      *Create a null distribution object*

---

## Description

Create a null distribution object

## Usage

```
new_null_distribution(nulls, method, observed, params = list())

## S3 method for class 'null_distribution'
print(x, ...)

## S3 method for class 'null_distribution'
summary(object, ...)

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

## S3 method for class 'null_distribution'
plot(x, parcel = 1L, ...)
```

## Arguments

nulls	Numeric matrix (n x n_perm) of surrogate values.
method	Character string identifying the null model method.
observed	Numeric vector of original data values.
params	Named list of algorithm parameters.
x	A null_distribution object.
...	Ignored.
object	A null_distribution object.
parcel	Integer index of the parcel to plot.

## Value

A null\_distribution object.

## Examples

```
nulls <- matrix(rnorm(30), nrow = 3, ncol = 10)
nd <- new_null_distribution(nulls, "test", observed = c(1, 2, 3))
print(nd)
summary(nd)
```

---

 null\_moran

*Moran spectral randomization null model*


---

### Description

Generates spatially-constrained surrogate brain maps using Moran's eigenvector maps (MEMs) for spectral randomization.

### Usage

```

null_moran(
  data,
  distmat,
  n_perm = 1000L,
  seed = NULL,
  procedure = c("pair", "singleton"),
  kernel = c("inverse_distance", "exponential", "gaussian", "bisquare"),
  tol = 1e-06
)

```

### Arguments

data	Numeric vector of brain map values.
distmat	Distance matrix between parcels/vertices.
n_perm	Integer number of null permutations to generate.
seed	Optional integer seed for reproducibility.
procedure	Character, either "pair" (default, random 2D rotations of near-degenerate eigenvector pairs, matches neuromaps Python) or "singleton" (random sign flips of individual eigenvectors).
kernel	Weight matrix kernel: "inverse_distance" (default, matches neuromaps Python), "exponential", "gaussian", or "bisquare".
tol	Numeric tolerance for eigenvalue comparison.

### Value

A [null\\_distribution](#) object.

### References

Wagner HH, Dray S (2015) Methods in Ecology and Evolution 6:1169-1178. doi:10.1111/2041-210X.12407

### Examples

```

data <- rnorm(50)
distmat <- as.matrix(dist(matrix(rnorm(100), 50, 2)))
nd <- null_moran(data, distmat, n_perm = 10L, seed = 1L)

```

---

null_spin_vasa	<i>Spin-test null models for brain maps</i>
----------------	---

---

### Description

Generate spatially-constrained null distributions using spin-based permutation of spherical coordinates.

### Usage

```
null_spin_vasa(  
  data,  
  coords,  
  n_perm = 1000L,  
  seed = NULL,  
  rotation = c("euler", "rodrigues")  
)
```

```
null_spin_hungarian(  
  data,  
  coords,  
  n_perm = 1000L,  
  seed = NULL,  
  rotation = c("euler", "rodrigues")  
)
```

### Arguments

data	Numeric vector of brain map values.
coords	List with \$lh and \$rh matrices of spherical coordinates (n x 3 each).
n_perm	Integer number of null permutations to generate.
seed	Optional integer seed for reproducibility.
rotation	Rotation generation method: "euler" (ZYZ Euler angles, default, matches neuromaps Python) or "rodrigues" (Rodrigues axis-angle formula).

### Value

A [null\\_distribution](#) object.

### References

Alexander-Bloch AF et al. (2018) NeuroImage 175:111-120. doi:10.1016/j.neuroimage.2018.04.023  
Vasa F et al. (2018) Cerebral Cortex 28:3293-3303. doi:10.1093/cercor/bhx195  
Markello RD, Misisic B (2021) NeuroImage 236:118052. doi:10.1016/j.neuroimage.2021.118052

**Examples**

```
coords <- list(lh = matrix(rnorm(30), 10, 3), rh = matrix(rnorm(30), 10, 3))
data <- rnorm(20)
nd <- null_spin_vasa(data, coords, n_perm = 10L, seed = 1L)
```

---

parcellate

*Parcellate brain map data*

---

**Description**

High-level wrapper that reads data and parcellation from file paths or vectors, then aggregates vertices into parcels.

**Usage**

```
parcellate(data, parcellation, summary_func = mean)
```

**Arguments**

data	Numeric vector or file path to a GIFTI/NIfTI brain map.
parcellation	Integer vector of labels or file path to a GIFTI label file.
summary_func	Function to summarise each parcel (default: <a href="#">mean</a> ).

**Value**

Named numeric vector of parcel-level values.

**References**

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

**Examples**

```
data <- c(1.0, 2.0, 3.0, 4.0)
labels <- c(1L, 1L, 2L, 2L)
parcellate(data, labels)
```

---

parcels\_to\_vertices     *Map parcel data back to vertices*

---

### Description

Expands parcel-level values to a vertex-level vector using parcellation labels.

### Usage

```
parcels_to_vertices(parcel_data, labels, fill = NA_real_)
```

### Arguments

parcel_data	Named numeric vector of parcel values (names match labels).
labels	Integer vector of parcel labels. 0 and NA are medial wall.
fill	Value for medial wall vertices (default: NA_real_).

### Value

Numeric vector of length(length(labels)).

### References

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

### Examples

```
parcel_data <- c("1" = 10, "2" = 20)
labels <- c(1L, 1L, 2L, 2L, 0L)
parcels_to_vertices(parcel_data, labels)
```

---

permtest\_metric     *Permutation test for any metric between brain maps*

---

### Description

Computes a user-specified metric between two vectors and tests significance using either spatially-constrained null surrogates or simple random permutation.

**Usage**

```
permtest_metric(
  x,
  y,
  metric_func = stats::cor,
  n_perm = 1000L,
  seed = NULL,
  null_method = NULL,
  distmat = NULL,
  coords = NULL,
  parcellation = NULL,
  ...
)
```

**Arguments**

<code>x, y</code>	Numeric vectors.
<code>metric_func</code>	Function taking ( <code>x</code> , <code>y</code> ) and returning a scalar.
<code>n_perm</code>	Integer number of permutations.
<code>seed</code>	Optional integer seed for reproducibility.
<code>null_method</code>	Optional null model method passed to <a href="#">generate_nulls()</a> . If NULL, uses simple random permutation.
<code>distmat</code>	Distance matrix (passed to <a href="#">generate_nulls()</a> if needed).
<code>coords</code>	Coordinate list (passed to <a href="#">generate_nulls()</a> if needed).
<code>parcellation</code>	Integer vector (passed to <a href="#">generate_nulls()</a> if needed).
<code>...</code>	Additional arguments passed to <a href="#">generate_nulls()</a> .

**Value**

List with `$observed`, `$null_values`, `$p_value`, and `$n_perm`.

**References**

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

**Examples**

```
x <- rnorm(100)
y <- x + rnorm(100)
result <- permtest_metric(x, y, n_perm = 99L, seed = 1L)
result$observed
result$p_value
```

---

read\_brain\_map\_values *Read vertex/voxel values from a brain map file*

---

### Description

Reads GIFTI (.func.gii) or NIfTI (.nii.gz) files and returns the values as a numeric vector. Used internally by [compare\\_maps\(\)](#) when file paths are passed instead of numeric vectors.

### Usage

```
read_brain_map_values(path)
```

### Arguments

path                    Path to a .func.gii (GIFTI) or .nii.gz (NIfTI) file.

### Value

A numeric vector of map values.

### Examples

```
## Not run:
read_brain_map_values("cortical_thickness.func.gii")
read_brain_map_values("brain_volume.nii.gz")

## End(Not run)
```

---

resample\_images            *Resample brain maps for comparison*

---

### Description

Aligns two brain maps into the same coordinate space and density before comparison. Supports multiple strategies for choosing the target space.

### Usage

```
resample_images(
  src,
  trg,
  src_space = c("fsaverage", "fsLR"),
  trg_space = c("fsaverage", "fsLR"),
  strategy = c("downsample_only", "transform_to_src", "transform_to_trg",
    "transform_to_alt"),
  alt_space = NULL,
```

```

    alt_density = NULL,
    hemisphere = c("left", "right"),
    area_surf_current = NULL,
    area_surf_new = NULL,
    wb_path = NULL,
    verbose = TRUE
)

```

### Arguments

src	Character, file path to the source GIFTI.
trg	Character, file path to the target GIFTI.
src_space	Source coordinate space ("fsaverage" or "fsLR").
trg_space	Target coordinate space ("fsaverage" or "fsLR").
strategy	Resampling strategy. One of "downsample_only", "transform_to_src", "transform_to_trg", or "transform_to_alt".
alt_space	Alternative space for "transform_to_alt" strategy.
alt_density	Alternative density for "transform_to_alt" strategy.
hemisphere	Which hemispheres: "left", "right", or both.
area_surf_current	Path to a current-resolution area-correction surface (e.g. midthickness). Passed to <code>transform_to_space()</code> .
area_surf_new	Path to a target-resolution area-correction surface. Passed to <code>transform_to_space()</code> .
wb_path	Path to wb_command executable.
verbose	Logical, print progress messages.

### Value

List with \$src and \$trg file paths in the aligned space.

### References

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

### Examples

```

## Not run:
resample_images("src.func.gii", "trg.func.gii",
  src_space = "fsaverage", trg_space = "fsaverage"
)

## End(Not run)

```

---

transform\_to\_space      *Transform brain maps between coordinate spaces*

---

### Description

Resamples GIFTI surface files between fsaverage and fsLR coordinate spaces using Connectome Workbench via ciftiTools.

### Usage

```
transform_to_space(
    paths,
    target_space = c("fsLR", "fsaverage"),
    target_density = "32k",
    hemisphere = c("left", "right"),
    method = c("adaptive", "barycentric"),
    area_surf_current = NULL,
    area_surf_new = NULL,
    wb_path = NULL,
    verbose = TRUE
)
```

### Arguments

paths	Character vector of GIFTI file paths to transform.
target_space	Target coordinate space: "fsLR" or "fsaverage".
target_density	Target mesh density (e.g., "32k", "164k").
hemisphere	Which hemispheres to transform: "left", "right", or both.
method	Resampling method: "adaptive" (default) or "barycentric".
area_surf_current	Path to the current-resolution area-correction surface (e.g. midthickness). Recommended when method = "adaptive" for proper vertex-area correction (matches neuromaps Python behaviour).
area_surf_new	Path to the target-resolution area-correction surface. If NULL and area_surf_current is provided, ciftiTools resamples it barycentrically.
wb_path	Path to wb_command executable. If NULL, auto-detected via ciftiTools.
verbose	Logical, print progress messages.

### Value

Character vector of transformed file paths.

### References

Robinson EC et al. (2014) NeuroImage 100:414-426. doi:10.1016/j.neuroimage.2014.07.033  
 Robinson EC et al. (2018) NeuroImage 167:150-165. doi:10.1016/j.neuroimage.2017.10.037

## Examples

```
## Not run:  
transform_to_space("map.func.gii", target_space = "fsLR")  
  
## End(Not run)
```

---

unparcellate

*Unparcellate brain map data*

---

## Description

Inverse of `parcellate()`: maps parcel-level values back to vertices.

## Usage

```
unparcellate(parcel_data, parcellation, fill = NA_real_)
```

## Arguments

<code>parcel_data</code>	Named numeric vector of parcel values.
<code>parcellation</code>	Integer vector of labels or file path to a GIFTI label file.
<code>fill</code>	Value for medial wall vertices (default: <code>NA_real_</code> ).

## Value

Numeric vector of vertex-level values.

## References

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

## Examples

```
parcel_data <- c("1" = 10, "2" = 20)  
labels <- c(1L, 1L, 2L, 2L, 0L)  
unparcellate(parcel_data, labels)
```

---

vertex_areas	<i>Compute per-vertex surface areas</i>
--------------	---

---

**Description**

Each triangle distributes one-third of its area to each of its three vertices. Triangle area is computed via the cross-product formula.

**Usage**

```
vertex_areas(vertices, faces)
```

**Arguments**

vertices	Numeric matrix (n x 3) of vertex coordinates.
faces	Integer matrix (m x 3) of face indices (1-indexed).

**Value**

Numeric vector of length `nrow(vertices)`.

**Examples**

```
vertices <- matrix(c(0, 1, 0, 0, 0, 1, 0, 0, 0), nrow = 3, byrow = TRUE)
faces <- matrix(c(1L, 2L, 3L), nrow = 1)
vertex_areas(vertices, faces)
```

---

vertices_to_parcel	<i>Aggregate vertex data into parcels</i>
--------------------	---

---

**Description**

Summarises vertex-level data by parcellation labels.

**Usage**

```
vertices_to_parcel(data, labels, summary_func = mean)
```

**Arguments**

data	Numeric vector of vertex-level values.
labels	Integer vector of parcel labels. 0 and NA are treated as medial wall and excluded.
summary_func	Function to summarise each parcel (default: <a href="#">mean</a> ).

**Value**

Named numeric vector of parcel-level values.

**References**

Markello RD et al. (2022) Nature Methods 19:1472-1480. doi:10.1038/s41592-022-01625-w

**Examples**

```
data <- c(1.0, 2.0, 3.0, 4.0)
labels <- c(1L, 1L, 2L, 2L)
vertices_to_parcel(data, labels)
```

# Index

annot\_to\_gifti, [2](#)  
as.matrix.null\_distribution  
    (null\_distribution), [17](#)

check\_wb\_command, [3](#)  
clear\_neuromaps\_cache, [4](#)  
compare\_maps, [4](#)  
compare\_maps(), [23](#)

fetch\_neuromaps\_annotation, [5](#)  
fetch\_neuromaps\_annotation(), [4](#), [11](#)  
fsmorph\_to\_gifti, [6](#)

generate\_nulls, [7](#)  
generate\_nulls(), [5](#), [22](#)  
get\_parcel\_centroids, [8](#)  
get\_surface\_distance, [9](#)  
grepl(), [11](#)

make\_surf\_graph, [10](#)  
mean, [20](#), [27](#)

neuromaps\_available, [11](#)  
neuromaps\_available(), [4](#)  
new\_null\_distribution  
    (null\_distribution), [17](#)  
null\_alexander\_bloch, [12](#)  
null\_baum, [13](#)  
null\_burt2018, [14](#)  
null\_burt2020, [14](#)  
null\_cornblath, [16](#)  
null\_distribution, [5](#), [8](#), [12–16](#), [17](#), [18](#), [19](#)  
null\_moran, [18](#)  
null\_spin\_hungarian (null\_spin\_vasa), [19](#)  
null\_spin\_vasa, [19](#)

parcellate, [20](#)  
parcellate(), [26](#)  
parcels\_to\_vertices, [21](#)  
permtest\_metric, [21](#)

plot.null\_distribution  
    (null\_distribution), [17](#)  
print.neuromaps\_enhanced\_comparison  
    (compare\_maps), [4](#)  
print.null\_distribution  
    (null\_distribution), [17](#)

read\_brain\_map\_values, [23](#)  
resample\_images, [23](#)

summary.null\_distribution  
    (null\_distribution), [17](#)

transform\_to\_space, [25](#)  
transform\_to\_space(), [24](#)

unparcellate, [26](#)

vertex\_areas, [27](#)  
vertices\_to\_parcels, [27](#)