

# Package ‘VertexWiseR’

January 20, 2025

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

**Version** 1.2.0

**Date** 2024-12-09

**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', 'fMRIPrep' and 'HCP' 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/>

**Suggests** R.rsp

**VignetteBuilder** R.rsp

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**SystemRequirements** Miniconda3 (v24.9.2), Python (<= v3.12), BrainStat (v0.4.2), vtk (v9.3.1)

**Depends** R (>= 4.0.0)

**Imports** ciftiTools, doParallel, doSNOW, foreach, freesurferformats, fs, gifti, grDevices, igraph, methods, plotly, png, rappdirs, reticulate, stats, stringr, 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** 2024-12-12 07:30:02 UTC

## Contents

atlas_to_surf . . . . .	2
CAT12vextract . . . . .	3
decode_surf_data . . . . .	5
edgelist-class . . . . .	6
edgelist_hip . . . . .	6
fs5_to_fs6 . . . . .	6
fs6_to_fs5 . . . . .	7
fs6_to_fs5_map . . . . .	8
FSLRvextract . . . . .	8
fs_stats . . . . .	10
HIPvextract . . . . .	10
hip_points_cells . . . . .	11
MNImap_hip . . . . .	12
MNIsurface-class . . . . .	12
plot_surf . . . . .	13
plot_surf3d . . . . .	14
RFT_vertex_analysis . . . . .	16
ROImap-class . . . . .	18
ROImap_fs5 . . . . .	18
ROImap_fs6 . . . . .	19
ROImap_fslr32k . . . . .	19
ROImap_hip . . . . .	20
smooth_surf . . . . .	20
SURFvextract . . . . .	21
surf_to_atlas . . . . .	22
surf_to_vol . . . . .	23
TFCE_threshold . . . . .	24
TFCE_vertex_analysis . . . . .	25
TFCE_vertex_analysis_mixed . . . . .	27
VWRfirstrun . . . . .	29
<b>Index</b>	<b>31</b>

---

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, fsaverage6 or fs1r32k 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', 'fsaverage6', 'fsLR32k', 'CIT168' (hippocampal)).

**Details**

The function currently supports the Desikan-Killiany-70, Schaefer-100, Schaefer-200, Schaefer-400, Glasser-360, or Destrieux-148 atlases for cortical surfaces, and the 'bigbrain' 10-parcels atlas for hippocampal surfaces. 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, fsaverage6, fsLR32k, or CIT168 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');
```

---

CAT12vextract

*CAT12vextract*

---

**Description**

Extracts vertex-wise surface-based measures for each subject from a **CAT12** preprocessed directory, resampled to a 32k mesh, and stores them all as a single .RDS file.

**Usage**

```
CAT12vextract(
  sdirpath = "./",
  filename,
  measure = "thickness",
  subj_ID = TRUE,
  silent = FALSE,
  VWR_check = TRUE
)
```

### Arguments

<code>sdirpath</code>	A string object containing the path to the CAT12 subjects preprocessed 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 <code>'CAT12_measure.rds'</code> in the R temporary directory ( <code>tempdir()</code> ).
<code>measure</code>	A string object containing the name of the measure of interest. Options are <code>'thickness'</code> , <code>'depth'</code> , <code>'fractaldimension'</code> , <code>'gyrification'</code> , and <code>'toroGI20mm'</code> . Default is <code>'thickness.'</code>
<code>subj_ID</code>	A logical object to determine whether to return a list object containing both subject ID and data matrix. Subject IDs are assumed to be the top directory names in the <code>sdirpath</code> .
<code>silent</code>	A logical object to determine whether messages will be silenced. Set to <code>'FALSE'</code> by default
<code>VWR_check</code>	A boolean object specifying whether to check and validate system requirements. Default is <code>TRUE</code> .

### Details

The function searches inside the CAT12 preprocessed for 32k meshes (.gii) with the user-selected measure, extracts the data from these files, and organizes the left and right hemisphere vertex data for each subject as rows in a  $N \times 64984$  data matrix within a .rds object. Python and reticulate are required as the [NiBabel](#) package is used to import .gii files outputted by CAT12.

### 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 only a data matrix object. The matrix has  $N$  subjects  $\times$   $M$  vertices dimensions and can be readily used by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' vertex-wise values.

### Examples

```
CAT12vextract(sdirpath="./",
filename='thickness.rds',
measure='thickness',
subj_ID = TRUE,
VWR_check=FALSE)
```

---

decode_surf_data	<i>Decode surface data</i>
------------------	----------------------------

---

### 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. Decoding currently works with surfaces in fsaverage5 space only."

### Usage

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

### Arguments

surf_data	A numeric vector or object containing the surface data, in fsaverage5 (1 x 20484 vertices). It can only be one row of vertices (not a cohort surface data matrix).
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)
```

---

edgelist-class	<i>Edge list object</i>
----------------	-------------------------

---

### Description

A class for adjacent vertex correspondance

### Slots

matrix A N x 2 matrix object listing each vertex of the template and the vertices adjacent to it (making an edge together).

name The name of the template surface

---

edgelist_hip	<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

```
edgelist_hip
```

### Format

```
edgelist_hip:
```

**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  $N \times V$  matrix object containing the surface data ( $N$  row for each subject,  $V$  for each vertex), in fsaverage5 (20484 vertices) space. See also 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  $N \times V$  matrix object containing the surface data ( $N$  row for each subject,  $V$  for each vertex), in fsaverage6 (81924 vertices) space. See also 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</i> template object for nearest neighbor conversion in <i>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

---

FSLRvextract	<i>FSLRvextract</i>
--------------	---------------------

---

**Description**

Extracts vertex-wise surface-based measures for each subject from HCP and fMRIprep preprocessed directory, and stores it as a single .RDS file.

**Usage**

```
FSLRvextract(
  sdirpath = "./",
  wb_path,
  filename,
  dscaler,
  subj_ID = TRUE,
  silent = FALSE
)
```

**Arguments**

sdirpath	A string object containing the path to the HCP or fMRIprep preprocessed directory. Default is the current working directory ("./").
wb_path	The filepath to the workbench folder that you have previously downloaded and unzipped
filename	A string object containing the desired name of the output RDS file. Default is 'fslr32k.rds' in the R temporary directory (tempdir()).



dscaler	<p>A string object containing the filename suffix of the dscaler file. These dscaler files are named differently depending on the preprocessing pipeline used. Examples of filename suffixes are shown below</p> <ul style="list-style-type: none"> <li>• <code>.thickness_MSMA11.32k_fs_LR.dscalar.nii</code> (HCP MSMA11 pipeline)</li> <li>• <code>.sulc_MSMA11.32k_fs_LR.dscalar.nii</code> (HCP MSMA11 pipeline)</li> <li>• <code>.thickness.32k_fs_LR.dscalar.nii</code> (HCP legacy pipeline)</li> <li>• <code>.sulc.32k_fs_LR.dscalar.nii</code> (HCP legacy pipeline)</li> <li>• <code>_space-fsLR_den-91k_thickness.dscalar.nii</code> (fMRIPrep; using the <code>--cifti-output 91k</code> flag)</li> <li>• <code>_space-fsLR_den-91k_curv.dscalar.nii</code> (fMRIPrep; using the <code>--cifti-output 91k</code> flag)</li> <li>• <code>_space-fsLR_den-91k_sulc.dscalar.nii</code> (fMRIPrep; using the <code>--cifti-output 91k</code> flag)</li> </ul>
subj_ID	A logical object to determine whether to return a list object containing both subject ID and data matrix.
silent	A logical object to determine whether messages will be silenced. Set to 'FALSE' by default

## Details

The function searches for the HCP preprocessed directory by listing out files with certain suffixes, extract the data from these files, and organize the left and right hemisphere vertex data for each subject as rows in a  $N \times 64984$  data matrix within a `.rds` object.

## Value

A `.RDS` file with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or only a data matrix object. The matrix has  $N$  subjects  $\times$   $M$  vertices dimensions and can be readily used by `VertexWiseR` statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' vertex-wise values.

## Examples

```
dat_fslr32k=FSLRvextract(sdirpath=".",
wb_path="/path/to/workbench",
filename="dat_fslr32k.rds",
dscaler=".thickness_MSMA11.32k_fs_LR.dscalar.nii",
subj_ID = TRUE, silent=FALSE)
```

---

fs_stats	<i>fs_stats()</i>
----------	-------------------

---

### Description

Extracts descriptive statistics, for the whole-brain and subcortical region-of-interests (ROI), within a FreeSurfer subjects directory. It reads them from the aseg.stats file, as generated by the default FreeSurfer preprocessing pipeline.

### Usage

```
fs_stats(sdirpath = "./", sublist, ROImeasure = "Volume_mm3")
```

### Arguments

sdirpath	A string object indicating the path to the 'FreeSurfer' subjects directory. Default is the current working directory ("./").
sublist	A string object indicating the path to the subject list generated by SURFvextract as 'sublist.txt' (optional). This allows users to retrieve stats only from a selected list of subjects. The subject list is a list with 1 subject ID per line.
ROImeasure	A string object indicating what summary measure to extract for the subcortical ROIs. Choices include: 'NVoxels', 'Volume_mm3', 'StructName', 'norm-Mean', 'normStdDev', 'normMin', 'normMax', and 'normRange'. Default is 'Volume_mm3'.

### Value

A data.frame object with N columns per aseg.stats measures and N row per subjects.

### Examples

```
fs_stats(sdirpath="freesurfer_subjdir")
```

---

HIPvextract	<i>HIPvextract</i>
-------------	--------------------

---

### 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**

sdirpath	A string object containing the path to the 'HippUnfold' subjects directory. Default is the current working directory (".").
filename	A string object containing the desired name of the output RDS file. Default is 'hip_measure.rds' in the R temporary directory (tempdir()).
measure	A string object containing the name of the measure of interest. Options are 'thickness', 'curvature', 'gyrification' and 'surfarea' (For more information see <a href="#">the 'HippUnfold' documentation</a> ). Default is thickness.
subj_ID	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 x 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 only a data matrix object. The matrix has N subjects x M vertices dimensions and can be readily used by VertexWiseR statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' hippocampal 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)

---

MNI`map_hip`

*Hippocampal surface in MNI space*

---

**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**

MNI`map_hip`

**Format**

MNI`map_hip`:

A 3x14524 matrix object

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

---

MNI`surface-class`

*MNI surface map object*

---

**Description**

A class for surface coordinates in MNI space

**Slots**

`matrix` A matrix object with N vertices columns x 3 rows corresponding to each vertex's X,Y,Z coordinates in MNI space.

`name` The name of the template surface

---

plot\_surf

*Surface plotter*


---

### 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,
  show.plot.window = TRUE,
  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(), FSLRvextract(), HIPvextract() as well as masks or vertex-wise results outputted by analyses functions. Alternatively, atlas ROI values as supported by atlas_to_surf() may be given.
filename	A string object containing the desired name of the output .png. Default is 'plot.png' in the R temporary directory (tempdir()). Only filenames with a .png extension are allowed.
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 <a href="#">'Matplotlib' plotting library</a> . 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.
show.plot.window	A logical object to determine if the generated plot is to be shown within RStudio's plot window
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)
```

---

plot\_surf3d

*3D Surface plotter*


---

**Description**

Plots 3D cortical surfaces

**Usage**

```
plot_surf3d(
  surf_data,
  surf_color = "grey",
  cmap,
  limits,
```

```

    atlas = 1,
    hemi = "b",
    medial_gap = 0,
    orientation_labels = TRUE,
    VWR_check = TRUE
)

```

### Arguments

surf_data	A numeric vector (length of V)
surf_color	color of the cortical surface. Set to 'grey' by default
cmap	A string vector containing 2 to 4 color names/codes specifying the colors to be used for the color scale. See <code>RColorBrewer::display.brewer.all()</code> for all possible cmap options. If none are specified, appropriate colors will be automatically selected according to <code>range(surf_data)</code>
limits	A combined pair of numeric vector composed of the lower and upper color scale limits of the plot. When left unspecified, the symmetrical limits <code>c(-max(abs(surf_data)), max(abs(surf_data)))</code> will be used.
atlas	atlas used for identifying region labels. 1=Desikan, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for hippocampal surfaces.
hemi	A string specifying the hemisphere to plot. Possible values are l (left), r (right) or b (both).
medial_gap	A numeric value specifying the amount of gap (in MNI coordinate units) to separate the left and right hemispheres. Set to 0 (no gap between hemispheres) by default. In order to view the medial surfaces clearly, it is recommended that this value is set to 20. This argument is ignored if <code>hemi != 'b'</code>
orientation_labels	A boolean object specifying if orientation labels are to be displayed. Set to TRUE by default
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

### Value

a plot\_ly object

### Examples

```

surf_data = runif(20484);
plot_surf3d(surf_data = surf_data, VWR_check=FALSE)

```

---

RFT\_vertex\_analysis     *Vertex-wise analysis with random field theory cluster correction*

---

### 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

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

### Arguments

model	An N X P data.frame object containing N rows for each subject and P columns for each predictor included in the model. This data.frame should not include the random effects variable.
contrast	A N x 1 numeric vector or object containing the values of the predictor of interest. Its length should equal the number of subjects in model (and can be a single column from model). The cluster-thresholded t-stat maps will be estimated only for this predictor.
random	A N x 1 numeric vector or object containing the values of the random variable (optional). Its length should be equal to the number of subjects in model (it should NOT be inside the model data.frame).
formula	An optional string or formula object describing the predictors to be fitted against the surface data, replacing the model, contrast, or random arguments. If this argument is used, the formula_dataset argument must also be provided. <ul style="list-style-type: none"> <li>• The dependent variable is not needed, as it will always be the surface data values.</li> <li>• The first independent variable in the formula will always be interpreted as the contrast of interest for which to estimate cluster-thresholded t-stat maps.</li> <li>• Only one random regressor can be given and must be indicated as '(1 variable_name)'</li> </ul>



formula_dataset	An optional data.frame object containing the independent variables to be used with the formula (the IV names in the formula must match their column names in the dataset).
surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fsLR32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract() or FSLRvextract output formats. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.
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=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for hippocampal surfaces.
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 ['BrainStat' Python library](#).

By default, false discovery rate correction is used together with the Random field theory (RFT) cluster correction. To look at data without any form of cluster correction, users can simply refer to the outputted 'tstat\_map'.

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, unthresholded t-stat map, thresholded t-stat map, positive, negative and bidirectional cluster maps, and a FDR-corrected p-value map.

## See Also

[TFCE\\_vertex\\_analysis](#), [TFCE\\_vertex\\_analysis\\_mixed](#)

**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=RFT_vertex_analysis(model=demodata[,c("sex","age")],
contrast=demodata[,"age"], surf_data = CTv, atlas=1,p = 0.05,
VWR_check=FALSE)

#Description of the output:
#vertexwise_model$cluster_level_results

#Formula alternative:
#formula= as.formula("~ age + sex")
#vertexwise_model=RFT_vertex_analysis(formula=formula,
#formula_dataset=demodata, surf_data = CTv, atlas=1, p = 0.05,
#VWR_check=FALSE)

```

---

ROImap-class	<i>Region-of-Interest mapping object</i>
--------------	--

---

**Description**

A class for surface vertices mapping on atlas labels

**Slots**

**matrix** A matrix object with N vertices from a template and each parcellation number the vertices correspond in 6 atlases (6 columns).

**atlases** Each available of the 6 available atlases and their corresponding labels (1=aparc, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400).

**name** The name of the template surface

---

ROImap_fs5	<i>Atlas parcellations of fsaverage5</i>
------------	--

---

**Description**

A list containing two data frames, 1) listing fsaverage5 vertices and each parcellation number they correspond to, 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 fsaverage6 vertices and each atlas parcellation number they correspond to, 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\_fs1r32k

*Atlas parcellations of FS\_LR32k***Description**

A list containing two data frames, 1) listing FS\_LR32k vertices and each atlas parcellation number they correspond to, 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\_fs1r32k

**Format**

ROImap\_fs1r32k:

A list object with two data frame objects: ()

**vertices** data frame with 64984 rows (vertices), 6 columns (atlases)**atlases** data frame with 400 rows (labels, not all are filled depending on atlas), 6 columns (atlases)

---

ROImap_hip	<i>Atlas parcellations of the hippocampus (CITI168)</i>
------------	---

---

### Description

A list containing 1) a matrix listing CITI168 vertices and each parcellation number they correspond to, and 2) a data frame listing the hippocampal atlas 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 N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fsLR32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract() or FSLRvextract output formats. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.

**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

*SURFvextract*


---

**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 in the unix environment and a 'FreeSurfer' license key.

**Usage**

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

**Arguments**

sdirpath	A string object containing the path to the 'FreeSurfer' preprocessed subjects directory. This directory must be the output directory from a <b>FreeSurfer pre-processing recon-all pipeline</b> . Default is the current working directory ("./").
filename	A string object containing the desired name of the output RDS file. Default is 'brain_measure.rds' in the R temporary directory (tempdir()).
template	A string object containing the name of surface template (available: 'fsaverage5', 'fsaverage6'). Default is fsaverage5.
measure	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.
subj_ID	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.

`fshomepath` An optional string object containing the path to the FreeSurfer installation directory. This makes sure R accesses FreeSurfer if the system environment variables are not inherited — as would be the case if you are running the function from RStudio.

### Details

Note that RStudio does not inherit the shell environment variables if it is open from a terminal. In that case, the `fshomepath` argument needs to be provided. 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 `.mgh` maps outputted by 'FreeSurfer' 's `mris_preproc`.

### Value

A `.RDS`file with a list containing 1. the list of subject IDs (first element) and 2. a surface data matrix object (second element), or only a data matrix object. The matrix has  $N$  subjects  $\times$   $M$  vertices dimensions and can be used readily by `VertexWiseR` statistical analysis functions. Each row corresponds to a subject (in the order they are listed in the folder) and contains the left to right hemispheres' vertex-wise values.

### Examples

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

---

<code>surf_to_atlas</code>	<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

<code>surf_data</code>	A $N \times V$ matrix object containing the surface data ( $N$ row for each subject, $V$ for each vertex), in <code>fsaverage5</code> (20484 vertices), <code>fsaverage6</code> (81924 vertices), <code>fslr32k</code> (64984 vertices) or <code>hippocampal</code> (14524 vertices) space. See also <code>Hipvextract()</code> , <code>SURFvextract()</code> or <code>FSLRvextract</code> output formats.
<code>atlas</code>	A numeric integer object corresponding to the atlas of interest. 1=Desikan, 2=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for <code>hippocampal</code> surfaces.
<code>mode</code>	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' 's `nilearn.datasets.fetch_atlas_surf_destrieux`

For hippocampal data, the function currently works with the "bigbrain" 10-parcels 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)
parcel_data = 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 numeric vector or object containing the surface data, either in fsaverage5 (1 x 20484 vertices) or fsaverage6 (1 x 81924 vertices) space. It can only be one row of vertices (not a 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(TFCEoutput, p = 0.05, atlas = 1, k = 20, VWR_check = TRUE)
```

### Arguments

TFCEoutput	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=Destrieux-148, 3=Glasser-360, 4=Schaefer-100, 5=Schaefer-200, 6=Schaefer-400. Set to 1 by default. This argument is ignored for hippocampal surfaces.
k	Cluster-forming threshold (Default is 20)
VWR_check	A boolean object specifying whether to check and validate system requirements. Default is TRUE.

### 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,
VWR_check=FALSE)
TFCEanalysis_output$cluster_level_results
```



---

TFCE\_vertex\_analysis *Vertex-wise analysis with threshold-free cluster enhancement (fixed effect)*

---

### Description

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

### Usage

```
TFCE_vertex_analysis(
  model,
  contrast,
  formula,
  formula_dataset,
  surf_data,
  nperm = 100,
  tail = 2,
  nthread = 10,
  smooth_FWHM,
  VWR_check = TRUE
)
```

### Arguments

model	An N X P data.frame object containing N rows for each subject and P columns for each predictor included in the model
contrast	A N x 1 numeric vector or object containing the values of the predictor of interest. Its length should equal the number of subjects in model (and can be a single column from model). The t-stat and TFCE maps will be estimated only for this predictor.
formula	An optional string or formula object describing the predictors to be fitted against the surface data, replacing the model, contrast, or random arguments. If this argument is used, the formula_dataset argument must also be provided. <ul style="list-style-type: none"> <li>• The dependent variable is not needed, as it will always be the surface data values.</li> <li>• The first independent variable in the formula will always be interpreted as the contrast of interest for which to estimate cluster-thresholded t-stat maps.</li> <li>• Only one random regressor can be given and must be indicated as '(1 variable_name)'</li> </ul>
formula_dataset	An optional data.frame object containing the independent variables to be used with the formula (the IV names in the formula must match their column names in the dataset).

surf_data	A $N \times V$ matrix object containing the surface data ( $N$ row for each subject, $V$ for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fsLR32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract() or FSLRvextract output formats. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.
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

[RFT\\_vertex\\_analysis](#), [TFCE\\_vertex\\_analysis\\_mixed](#), [TFCE\\_threshold](#)

### Examples

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

TFCEpos=TFCE_vertex_analysis(model=demodata[,c("sex","age")],
contrast=demodata[,"age"], surf_data=CTv, tail=1,
nperm=5, nthread = 2, VWR_check=FALSE)

#To threshold the results, you may then run:
#results=TFCE_threshold(TFCEpos, p=0.05, atlas=1)
#results$cluster_level_results

#Formula alternative:
#formula= as.formula("~ age + sex")
#TFCEpos=TFCE_vertex_analysis(formula=formula,
#formula_dataset=demodata, surf_data=CTv, tail=1,
#nperm=5, nthread = 2, VWR_check=FALSE)
```

---

 TFCE\_vertex\_analysis\_mixed

*Vertex-wise analysis with threshold-free cluster enhancement (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 threshold-free cluster enhancement (TFCE) statistical maps for the selected contrast.

## Usage

```
TFCE_vertex_analysis_mixed(
  model,
  contrast,
  random,
  formula,
  formula_dataset,
  surf_data,
  nperm = 100,
  tail = 2,
  nthread = 10,
  smooth_FWHM,
  perm_type = "row",
  VWR_check = TRUE
)
```

## Arguments

- |          |   |
|----------|---|
| model    | An N X P data.frame object containing N rows for each subject and P columns for each predictor included in the model. This data.frame should not include the random effects variable.   |
| contrast | A N x 1 numeric vector or object containing the values of the predictor of interest. Its length should equal the number of subjects in model (and can be a single column from model). The t-stat and TFCE maps will be estimated only for this predictor.   |
| random   | A N x 1 numeric vector or object containing the values of the random variable (optional). Its length should be equal to the number of subjects in model (it should NOT be inside the model data.frame).   |
| formula  | An optional string or formula object describing the predictors to be fitted against the surface data, replacing the model, contrast, or random arguments. If this argument is used, the formula_dataset argument must also be provided. <ul style="list-style-type: none"> <li>• The dependent variable is not needed, as it will always be the surface data values.</li> </ul> |

- The first independent variable in the formula will always be interpreted as the contrast of interest for which to estimate cluster-thresholded t-stat maps.
- Only one random regressor can be given and must be indicated as '(1|variable\_name)'

formula_dataset	An optional data.frame object containing the independent variables to be used with the formula (the IV names in the formula must match their column names in the dataset).
surf_data	A N x V matrix object containing the surface data (N row for each subject, V for each vertex), in fsaverage5 (20484 vertices), fsaverage6 (81924 vertices), fsLR32k (64984 vertices) or hippocampal (14524 vertices) space. See also Hipvextract(), SURFvextract() or FSLRvextract output formats. Alternatively, a string object containing the path to the surface object (.rds file) outputted by extraction functions may be given.
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()`

### See Also

[RFT\\_vertex\\_analysis](#), [TFCE\\_vertex\\_analysis](#), [TFCE\\_threshold](#)

### Examples

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

TFCEpos=TFCE_vertex_analysis_mixed(model=demodata[,c("sex",
```

```

"age"]], contrast=demodata[, "age"], random=demodata[, "id"],
surf_data=CTv, nperm =5,tail = 1, nthread = 2, VWR_check=FALSE)

#To get significant clusters, you may then run:
#results=TFCE_threshold(TFCEpos, p=0.05, atlas=1)
#results$cluster_level_results

#Formula alternative:
#formula= as.formula("~ age + sex + (1|id)")
#TFCEpos=TFCE_vertex_analysis_mixed(formula=formula,
#formula_dataset=demodata, surf_data=CTv, tail=1,
#nperm=5, nthread = 2, VWR_check=FALSE)

```

---

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, promptless = FALSE)
```

## Arguments

requirement	String that specifies a requirement to enquire about: <ul style="list-style-type: none"> <li>• For only Python/Conda installation: 'python/conda only'</li> <li>• For Python/Conda and Brainstat installation: 'conda/brainstat'</li> <li>• For specific 'BrainStat' libraries: 'fsaverage5', 'fsaverage6', 'fslr32k', 'yeo_parcel'</li> <li>• For the neurosynth database: 'neurosynth'. Default is 'any' and checks everything.</li> </ul>
n_vert	Numeric vector indicating the number of vertices of a given surface data so that only the required templates are asked for. It will modify the requirement argument accordingly.
promptless	A boolean object specifying whether to prompt the user for action when system requirements are missing. If TRUE, VWRfirstrun() will simply inform of what is missing and will not prompt for action. Default is FALSE.

## 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,

'Miniconda' needs to be installed with it — 'Miniconda' is a lightweight version of Python, specifically for use within 'RStudio'. If for a reason Miniconda cannot be installed, the function gives the choice to install a reticulate-suitable Python environment. Vertex-wise statistical analyses of cortical surface require fsaverage and parcellation templates as imported by default in '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

- edgelist\_hip, 6
  - fs6\_to\_fs5\_map, 8
  - hip\_points\_cells, 11
  - MNImap\_hip, 12
  - ROImap\_fs5, 18
  - ROImap\_fs6, 19
  - ROImap\_fslr32k, 19
  - ROImap\_hip, 20
- atlas\_to\_surf, 2, 23
- CAT12vextract, 3
- decode\_surf\_data, 5
- edgelist-class, 6
- edgelist\_hip, 6
- fs5\_to\_fs6, 6, 7
- fs6\_to\_fs5, 7, 7
- fs6\_to\_fs5\_map, 8
- fs\_stats, 10
- FSLRvextract, 8
- hip\_points\_cells, 11
- HIPvextract, 10
- MNImap\_hip, 12
- MNIsurface-class, 12
- plot\_surf, 13
- plot\_surf3d, 14
- RFT\_vertex\_analysis, 16, 26, 28
- ROImap-class, 18
- ROImap\_fs5, 18
- ROImap\_fs6, 19
- ROImap\_fslr32k, 19
- ROImap\_hip, 20
- smooth\_surf, 20
- surf\_to\_atlas, 3, 22
- surf\_to\_vol, 23
- SURFvextract, 21
- TFCE\_threshold, 24, 26, 28
- TFCE\_vertex\_analysis, 17, 25, 28
- TFCE\_vertex\_analysis\_mixed, 17, 26, 27
- VWRfirstrun, 29