

Package ‘VertexWiseR’

June 13, 2024

Title Simplified Vertex-Wise Analyses of Whole-Brain and Hippocampal Surface

Version 1.0.0

Date 2024-06-13

Maintainer Charly Billaud <charly.billaud@ntu.edu.sg>

Description Provides functions to run statistical analyses on surface-based neuroimaging data, computing measures including cortical thickness and surface area of the whole-brain and of the hippocampi. It can make use of 'FreeSurfer' preprocessed datasets and 'HippUnfold' hippocampal segmentation outputs for a given sample by restructuring the data values into a single file. The single file can then be used by the package for analyses independently from its base dataset and without need for its access.

License GPL-3

URL <https://cogbrainhealthlab.github.io/VertexWiseR/>

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

SystemRequirements Miniconda3 (>= 23.10.0); BrainStat (reticulate installation, >=0.4.2)

Depends R (>= 4.0.0)

Imports doParallel, doSNOW, foreach, freesurferformats, fs, gifti, grDevices, igraph, methods, reticulate, stats, utils

NeedsCompilation no

Author Junhong Yu [aut] (<<https://orcid.org/0000-0002-2563-9658>>),
Charly Billaud [aut, cre] (<<https://orcid.org/0009-0001-3466-9963>>)

Repository CRAN

Date/Publication 2024-06-13 18:00:10 UTC

Contents

atlas_to_surf	2
decode_surf_data	3
edgelistfs5	4
edgelistfs6	4
edgelistHIP	5
fs5_to_fs6	5
fs6_to_fs5	6
fs6_to_fs5_map	6
HIPvextract	7
hip_points_cells	8
MNImap_fs5	8
MNImap_fs6	9
MNImap_hip	9
plot_surf	10
ROImap_fs5	11
ROImap_fs6	12
ROImap_HIP	12
smooth_surf	13
SURFvextract	13
surf_to_atlas	15
surf_to_vol	16
TFCE.threshold	16
TFCE.vertex_analysis	17
TFCE.vertex_analysis.mixed	19
vertex_analysis	20
VWRfirstrun	22

Index	23
--------------	-----------

atlas_to_surf	<i>Atlas to surface</i>
---------------	-------------------------

Description

Maps average parcellation surface values (e.g. produced with the surf_to_atlas() function) to the fsaverage5 or fsaverage6 space

Usage

```
atlas_to_surf(parcel_data, template)
```

Arguments

parcel_data	A matrix or vector object containing average surface measures for each region of interest, see the surf_to_atlas() output format.
template	A string object stating the surface space on which to map the data ('fsaverage5' or 'fsaverage6').

Details

The function currently works with the Desikan-Killiany-70, Schaefer-100, Schaefer-200, Schaefer-400, Glasser-360, or Destrieux-148 atlases. ROI to vertex mapping data for 1 to 4 were obtained from the '[ENIGMA toolbox](#)'; and data for 5 from '[Nilearn](#)''s `nilearn.datasets.fetch_atlas_surf_destrieux`. `atlas_to_surf()` will automatically detect the atlas based on the number of columns.

Value

A matrix or vector object containing vertex-wise surface data mapped in fsaverage5 or fsaverage6 space

See Also

[surf_to_atlas](#)

Examples

```
parcel_data = t(runif(100,min=0, max=100));
surf_data = atlas_to_surf(parcel_data, template='fsaverage5');
```

decode_surf_data

Decode surface data

Description

Correlates the significant clusters of an earlier vertex-wise analysis with a database of task-based fMRI and voxel-based morphometric statistical maps and associate them with relevant key words

Usage

```
decode_surf_data(surf_data, contrast = "positive", VWR_check = TRUE)
```

Arguments

surf_data	a numeric vector with a length of 20484
contrast	A string object indicating whether to decode the positive or negative mask ('positive' or 'negative')
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

The '[NiMARE](#)' python module is used for the imaging decoding and is imported via the `reticulate` package. The function also downloads the '[Neurosynth](#)' database in the package's `inst/extdata` directory (~8 Mb) for the analysis.

Value

A data.frame object listing the keywords and their Pearson's R values

Examples

```
CTv = rbinom(20484, 1, 0.001)
decoding = decode_surf_data(CTv, 'positive', VWR_check=FALSE);
head(decoding)
```

edgelistfs5	<i>List of edges for the fsaverage5 template</i>
-------------	--

Description

A Nx2 matrix object listing each vertex of the fsaverage5 template and the vertices adjacent to it (making an edge together).

Usage

```
edgelistfs5
```

Format

edgelistfs5:

Nx2 matrix object Matrix with two columns and N rows corresponding to the unique edges in the fsaverage5 surface

edgelistfs6	<i>List of edges for the fsaverage6 template</i>
-------------	--

Description

A Nx2 matrix object listing each vertex of the fsaverage5 template and the vertices adjacent to it (making an edge together).

Usage

```
edgelistfs6
```

Format

edgelistfs6:

Nx2 matrix object Matrix with two columns and N rows corresponding to the unique edges in the fsaverage6 surface

edgelistHIP	<i>List of edges for the hippocampal template</i>
-------------	---

Description

A Nx2 matrix object listing each vertex of the hippocampal template and the vertices adjacent to it (making an edge together).

Usage

```
edgelistHIP
```

Format

edgelistHIP:

Nx2 matrix object Matrix with two columns and N rows corresponding to the unique edges in the fsaverage5 surface

fs5_to_fs6	<i>fsaverage5 to fsaverage6</i>
------------	---------------------------------

Description

Remaps vertex-wise surface data in fsaverage5 space to fsaverage6 space using the nearest neighbor approach

Usage

```
fs5_to_fs6(surf_data)
```

Arguments

surf_data A numeric vector or matrix object containing the surface data, see SURFvextract() output format.

Value

A matrix object containing vertex-wise surface data mapped in fsaverage6 space

See Also

[fs6_to_fs5](#)

Examples

```
CTv = runif(20484,min=0, max=100);
CTv_fs6 = fs5_to_fs6(CTv);
```

fs6_to_fs5	<i>fsaverage6 to fsaverage5</i>
------------	---------------------------------

Description

Remaps vertex-wise surface data in fsaverage6 space to fsaverage5 space using the nearest neighbor approach

Usage

```
fs6_to_fs5(surf_data)
```

Arguments

surf_data A numeric vector or matrix object containing the surface data, see SURFvextract() output format.

Value

A matrix object containing vertex-wise surface data mapped in fsaverage5 space

See Also

[fs5_to_fs6](#)

Examples

```
surf_data = runif(81924,min=0, max=100);
fs5_data=fs6_to_fs5(surf_data)
```

fs6_to_fs5_map	<i>fsaverage6 template object for nearest neighbor conversion in fs6_to_fs5()</i>
----------------	---

Description

fsaverage6 template object for nearest neighbor conversion in fs6_to_fs5()

Usage

```
fs6_to_fs5_map
```

Format

fs6_to_fs5_map:

An array of 81924 integers ()

vertices 81924 integers corresponding to each fsaverage6 vertex

`HIPvextract`*HIPvextract*

Description

Extracts hippocampal vertex-wise surface-based measures for each subject in the 'HippUnfold' subjects directory, and stores it as a single .RDS file.

Usage

```
HIPvextract(sdirpath = "./", filename, measure = "thickness", subj_ID = TRUE)
```

Arguments

<code>sdirpath</code>	A string object containing the path to the 'HippUnfold' subjects directory. Default is the current working directory (" <code>./</code> ").
<code>filename</code>	A string object containing the desired name of the output RDS file. Default is 'hip_measure.rds' in the R temporary directory (<code>tempdir()</code>).
<code>measure</code>	A string object containing the name of the measure of interest. Options are 'thickness', 'curvature', 'gyrification' and 'surfarea' (For more information see the 'HippUnfold' documentation). Default is thickness.
<code>subj_ID</code>	A logical object stating whether to return a list object containing both subject ID and data matrix.

Details

The function searches for the hippocampal surface data by listing out files with certain suffixes, extract the data from these files, and organize the left and right hippocampal vertex data for each subject as rows in a $N \times 14524$ data matrix within a .rds object.

Value

A .RDSfile with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or a data matrix object. The matrix can be readily used by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the same order as 1) and contains the left to right hemispheres' vertex-wise values.

Examples

```
HIPvextract(sdirpath = "./", filename = paste0(tempdir(), "/hip_data.RDS"), measure = "thickness")
```

hip_points_cells	<i>points and cells data required to build the hippocampus surface template</i>
------------------	---

Description

points and cells data required to build the hippocampus surface template

Usage

```
hip_points_cells
```

Format

hip_points_cells:

A list object with two data frame objects: ()

vertices data frame with 7262 rows (vertices), 3 columns (MNI coordinates X, y, Z)

vertices data frame with 14266 rows (vertices), 3 columns (vertices of all unique triangles)

vertices data frame with 7262 rows (vertices), 3 columns (MNI coordinates X, y, Z for unfolded hippocampal surface)

MNImap_fs5	<i>fsaverage5 surface in MNI space</i>
------------	--

Description

A matrix with 20484 columns corresponding to the fsaverage5 vertices and 3 rows corresponding to each vertex's X,Y,Z coordinates in MNI space

Usage

```
MNImap_fs5
```

Format

MNImap_fs5:

A 3x20494 matrix object

coordinates 20484 rows (vertices), 3 columns (X,Y,Z coordinates)

MNImap_fs6	<i>fsaverage6 surface in MNI space</i>
------------	--

Description

A matrix with 81924 columns corresponding to the fsaverage6 vertices and 3 rows corresponding to each vertex's X,Y,Z coordinates in MNI space

Usage

MNImap_fs6

Format

MNImap_fs6:

A 3x81924 matrix object

coordinates 81924 rows (vertices), 3 columns (X,Y,Z coordinates)

MNImap_hip	<i>Hippocampal surface in MNI space</i>
------------	---

Description

A matrix with 14524 columns corresponding to the hippocampal template vertices and 3 rows corresponding to each vertex's X,Y,Z coordinates in MNI space

Usage

MNImap_hip

Format

MNImap_hip:

A 3x14524 matrix object

coordinates 14524 rows (vertices), 3 columns (X,Y,Z coordinates)

plot_surf	<i>Surface plotter</i>
-----------	------------------------

Description

Plots surface data in a grid with one or multiple rows in a .png file

Usage

```
plot_surf(
  surf_data,
  filename,
  title = "",
  surface = "inflated",
  cmap,
  limits,
  colorbar = TRUE,
  size,
  zoom,
  VWR_check = TRUE
)
```

Arguments

surf_data	A numeric vector (length of V) or a matrix (N rows x V columns), where N is the number of subplots, and V is the number of vertices. It can be the output from SURFvextract() as well as masks or vertex-wise results outputted by analyses functions.
filename	A string object containing the desired name of the output .png. Default is 'plot.png' in the R temporary directory (tempdir()).
title	A string object for setting the title in the plot. Default is none. For titles that too long to be fully displayed within the plot, we recommend splitting them into multiple lines by inserting "\n".
surface	A string object containing the name of the type of cortical surface background rendered. Possible options include "white", "smoothwm", "pial" and "inflated" (default). The surface parameter is ignored for hippocampal surface data.
cmap	A string object specifying the name of an existing colormap or a vector of hexadecimal color codes to be used as a custom colormap. The names of existing colormaps are listed in the 'Matplotlib' plotting library . Default cmap is set to "Reds" for positive values, "Blues_r" for negative values and "RdBu" when both positive and negative values exist.
limits	A combined pair of numeric vector composed of the lower and upper color scale limits of the plot. If the limits are specified, the same limits will be applied to all subplots. When left unspecified, the same symmetrical limits $c(-\max(\text{abs}(\text{surf_dat}), \max(\text{abs}(\text{surf_dat})))$ will be used for all subplots. If set to

	NULL, each subplot will have its own limits corresponding to their min and max values
colorbar	A logical object stating whether to include a color bar in the plot or not (default is TRUE).
size	A combined pair of numeric vector indicating the image dimensions (width and height in pixels). Default is c(1920,400) for whole-brain surface and c(400,200) for hippocampal surface.
zoom	A numeric value for adjusting the level of zoom on the figures. Default is 1.25 for whole-brain surface and 1.20 for hippocampal surface.
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

Outputs the plot as a .png image

Examples

```
results = runif(20484,min=0, max=1);
plot_surf(surf_data = results, filename=paste0(tempdir(),"/output.png"),title =
'Cortical thickness', surface = 'inflated', cmap = 'Blues',
VWR_check=FALSE)
```

ROImap_fs5

Atlas parcellations of fsaverage5

Description

A list containing two data frames, 1) listing vertex coordinates for each atlas label in fsaverage5 template space, and 2) listing each available atlas and their corresponding labels (1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400).

Usage

```
ROImap_fs5
```

Format

ROImap_fs5:

A list object with two data frame objects: ()

vertices data frame with 20484 rows (vertices), 6 columns (atlases)

atlases data frame with 400 rows (labels, not all are filled depending on atlas), 6 columns (atlases)

ROImap_fs6

*Atlas parcellations of fsaverage6***Description**

A list containing two data frames, 1) listing vertex coordinates for each atlas label in fsaverage6 template space, and 2) listing each available atlas and their corresponding labels (1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400).

Usage

ROImap_fs6

Format

ROImap_fs6:

A list object with two data frame objects: ()

vertices data frame with 81924 rows (vertices), 6 columns (atlases)**atlases** data frame with 400 rows (labels, not all are filled depending on atlas), 6 columns (atlases)

ROImap_HIP

*Atlas parcellations of the hippocampus***Description**

A list containing 1) a matrix listing vertex coordinates for each template hippocampal surface, and 2) a data frame listing 10 bilateral hippocampal subfields and corresponding labels.

Usage

ROImap_HIP

Format

ROImap_HIP:

A list object with two data frame objects: ()

vertices array of 14524 numeric vectors (vertices)**atlases** data frame with 10 rows listing names of left and right hippocampal subfields

smooth_surf	<i>Smooth surface</i>
-------------	-----------------------

Description

Smooths surface data at defined full width at half maximum (FWHM) as per the corresponding template of surface data

Usage

```
smooth_surf(surf_data, FWHM, VWR_check = TRUE)
```

Arguments

surf_data	A matrix object containing the surface data, see SURFvextract() or HIPvextract() output format
FWHM	A numeric vector object containing the desired smoothing width in mm
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

A matrix object with smoothed vertex-wise values

Examples

```
surf_data = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"FINK_Tv_ses13.rds?raw=TRUE")))[1:3,]
surf_data_smoothed=smooth_surf(surf_data, 10, VWR_check=FALSE);
```

SURFvextract	<i>SURFvextract</i>
--------------	---------------------

Description

Extracts whole-brain vertex-wise surface-based measures for each subject in a 'FreeSurfer' output subjects directory, resamples the data to a common surface template, and stores it as a .rds file. This function requires the 'FreeSurfer' environment to be preset.

Usage

```

SURFvextract(
  sdirpath = "./",
  filename,
  template = "fsaverage5",
  measure = "thickness",
  subj_ID = TRUE
)

```

Arguments

<code>sdirpath</code>	A string object containing the path to the 'FreeSurfer' subjects directory. Default is the current working directory ("./").
<code>filename</code>	A string object containing the desired name of the output RDS file. Default is 'brain_measure.rds' in the R temporary directory (<code>tempdir()</code>).
<code>template</code>	A string object containing the name of surface template (available: 'fsaverage5', 'fsaverage6'). Default is fsaverage5.
<code>measure</code>	A string object containing the name of the measure of interest. Options are thickness, curv, sulc, area, and volume (for freesurfer 7.4.1 or later). Default is thickness.
<code>subj_ID</code>	A logical object stating whether to include subject IDs (folder names in the subjects directory) as a first column to the output matrix. Default is TRUE.

Details

The function runs system shell commands that will produce in the set subjects directory: 1) a sorted list of subjects "sublist.txt"; 2) a link file to the selected surface fsaverage template. 3) left and right hemisphere .mgf maps outputted by 'FreeSurfer' 's mris_preproc. This function was currently not tested on a MacOS system.

Value

A .RDSfile with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or a data matrix object. The matrix can be used readily by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the same order as 1) and contains the left to right hemispheres' vertex-wise values.

Examples

```

SURFvextract(sdirpath = "freesurfer_subjdir",
  filename=paste0(tempdir(), "/CTv.rds"), template="fsaverage5",
  measure="curv")

```

surf_to_atlas	<i>Surface to atlas</i>
---------------	-------------------------

Description

Returns the mean or sum of vertex-wise surface data for each ROI of a selected atlas

Usage

```
surf_to_atlas(surf_data, atlas, mode = "mean")
```

Arguments

surf_data	A matrix object containing the surface data in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices) or hippocampal (14524 vertices) space. See also <code>Hipvextract()</code> or <code>SURFvextract()</code> output format.
atlas	A numeric integer object corresponding to the atlas of interest. 1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. For hippocampal surface, the 'bigbrain' hippocampal atlas is used by default and ignores the option.
mode	A string indicating whether to extract the sum ('sum') or the average ('mean') of the ROI vertices values. Default is 'mean'.

Details

The function currently works with the aparc/Desikan-Killiany-70, Destrieux-148, Glasser-360, Schaefer-100, Schaefer-200, Schaefer-400 atlases. ROI to vertex mapping data were obtained from the 'ENIGMA toolbox'; data for Destrieux came from 'Nilearn' `nilearn.datasets.fetch_atlas_surf_destrieux`

For hippocampal data, the function currently works with the "bigbrain" atlas integrated in 'HippUnfold.' See also [doi:10.1016/j.neuroimage.2019.116328](https://doi.org/10.1016/j.neuroimage.2019.116328).

Value

A matrix object with ROI as column and corresponding average vertex-wise values as row

See Also

[atlas_to_surf](#)

Examples

```
CTv = runif(20484,min=0, max=100)
surf_to_atlas(CTv, 1)
```

surf_to_vol	<i>Surface to volume</i>
-------------	--------------------------

Description

Converts surface data to volumetric data (.nii file)

Usage

```
surf_to_vol(surf_data, filename, VWR_check = TRUE)
```

Arguments

surf_data	A vector object containing the surface data, either in fsaverage5 or fsaverage6 space. It can only be one row of vertices (no cohort surface data matrix).
filename	A string object containing the desired name of the output .nii file (default is 'output.nii' in the R temporary directory (tempdir())).
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Value

A .nii volume file

Examples

```
CTv = runif(20484,min=0, max=100);
surf_to_vol(CTv, filename = paste0(tempdir(),'/volume.nii'), VWR_check=FALSE)
```

TFCE.threshold	<i>Thresholding TFCE output</i>
----------------	---------------------------------

Description

Threshold TFCE maps from the TFCE.vertex_analysis() output and identifies significant clusters at the desired threshold.

Usage

```
TFCE.threshold(TFCE.output, p = 0.05, atlas = 1, k = 20)
```


Arguments

TFCE.output	An object containing the output from TFCE.vertex_analysis()
p	A numeric object specifying the p-value to threshold the results (Default is 0.05)
atlas	A numeric integer object corresponding to the atlas of interest. 1=Desikan, 2=Schaefer-100, 3=Schaefer-200, 4=Glasser-360, 5=Destrieux-148 (Default is 1)
k	Cluster-forming threshold (Default is 20)

Value

A list object containing the cluster level results, thresholded t-stat map, and positive, negative and bidirectional cluster maps.

Examples

```
model1_TFCE=readRDS(system.file('demo_data/model1_TFCE.rds',
package = 'VertexWiseR'))

TFCEanalysis_output=TFCE.threshold(model1_TFCE, p=0.05, atlas=1)
TFCEanalysis_output$cluster_level_results
```

TFCE.vertex_analysis *Vertex-wise analysis with TFCE (fixed effect)*

Description

Fits a linear model with the cortical or hippocampal surface data as the predicted outcome, and returns t-stat and TFCE statistical maps for the selected contrast.

Usage

```
TFCE.vertex_analysis(
  model,
  contrast,
  surf_data,
  nperm = 100,
  tail = 2,
  nthread = 10,
  smooth_FWHM,
  VWR_check = TRUE
)
```

Arguments

model	An N X V data.frame object containing N rows for each subject and V columns for each predictor included in the model
contrast	A numeric vector or object containing the values of the predictor of interest. The t-stat and TFCE maps will be estimated only for this predictor
surf_data	A matrix object containing the surface data, see SURFvextract() or HIPvextract() output format.
nperm	A numeric integer object specifying the number of permutations generated for the subsequent thresholding procedures (default = 100)
tail	A numeric integer object specifying whether to test a one-sided positive (1), one-sided negative (-1) or two-sided (2) hypothesis
nthread	A numeric integer object specifying the number of CPU threads to allocate
smooth_FWHM	A numeric vector object specifying the desired smoothing width in mm
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

This TFCE method is adapted from the ['Nilearn' Python library](#).

Value

A list object containing the t-stat and the TFCE statistical maps which can then be subsequently thresholded using TFCE.threshold()

See Also

[TFCE.threshold](#)

Examples

```
demodata = readRDS(system.file('demo_data/SPRENG_behdata_site1.rds',
package = 'VertexWiseR'))[1:5,]
surf_data = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"SPRENG_CTv_site1.rds?raw=TRUE")))[1:5,]
model=demodata[,c(2,7)]
contrast=demodata[,7]

TFCE.pos=TFCE.vertex_analysis(model, contrast, surf_data, tail=1,
nperm=5, nthread = 2, VWR_check=FALSE)

#To threshold the results, you may then run:
#results=TFCE.threshold(TFCE.pos, p=0.05, atlas=1)
#results$cluster_level_results
```

 TFCE.vertex_analysis.mixed

Vertex-wise analysis with TFCE (mixed effect)

Description

Fits a linear mixed effects model with the cortical or hippocampal surface data as the predicted outcome, and returns t-stat and TFCE statistical maps for the selected contrast.

Usage

```
TFCE.vertex_analysis.mixed(
  model,
  contrast,
  surf_data,
  random,
  nperm = 100,
  tail = 2,
  nthread = 10,
  smooth_FWHM,
  perm_type = "row",
  VWR_check = TRUE
)
```

Arguments

model	An N X V data.frame object containing N rows for each subject and V columns for each predictor included in the model. This data.frame should not include the random effects variable.
contrast	A numeric vector or object containing the values of the predictor of interest. The t-stat and TFCE maps will be estimated only for this predictor
surf_data	A matrix object containing the surface data, see SURFvextract() or HIPvextract() output format.
random	An object or vector containing the values of the random variable
nperm	A numeric integer object specifying the number of permutations generated for the subsequent thresholding procedures (default = 100)
tail	A numeric integer object specifying whether to test a one-sided positive (1), one-sided negative (-1) or two-sided (2) hypothesis
nthread	A numeric integer object specifying the number of CPU threads to allocate
smooth_FWHM	A numeric vector object specifying the desired smoothing width in mm
perm_type	A string object specifying whether to permute the rows ("row"), between subjects ("between"), within subjects ("within") or between and within subjects ("within_between") for random subject effects. Default is "row".
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

This TFCE method is adapted from the ['Nilearn' Python library](#).

Value

A list object containing the t-stat and the TFCE statistical maps which can then be subsequently thresholded using `TFCE.threshold()`

Examples

```
demodata = readRDS(system.file('demo_data/SPRENG_behdata_site1.rds', package = 'VertexWiseR'))[1:5,]
surf_data = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"SPRENG_CTv_site1.rds?raw=TRUE")))[1:5,]

TFCE.pos=TFCE.vertex_analysis.mixed(model=demodata[,c(2,7)],
contrast=demodata[,7], surf_data=surf_data,random=demodata[,1],
nperm =5,tail = 1, nthread = 2, VWR_check=FALSE)

#To get significant clusters, you may then run:
#results=TFCE.threshold(TFCE.pos, p=0.05, atlas=1)
#results$cluster_level_results
```

vertex_analysis

Vertex-wise analysis

Description

Fits a linear or linear mixed model with the cortical or hippocampal surface data as the predicted outcome, and returns cluster-thresholded (Random field theory) t-stat map selected contrast.

Usage

```
vertex_analysis(  
  model,  
  contrast,  
  random,  
  surf_data,  
  p = 0.05,  
  atlas = 1,  
  smooth_FWHM,  
  VWR_check = TRUE  
)
```

Arguments

model	An N X V data.frame object containing N rows for each subject and V columns for each predictor included in the model. This data.frame should not include the random effects variable.
contrast	A numeric vector or object containing the values of the predictor of interest. The cluster-thresholded t-stat maps will be estimated only for this predictor
random	An object containing the values of the random variable (optional)
surf_data	A matrix object containing the surface data, see SURFvextract() or HIPvextract() output format.
p	A numeric object specifying the p-value to threshold the results (Default is 0.05)
atlas	A numeric integer object corresponding to the atlas of interest. 1=Desikan, 2=Schaefer-100, 3=Schaefer-200, 4=Glasser-360, 5=Destrieux-148.
smooth_FWHM	A numeric vector object specifying the desired smoothing width in mm
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

Details

The function imports and adapts the **'BraiStat' Python library**.

Output definitions:

- nverts: number of vertices in the cluster
- P: p-value of the cluster
- X, Y and Z: MNI coordinates of the vertex with the highest t-statistic in the cluster.
- tstat: t statistic of the vertex with the highest t-statistic in the cluster
- region: the region this highest t-statistic vertex is located in, as determined/labelled by the selected atlas

Value

A list object containing the cluster level results, thresholded t-stat map, and positive, negative and bidirectional cluster maps.

Examples

```
demodata = readRDS(system.file('demo_data/SPRENG_behdata_site1.rds',
package = 'VertexWiseR'))[1:100,]
CTv = readRDS(file = url(paste0("https://github.com",
"/CogBrainHealthLab/VertexWiseR/blob/main/inst/demo_data/",
"SPRENG_CTv_site1.rds?raw=TRUE")))[1:100,]

vertexwise_model=vertex_analysis(model=demodata[,c(2,7)],
contrast=demodata[,7], surf_data = CTv, atlas=1,p = 0.05,
VWR_check=FALSE)

#Description of the output:
#vertexwise_model$cluster_level_results
```

VWRfirstrun

VertexWiseR system requirements installation

Description

Helps the user verify if VertexWiseR's system requirements are present and install them ('Miniconda', 'BrainStat' toolbox and libraries). If they are installed already, nothing will be overwritten.

Usage

```
VWRfirstrun(requirement = "any", n_vert = 0)
```

Arguments

requirement	String that specifies a requirement to enquire about (for specific 'BrainStat' libraries: 'fsaverage5', 'fsaverage6', 'yeo_parcel'; for neurosynth database: "neurosynth"). Default is 'any' requirement and checks everything.
n_vert	Numeric vector indicating the number of vertices of a given surface data so that only the required templates are asked for

Details

VertexWiseR imports and makes use of the R package 'reticulate.' 'reticulate' is a package that allows R to borrow or translate Python functions into R. Using 'reticulate', the package calls functions from the 'BrainStat' Python module. For 'reticulate' to work properly with VertexWiseR, the latest version of 'Miniconda' needs to be installed with it — 'Miniconda' is a lightweight version of Python, specifically for use within 'RStudio'. Likewise, analyses of cortical surface require fsaverage templates as imported by 'BrainStat' The `decode_surf_data()` function also requires the 'Neurosynth' database to be downloaded.

Value

No returned value in interactive session. In non-interactive sessions, a string object informing that system requirements are missing.

Examples

```
VWRfirstrun()
```

Index

* datasets

- edgelistfs5, 4
- edgelistfs6, 4
- edgelistHIP, 5
- fs6_to_fs5_map, 6
- hip_points_cells, 8
- MNImap_fs5, 8
- MNImap_fs6, 9
- MNImap_hip, 9
- ROImap_fs5, 11
- ROImap_fs6, 12
- ROImap_HIP, 12

atlas_to_surf, 2, 15

decode_surf_data, 3

- edgelistfs5, 4
- edgelistfs6, 4
- edgelistHIP, 5

- fs5_to_fs6, 5, 6
- fs6_to_fs5, 5, 6
- fs6_to_fs5_map, 6

- hip_points_cells, 8
- HIPvextract, 7

- MNImap_fs5, 8
- MNImap_fs6, 9
- MNImap_hip, 9

plot_surf, 10

- ROImap_fs5, 11
- ROImap_fs6, 12
- ROImap_HIP, 12

- smooth_surf, 13
- surf_to_atlas, 3, 15
- surf_to_vol, 16

SURFvextract, 13

- TFCE.threshold, 16, 18
- TFCE.vertex_analysis, 17
- TFCE.vertex_analysis.mixed, 19

- vertex_analysis, 20
- VWRfirstrun, 22