Package ‘ciftiTools’

August 19, 2021

Type Package

Title Tools for Reading, Writing, Viewing and Manipulating CIFTI Files

Version 0.4.3

Maintainer Amanda Mejia <mandy.mejia@gmail.com>

Description CIFTI files contain brain imaging data in ``grayordinates,'' which represent the gray matter as cortical surface vertices (left and right) and subcortical voxels (cerebellum, basal ganglia, and other deep gray matter). 'ciftiTools' provides a unified environment for reading, writing, visualizing and manipulating CIFTI-format data. It supports the "dscalar," "dlabel," and "dtseries" intents. Greyordinate data is read in as a "xifti" object, which is structured for convenient access to the data and metadata, and includes support for surface geometry files to enable spatially-dependent functionality such as static or interactive visualizations and smoothing.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

Imports fields, gifti (>= 0.7.5), grDevices, oro.nifti, RNifti, RColorBrewer, rgl, viridisLite, xml2

Suggests covr, ggplot2, ggpubr, htmlwidgets, knitr, rmarkdown, papayar, testthat (>= 2.1.0)

RoxygenNote 7.1.1

URL https://github.com/mandymejia/ciftiTools

BugReports https://github.com/mandymejia/ciftiTools/issues

NeedsCompilation no

Author Amanda Mejia [aut, cre], Damon Pham [aut] (<https://orcid.org/0000-0001-7563-4727>), John Muschelli [ctb] (<https://orcid.org/0000-0001-6469-1750>)

Repository CRAN

Date/Publication 2021-08-19 13:40:09 UTC
R topics documented:

- add_surf .................................................. 3
- apply_xifti ................................................ 4
- as.matrix.xifti ......................................... 5
- as.xifti ..................................................... 5
- ciftiTools .................................................. 8
- ciftiTools.files ......................................... 10
- ciftiTools.getOption ..................................... 10
- ciftiTools.listOptions ................................. 11
- ciftiTools.setOption .................................... 11
- combine_xifti ............................................ 12
- dim.xifti .................................................. 12
- expand_color_pal ........................................ 13
- fix_xifti .................................................. 13
- get_wb_cmd_path ........................................ 14
- info_cifti ................................................ 14
- is.cifti .................................................... 16
- is.surf ..................................................... 18
- is.xifti .................................................... 18
- load_parc ................................................ 20
- load_surf .................................................. 21
- make_color_pal ........................................... 22
- merge_xifti .............................................. 23
- move_from_mwall ......................................... 24
- move_to_mwall ............................................ 25
- newdata_xifti ............................................ 26
- parc_borders ............................................. 26
- plot.surf .................................................. 27
- plot.xifti ................................................ 28
- read_cifti ............................................... 28
- read_surf ................................................ 32
- read_xifti2 .............................................. 32
- remove_xifti ............................................. 34
- resample_cifti ......................................... 35
- resample_cifti_from_template .......................... 37
- resample_gifti ......................................... 38
- resample_surf ........................................... 39
- rotate_surf .............................................. 40
- ROY_BIG_BL .............................................. 41
- run_wb_cmd ............................................... 42
- select_xifti ............................................. 42
- separate_cifti ......................................... 43
- smooth_cifti ............................................ 45
- smooth_gifti ............................................ 48
- substructure_table ...................................... 49
- summary.surf ........................................... 50
- summary.xifti .......................................... 50
**add_surf**

---

`add_surf` **Add surface(s) to a "xifti"**

**Description**

Add left or right cortical surface geometry to a "xifti" object.

**Usage**

```r
add_surf(xifti, surfL = NULL, surfR = NULL)
```

**Arguments**

- `xifti` A "xifti" object.
- `surfL` (Optional) Left brain surface model. Can be a file path to a GIFTI surface geometry file (ends in "*.surf.gii"), a "gifti" object representing surface geometry, or a "surf" object.
- `surfR` (Optional) Right brain surface model. Can be a file path to a GIFTI surface geometry file (ends in "*.surf.gii"), a "gifti" object representing surface geometry, or a "surf" object.

**Details**

`surfL` will be added to `xifti$surf$cortex_left` and `surfR` will be added to `xifti$surf$cortex_right`. Any existing surfaces will be overwritten.

**Value**

the "xifti" object with added surface geometry components.
apply_xifti

See Also

Other functions for manipulating 'xifti' objects: apply_xifti(), combine_xifti(), convert_to_dlabel(), merge_xifti(), newdata_xifti(), remove_xifti(), select_xifti(), transform_xifti()

---

**Description**

Apply a many-to-N function (e.g. mean) to the rows or columns of a "xifti". If applied row-wise, a "xifti" with N data column(s) is returned. (If the "xifti" had the dlabel intent, and values that are not labels are created, then it is converted to dscalar.) If applied column-wise, a numeric matrix with N rows is returned.

For univariate functions, use transform_xifti instead.

**Usage**

apply_xifti(xifti, margin = c(1, 2), FUN, ...)

**Arguments**

- **xifti**: A "xifti" object.
- **margin**: The dimension along which to apply FUN: 1 for rows (default) and 2 for columns.
- **FUN**: The function. It should take in a numeric vector and return a length-N numeric vector.
- **...**: Additional arguments to FUN

**Value**

A "xifti" if margin == 1, or a numeric matrix if margin == 2

**See Also**

Other functions for manipulating 'xifti' objects: add_surf(), combine_xifti(), convert_to_dlabel(), merge_xifti(), newdata_xifti(), remove_xifti(), select_xifti(), transform_xifti()
as.matrix.xifti

Convert a "xifti" to a matrix

Description

Converts a "xifti" to a matrix by concatenating the data from each brainstructure along the rows. Surfaces and metadata are discarded.

Usage

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

Arguments

x
A "xifti" object.

... Unused

Value

The input as a matrix. Each brainstructure’s data is concatenated.

as.xifti

Assemble a "xifti" from data

Description

Assembles cortical data, subcortical data, and/or surface geometry to form a "xifti". The inputs must be data objects (vectors, matrices or arrays, depending on the argument).

Usage

as.xifti(
cortexL = NULL,
cortexL_mwall = NULL,
cortexR = NULL,
cortexR_mwall = NULL,
mwall_values = c(NA, NaN),
subcortVol = NULL,
subcortLabs = NULL,
subcortMask = NULL,
surfL = NULL,
surfR = NULL,
col_names = NULL,
HCP_32k_auto_mwall = TRUE,

... Not used...

... Not used...

... Not used...

... Not used...
as.xifti

as.xifti(
    validate = TRUE
)

as_cifti(
    cortexL = NULL,
    cortexL_mwall = NULL,
    cortexR = NULL,
    cortexR_mwall = NULL,
    mwall_values = c(NA, NaN),
    subcortVol = NULL,
    subcortLabs = NULL,
    surfL = NULL,
    surfR = NULL
)

as.cifti(
    cortexL = NULL,
    cortexL_mwall = NULL,
    cortexR = NULL,
    cortexR_mwall = NULL,
    mwall_values = c(NA, NaN),
    subcortVol = NULL,
    subcortLabs = NULL,
    surfL = NULL,
    surfR = NULL
)

Arguments

cortexL, cortexL_mwall

Left cortex data and ROI. Each must be a data matrix or vector.

If cortexL_mwall is not provided, cortexL should have data for all vertices on the left cortical surface ($V_{L \times T}$ data matrix). There will not be a mask for the medial wall. Not providing the medial wall mask is appropriate for ".dlabels.nii"
files where the medial wall may have its own label and therefore should not be treated as missing data.

If cortexL_mwall is provided, cortexL should either have data for all vertices on the left cortical surface ($V_L^L x T$ data matrix, with filler values e.g. 0 or NaN for medial wall vertices), or have data only for non-medial wall vertices ($(V_L - mwall_L)^L x T$ data matrix). The medial wall mask will be the 0 values in cortexL_mwall. The medial wall mask should be provided whenever the medial wall should be treated as missing data.

Since the unmasked cortices must have the same number of vertices, $V_L$ should match $V_R$.

cortexR, cortexR_mwall

Right cortex data and ROI. Each must be a data matrix or vector.

If cortexR_mwall is not provided, cortexR should have data for all vertices on the right cortical surface ($V_R^R x T$ data matrix) or it will not be a mask for the medial wall. Not providing the medial wall mask is appropriate for ".dlabels.nii" files where the medial wall may have its own label and therefore should not be treated as missing data.

If cortexR_mwall is provided, cortexR should either have data for all vertices on the right cortical surface ($V_R^R x T$ data matrix, with filler values e.g. 0 or NaN for medial wall vertices), or have data only for non-medial wall vertices ($(V_R - mwall_R)^R x T$ data matrix). The medial wall mask will be the 0 values in cortexR_mwall. The medial wall mask should be provided whenever the medial wall should be treated as missing data.

Since the unmasked cortices must have the same number of vertices, $V_L$ should match $V_R$.

mwall_values

If cortex[L/R]_mwall was not provided, or if it was invalid (i.e. bad length or all TRUE), the medial wall mask will be inferred from rows in cortex[L/R] that are constantly one of these values. Default: c(NA,NaN). If NULL, do not attempt to infer the medial wall from the data values. NULL should be used if NA or NaN are legitimate values that non-medial wall vertices might take on.

subcortVol, subcortLabs, subcortMask

subcortVol represents the data values of the subcortex. It is either a 3D/4D numeric array ($i x j x k x T$), or a vectorized matrix ($V_S$ voxels by $T$ measurements). If it’s vectorized, the voxels should be in spatial order ($i$ index increasing fastest, then $j$, then $k$).

subcortLabs represents the brainstructure labels of each voxel: see substructure_table. It is either a 3D data array ($i x j x k$) of integer brainstructure indices, or a $V_S$ length vector in spatial order with brainstructure names as factors or integer indices. The indices should be 3-21 (1 and 2 correspond to left and right cortex, respectively) or 1-19 (cortex labels omitted), with 0 representing out-of-mask voxels.

subcortMask is logical 3D data array ($i x j x k$) where TRUE values indicate subcortical voxels (in-mask). If it is not provided, the mask will be inferred from voxels with labels 0, NA, or NaN in subcortLabs. If subcortLabs are vectorized and subcortMask is not provided, the mask cannot be inferred so an error will occur.
surfL, surfR  (Optional) Surface geometries for the left or right cortex. Can be a surface GIFTI file path or "surf" object; see `make_surf` for a full description of valid inputs.
col_names   Names of each measurement/column in the data.
HCP_32k_auto_mwall

If left and/or right cortex data is provided, and the number of vertices matches that of the HCP 32k mesh (29696 on left, and 29716 on right), should the medial wall masks be added to the "xifti" if not provided? Default: TRUE.
validate    Validate that the result is a "xifti"? Default: TRUE. If FALSE, the result may not be properly formatted if the inputs were invalid.

Details

Each data or surface component is optional. Metadata components (cortex[L/R]_mwall, subcortLabs, and subcortMask) will be ignored if its corresponding data component is not provided. If no data or surface components are provided, then the template_xifti will be returned.

If cortical data are provided without a corresponding medial wall mask, or if the provided mask is invalid or empty, then the medial wall will be inferred from data rows that are constantly a value in mwall_values. But if mwall_values is NULL, no attempt to infer the medial wall will be made and the medial wall metadata entry will be NULL.

The total number of greyordinates will be \( G = (V_L - mwall_L) + (V_R - mwall_R) + V_S \): \( V_L - mwall_L \) left vertices, \( V_R - mwall_R \) right vertices and \( V_S \) subcortical voxels. \( T \), the total number of measurements (columns of data), must be the same for each brainstructure.

Value

A "xifti"

See Also

Other functions for reading in CIFTI or GIFTI metric data: `info_cifti()`, `load_parc()`, `load_surf()`, `read_cifti()`, `read_surf()`, `read_xifti2()`

ciftiTools

ciftiTools: Tools for Reading and Visualizing CIFTI Brain Files

Description

Here are groups of commonly-used functions in ciftiTools:

Details

Functions for reading in CIFTI or GIFTI metric data:

- `read_xifti`: Read in a CIFTI file as a "xifti"
- `read_xifti2`: Read in GIFTI files as a "xifti"
- `as.xifti`: Combine numeric data to form a "xifti"
• read_surf: Read in a surface GIFTI file as a "surf"
• info_cifti: Read the metadata in a CIFTI file
• load_surf: Read in a surface geometry included in ciftiTools
• load_parc: Read in a dlabel parcellation included in ciftiTools

Functions for writing CIFTI or GIFTI metric data:

• write_cifti: Write a "xifti" to a CIFTI file
• write_metric_gifti: Write a data matrix to a metric GIFTI file
• write_surf_gifti: Write a "surf" to a surface GIFTI file
• write_subcort_nifti: Write subcortical data to NIFTI files
• separate_cifti: Separate a CIFTI file into GIFTI and NIFTI files

Functions for manipulating "xifti"s:

• apply_xifti: Apply a function along the rows or columns of a "xifti"
• combine_xifti: Combine "xifti"s with non-overlapping brain structures
• concat_xifti: Concatenate "xifti"s
• convert_xifti: Convert the intent of a "xifti"
• newdata_xifti: Replace the data in a "xifti"
• remove_xifti: Remove a component from a "xifti"
• select_xifti: Select columns of a "xifti"
• transform_xifti: Apply a univariate transformation to a "xifti" or pair of "xifti"s
• add_surf: Add surfaces to a "xifti"

Functions for working with surfaces:

• read_surf: Read in a surface GIFTI file as a "surf"
• is.surf: Verify a "surf"
• write_surf_gifti: Write a "surf" to a surface GIFTI file
• view_surf: Visualize a "surf"
• resample_surf: Resample a "surf"
• rotate_surf: Rotate the geometry of a "surf"
ciftiTools.files

description
cifti and surface GIFTI files included in the ciftiTools package

usage
ciftiTools.files()

details
the CIFTI files are from NITRC: cifti-2_test_data-1.2.zip at https://www.nitrc.org/frs/?group_id=454
the surfaces are from the HCP and are included according to these data use terms: Data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.
only the inflated surfaces are available as GIFTI files. To access the other surfaces included in the package (very inflated and midthickness), see load_surf.

value
a list of file paths

ciftiTools.getOption

get a ciftiTools option

description
gets an R option (with prefix "ciftiTools_") value. See ciftiTools.listOptions.

usage
ciftiTools.getOption(opt)

arguments
opt the option.

value
the value, val
ciftiTools.listOptions  

*List ciftiTools options*

**Description**

List ciftiTools options

**Usage**

ciftiTools.listOptions()

**Value**

data.frame describing the options

---

ciftiTools.setOption  

*Set a ciftiTools option*

**Description**

Sets an R option (with prefix "ciftiTools_”). See ciftiTools.listOptions.

**Usage**

ciftiTools.setOption(opt, val)

**Arguments**

opt          The option.
val          The value to set the option as.

**Value**

The new value, val
### combine_xifti

**Combine "xifti"s with non-overlapping brain structures**

**Description**

Combine "xifti"s with non-overlapping brain structures into a single "xifti". The names and intent of the first will be used (if present).

**Usage**

```r
combine_xifti(..., xii_list = NULL, meta = c("first", "all"))
```

**Arguments**

- `...`: The "xifti" objects
- `xii_list`: Alternatively, a list of "xifti" objects. If specified, will ignore ...
- `meta`: "first" (default) to just use the metadata from the first argument, or "all" to include the other metadata in a list.

**Value**

A "xifti" with data from the inputs

**See Also**

Other functions for manipulating 'xifti' objects: `add_surf()`, `apply_xifti()`, `convert_to_dlabel()`, `merge_xifti()`, `newdata_xifti()`, `remove_xifti()`, `select_xifti()`, `transform_xifti()`

### dim.xifti

**Dimensions of a "xifti"**

**Description**

Returns the number of rows (vertices + voxels) and columns (measurements) in the "xifti" data.

**Usage**

```r
## S3 method for class 'xifti'
dim(x)
```

**Arguments**

- `x`: A "xifti" object.

**Value**

The number of rows and columns in the "xifti" data.
expand_color_pal

Interpolates between entries in the input palette to make a larger palette with COLOR_RES entries.

Description

Interpolates between entries in the input palette to make a larger palette with COLOR_RES entries.

Usage

expand_color_pal(pal, COLOR_RES = 255)

Arguments

- **pal**: The color palette to expand, as a data.frame with two columns: "color" (character: color hex codes) and "value" (numeric).
- **COLOR_RES**: The number of entries to have in the output palette.

Value

A data.frame with two columns: "color" (character: color hex codes) and "value" (numeric)

fix_xifti

Fix a "xifti"

Description

Make adjustments to a putative "xifti" so that it is valid. Each adjustment is reported.

Usage

fix_xifti(xifti, verbose = TRUE)

Arguments

- **xifti**: A "xifti" object.
- **verbose**: Report each adjustment? Default: TRUE

Details

Right now it only coerces the data to numeric matrices.

Value

The fixed "xifti"
get_wb_cmd_path  
*Get the Connectome Workbench command path*

**Description**

Retrieves the path to the Connectome Workbench executable from a file path that may point to the executable itself, or to the Workbench folder which contains it (i.e., "path/to/workbench/bin_linux64/wb_command" or "path/to/workbench").

**Usage**

get_wb_cmd_path(wb_path)

**Arguments**

- **wb_path** (Optional) Path to the Connectome Workbench folder or executable.

**Value**

The path to the Connectome Workbench executable

---

info_cifti  
*Get CIFTI metadata*

**Description**

Get CIFTI metadata from the NIFTI header and XML using the Connectome Workbench command -nifti-information. The information is formatted as the meta component in a "xifti" object (see template_xifti), and includes:

1. medial wall masks for the left and right cortex
2. the subcortical labels (ordered spatially)
3. the subcortical mask
4. other NIFTI intent-specific metadata

**Usage**

info_cifti(cifti_fname)

cifti_info(cifti_fname)
infoCIfTI(cifti_fname)

**Arguments**

- **cifti_fname** File path to a CIFTI file (ending in ".d*.nii").
Details

Additional metadata depends on the type of CIFTI file:

1. "dtseries"
   (a) time_start: Start time
   (b) time_step: The TR
   (c) time_unit: Unit of time
2. "dscalar"
   (a) names: Name of each data column
3. "dlabels"
   (a) names: (Names of each data column.)
   (b) labels: (List of Lx5 data.frames. Row names are the label names. Column names are Key, Red, Green, Blue, and Alpha. List entry names are the names of each data column.)

Value

The metadata component of a "xifti" for the input CIFTI file

Connectome Workbench

This function interfaces with the "-nifti-information" Workbench command.

Label Levels

xifti$meta$subcort$labels is a factor with the following levels:

1. Cortex-L
2. Cortex-R
3. Accumbens-L
4. Accumbens-R
5. Amygdala-L
6. Amygdala-R
7. Brain Stem
8. Caudate-L
9. Caudate-R
10. Cerebellum-L
11. Cerebellum-R
12. Diencephalon-L
13. Diencephalon-R
14. Hippocampus-L
15. Hippocampus-R
16. Pallidum-L
17. Pallidum-R
18. Putamen-L
19. Putamen-R
20. Thalamus-L
21. Thalamus-R

These correspond to the same structures as given by `ft_read_cifti` in the cifti-matlab MATLAB toolbox. Note that the first two levels (left and right cortex) are not used.

**See Also**

Other functions for reading in CIFTI or GIFTI metric data: `as.xifti()`, `load_parc()`, `load_surf()`, `read_cifti()`, `read_surf()`, `read_xifti2()`

---

**is.cifti**

*Validate a "xifti" object*

**Description**

Check if object is valid for a "xifti". This alias for `is.xifti` is offered as a convenience, and a message will warn the user. We recommend using `is.xifti` instead.

**Usage**

```r
is.cifti(x, messages = TRUE)
```

**Arguments**

- `x` The putative "xifti".
- `messages` If `x` is not a "xifti", print messages explaining the problem? Default is `TRUE`.

**Details**

Requirements: it is a list with the same structure as `template_xifti`. The size of each data entry must be compatible with its corresponding mask (medial wall for the cortex and volumetric mask for the subcortex). Metadata should be present if and only if the corresponding data is also present. The surfaces can be present whether or not the cortex data are present.

See the "Label Levels" section for the requirements of `xifti$meta$subcort$labels`.

**Value**

Logical. Is `x` a valid "xifti"?
Label Levels

\texttt{xifti$\text{meta$subcort$labels}} is a factor with the following levels:

1. Cortex-L
2. Cortex-R
3. Accumbens-L
4. Accumbens-R
5. Amygdala-L
6. Amygdala-R
7. Brain Stem
8. Caudate-L
9. Caudate-R
10. Cerebellum-L
11. Cerebellum-R
12. Diencephalon-L
13. Diencephalon-R
14. Hippocampus-L
15. Hippocampus-R
16. Pallidum-L
17. Pallidum-R
18. Putamen-L
19. Putamen-R
20. Thalamus-L
21. Thalamus-R

These correspond to the same structures as given by \texttt{ft_read_cifti} in the \texttt{cifti-matlab} MATLAB toolbox. Note that the first two levels (left and right cortex) are not used.

See Also

Other commonly-used functions: \texttt{read_cifti()}, \texttt{resample_cifti()}, \texttt{smooth_cifti()}, \texttt{view_xifti_surface()}, \texttt{view_xifti_volume()}, \texttt{write_cifti()}

is.surf

Validate a "surf" object (vertices + faces)

Description

Check if object is valid for xifti$surf$cortex_left or xifti$surf$cortex_right, where xifti is a "xifti" object.

Usage

is.surf(x)

Arguments

x

The putative "surf".

Details

This is a helper function for is.xifti.

Requirements: the "surf" must be a list of three components: "vertices", "faces", and "hemisphere". The first two should each be a numeric matrix with three columns. The values in "vertices" represent spatial coordinates whereas the values in "faces" represent vertex indices defining the face. Thus, values in "faces" should be integers between 1 and the number of vertices. The last list entry, "hemisphere", should be "left", "right", or NULL indicating the brain hemisphere which the surface represents.

Value

Logical. Is x a valid "surf"?

See Also

Other functions for working with GIFTI surface geometry data: read_surf(), resample_surf(), rotate_surf(), view_surf(), write_surf_gifti()

is.xifti

Validate a "xifti" object.

Description

Check if object is valid for a "xifti" object.

Usage

is.xifti(x, messages = TRUE)

is_xifti(x, messages = TRUE)
is.xifti

Arguments

- **x**  
  The putative "xifti" object.

- **messages**  
  If x is not a "xifti" object, print messages explaining the problem? Default is TRUE.

Details

Requirements: it is a list with the same structure as template_xifti. The size of each data entry must be compatible with its corresponding mask (medial wall for the cortex and volumetric mask for the subcortex). Metadata should be present if and only if the corresponding data is also present. The surfaces can be present whether or not the cortex data are present.

See the "Label Levels" section for the requirements of xifti$meta$subcort$labels.

Value

Logical. Is x a valid "xifti" object?

Label Levels

xifti$meta$subcort$labels is a factor with the following levels:

1. Cortex-L
2. Cortex-R
3. Accumbens-L
4. Accumbens-R
5. Amygdala-L
6. Amygdala-R
7. Brain Stem
8. Caudate-L
9. Caudate-R
10. Cerebellum-L
11. Cerebellum-R
12. Diencephalon-L
13. Diencephalon-R
14. Hippocampus-L
15. Hippocampus-R
16. Pallidum-L
17. Pallidum-R
18. Putamen-L
19. Putamen-R
20. Thalamus-L
21. Thalamus-R

These correspond to the same structures as given by ft_read_cifti in the cifti-matlab MATLAB toolbox. Note that the first two levels (left and right cortex) are not used.
load_parc  

Load a parcellation included in ciftiTools.

Description

Load a parcellation included in ciftiTools.

Usage

load_parc(
  name = c("Schaefer_100", "Schaefer_400", "Schaefer_1000", "Yeo_7", "Yeo_17")
)

Arguments

name

The name of the parcellation to load:

- "Schaefer_100": (2018) 100 parcels based on the "local-global" approach.
- "Schaefer_400": (2018) 400 parcels based on the "local-global" approach.
- "Schaefer_1000": (2018) 1000 parcels based on the "local-global" approach.
- "Yeo_7": (2011) 7 networks based on fcMRI clustering. Networks are further divided into 51 components.
- "Yeo_17": (2011) 17 networks based on fcMRI clustering. Networks are further divided into 114 components.

NULL (default) will load the first choice, where applicable. This argument will affect the indices, colors, and names of each parcel, but not the parcel boundaries.

Details

When using these parcellations, please cite the corresponding paper(s):


Note that the Schaefer parcels have been matched to networks from Kong (2021+).

Value

The parcellation as a dlabel "xifti" with one column. Each key represents one unique parcel.
load_surf

See Also
Other functions for reading in CIFTI or GIFTI metric data: as.xifti(), info_cifti(), load_surf(), read_cifti(), read_surf(), read_xifti2()

load_surf

Load a "surf" included in ciftiTools

Description
Load a "surf" object from one of the three 32k surface geometries included in ciftiTools.

Usage
load_surf(
  hemisphere = c("left", "right"),
  name = c("inflated", "very inflated", "midthickness"),
  resamp_res = NULL
)

Arguments
  hemisphere "left" (default) or "right"
  name The name of the surface geometry to load: "inflated" (default), "very inflated", and "midthickness".
  resamp_res The resolution to resample the surfaces to. If NULL (default) or 32492, do not resample.

Details
The surfaces are from the HCP and are included according to these data use terms: Data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

Value
The "surf" object

See Also
Other functions for reading in CIFTI or GIFTI metric data: as.xifti(), info_cifti(), load_parc(), read_cifti(), read_surf(), read_xifti2()
# make_color_pal

**Make a color palette.**

## Description

Control the mapping of values to colors with `colors`, `color_mode`, and `zlim`.

## Usage

```r
make_color_pal(
  colors = NULL,
  color_mode = c("sequential", "qualitative", "diverging"),
  zlim = NULL
)
```

## Arguments

- **colors** *(Optional)*: "ROY_BIG_BL", the name of a ColorBrewer palette (see `RColorBrewer::brewer.pal.info` and `colorbrewer2.org`), the name of a viridisLite palette, or a character vector of color names. NULL (default) will use "ROY_BIG_BL" if `color_mode` is "sequential" or "diverging", and "Set2" if `color_mode` is "qualitative". See the description for more details.

- **color_mode** *(Optional)*: "sequential", "qualitative", or "diverging". Default: "sequential". See the description for more details.

- **zlim** *(Optional)*: Controls the mapping of values to each color in `colors`. See the description for more details.

## Details

There are three kinds of arguments for `colors`: "ROY_BIG_BL", the name of a ColorBrewer palette (see `RColorBrewer::brewer.pal.info` and `colorbrewer2.org`), the name of a viridisLite palette, or a character vector of color names.

If `colors=="ROY_BIG_BL"`, the "ROY_BIG_BL" palette will be used. It is the same palette as the default for the Connectome Workbench application (https://github.com/Washington-University/workbench/blob/master/src/Files/PaletteFile.cxx). The midpoint will be colored black. From the midpoint toward the upper bound, colors will proceed from black to red to yellow. From the midpoint toward the lower bound, colors will proceed from black to blue to purple to green to aqua. Here is how each color mode behaves if `colors=="ROY_BIG_BL"`:

- **color_mode=="sequential"** Only half of the palette will be used. If `zlim` is length 2, the higher value will be the maximum and the lower value will be the minimum. Set `zlim[1] > zlim[2]` to reverse the color scale. (Note that the second half, black -> red -> yellow, is used by default. To use the negative half specify `colors=="ROY_BIG_BL_neg"` instead. It will also be used automatically by `xifti_read_surface` when the data range is negative.) `zlim` can also be length 10, in which case each value corresponds to the position of an individual color in the half palette.
color_mode=="qualitative" "ROY_BIG_BL" is not recommended for qualitative data, so a warning will be issued. Palette colors will be selected from the landmark "ROY_BIG_BL" colors, with interpolated colors added if the number of colors in the palette (18) is less than this range. zlim should be a single number: the number of unique colors to get.

color_mode=="diverging" If zlim is length 2 or 3, the lowest number will be the lower bound and the highest number will be the upper bound. If zlim is length 3, the middle number will be the midpoint (black). The lower and upper bounds will be aqua and yellow, respectively, except if zlim is in descending order, in which case the color scale will be reversed (lowest is yellow; highest is aqua). zlim can also be length 19, in which case each value corresponds to the position of an individual color in the palette.

If colors is the name of an RColorBrewer palette (see RColorBrewer::brewer.pal.info) or viridisLite palette, the colors in that palette will be used, and the following behavior applies. If colors is a character vector of color names (hex codes or standard R color names), the following behavior applies directly:

color_mode=="sequential" If zlim is length 2, the higher value will be the maximum and the lower value will be the minimum. Set zlim[1] > zlim[2] to reverse the color scale. zlim can also be the same length as the palette, in which case each value corresponds to the position of an individual color in the palette.

color_mode=="qualitative" zlim should be a single number: the number of unique colors to get. Color interpolation will be used if the number of colors in the palette is less than this range. If length(zlim)==length(colors), each color will be mapped to each corresponding value.

color_mode=="diverging" If zlim is length 2 or 3, the lowest number will be the lower bound and the highest number will be the upper bound. If zlim is length 3, the middle number will be the midpoint. Set zlim in descending order to reverse the color scale. zlim can also be the same length as the palette, in which case each value corresponds to the position of an individual color in the palette.

Value

A data.frame with two columns: "color" (character: color hex codes) and "value" (numeric)

merge_xifti

Concatenate "xifti"s

Description

Concatenate "xifti" objects along the columns. They must have the same brainstructures and resolutions. The first "xifti"'s metadata will be retained, including its intent.

Usage

merge_xifti(..., xifti_list = NULL)
Arguments

..., xifti_list

Provide as arguments the "xifti"s to concatenate, OR the single argument xifti_list which should be a list of "xifti"s. (If xifti_list is provided all other inputs will be ignored.)

Value

The concatenated "xifti"

See Also

Other functions for manipulating 'xifti' objects: add_surf(), apply_xifti(), combine_xifti(), convert_to_dlabel(), newdata_xifti(), remove_xifti(), select_xifti(), transform_xifti()

---

move_from_mwall: Move data locations from medial wall

Description

Move all medial wall locations into the cortical data matrices by assigning them a specific value (e.g. NA).

Usage

move_from_mwall(xifti, value = NA, name = "Medial_Wall", RGBA = c(1, 1, 1, 0))

Arguments

- xifti: A "xifti" object.
- value: The value to assign the medial wall locations. Default: NA.
- name, RGBA: Only used if the "xifti" has the dlabel intent and value is not an already-existing Key. This is the name to assign to the new key for the medial wall locations, as well as a length-four numeric vector indicating the red, green, blue, and alpha values for the color to assign to the new key. These will be reflected in the updated label table. Note that RGBA values must all be in [0, 1]. Currently, only one name and set of RGBA values are supported, meaning that the medial wall locations will have the same Key, name, and color across all data columns in the "xifti". An error will occur if the Key already exists for some columns but not others. Defaults: "Medial_Wall" for "name" and white with 0 alpha for RGBA.

Value

The "xifti" with re-organized data and medial wall masks
move_to_mwall

See Also

move_to_mwall
unmask_cortex

move_to_mwall

Move data locations to the medial wall

Description

Move cortical data locations with a specific value(s) to the medial wall mask. For example, dlabel CIFTIs often have medial wall vertices set to a specific key value, rather than a medial wall mask. This function can move those data locations from the data matrix to the medial wall mask in the metadata.

Usage

move_to_mwall(xifti, values = c(NA, NaN), drop = FALSE)

Arguments

xifti A "xifti" object.
values Medial wall values. Default: NA and NaN. Data locations in the left and right cortex that are one of these values (across all columns) will be moved to the medial wall mask in the metadata.
drop Only used if the "xifti" has the dlabel intent. Drop the key(s) in values from the label tables, for columns in which they no longer exist? Default: FALSE.

Value

The "xifti" with re-organized data and medial wall masks

See Also

move_from_mwall
newdata_xifti  Replace the data in a "xifti"

Description
Replace the data in a "xifti" with new data from a data matrix.

Usage
newdata_xifti(xifti, newdata, newnames = NULL)

Arguments
- xifti: The "xifti"
- newdata: The V \times T matrix of data values to replace those in xifti with. The left cortex vertices should be at the top, right cortex vertices in the middle, and subcortex vertices at the bottom (when present). Can also be a length-one vector to set all values equally.
- newnames: Replace the names in the xifti. If NULL (default), keep the original names.

Details
If the "xifti" has V greyordinates and T timepoints in total, newdata should be a V \times T matrix.

Value
The new "xifti"

See Also
Other functions for manipulating 'xifti' objects: add_surf(), apply_xifti(), combine_xifti(), convert_to_dlabel(), merge_xifti(), remove_xifti(), select_xifti(), transform_xifti()

parc_borders  Parcellation borders

Description
Identify vertices which lie on the border of different parcels.

Usage
parc_borders(parc, surf = NULL, hemisphere = c("left", "right"))
**Arguments**

- **parc**: Integer vector the same length as the number of vertices. Each entry indicates the parcel that vertex belongs to.
- **surf**: The surface which the vertices belong to, or just the "faces" component ($F \times 3$ matrix where each row indicates the vertices which comprise a face). If not provided, the (resampled) default hemisphere surface included with ciftiTools will be used.
- **hemisphere**: Only used to choose which default surface to use if `is.null(surf)`. Should be "left" (default) or "right".

**Value**

Logical vector the same length as `parc` indicating if the vertex lies on a border.

---

**Description**

Visualize a single surface

**Usage**

```r
## S3 method for class 'surf'
plot(x, ...)
```

**Arguments**

- **x**: A "surf" object
- **...**: Additional arguments to `view_xifti_surface`. But, the hemisphere argument behaves differently: it can be either left or right to indicate which hemisphere `x` represents. It is only used if the "hemisphere" metadata entry in `x` is NULL. If both the argument and the metadata entry are NULL, the surface will be treated as the left hemisphere.
plot.xifti

S3 method: use view_xifti to plot a "xifti" object

Description

S3 method: use view_xifti to plot a "xifti" object

Usage

## S3 method for class 'xifti'
plot(x, ...)

Arguments

x
A "xifti" object.
...
Additional arguments to view_xifti, except what, which will be set to NULL.

read_cifti

Read a CIFTI file

Description

Read in a CIFTI file as a "xifti" object.

Usage

read_cifti(
  cifti_fname = NULL,
  surfL_fname = NULL,
  surfR_fname = NULL,
  brainstructures = c("left", "right"),
  idx = NULL,
  resamp_res = NULL,
  flat = FALSE,
  mwall_values = c(NA, NaN),
  verbose = FALSE,
  ...
)

readCIfTI(
  cifti_fname = NULL,
  surfL_fname = NULL,
  surfR_fname = NULL,
  brainstructures = c("left", "right"),
  idx = NULL,
Arguments

cifti_fname  File path to a CIFTI file (ending in ".d*.nii").
surfL_fname  (Optional) File path to a GIFTI surface geometry file representing the left cortex.
surfR_fname  (Optional) File path to a GIFTI surface geometry file representing the right cortex.

brainstructures  Character vector indicating which brain structure(s) to obtain: "left" (left cortex), "right" (right cortex) and/or "subcortical" (subcortex and cerebellum). Can also be "all" (obtain all three brain structures). Default: c("left", "right") (cortex only).
If a brain structure is indicated but does not exist in the CIFTI file, a warning will occur and that brain structure will be skipped.

idx  Numeric vector indicating the data indices (columns) to read. If NULL (default), read in all the data. Must be a subset of the indices present in the file, or an error
will occur.
For high-resolution CIFTI files, reading in only a subset of the data saves memory, but will be slower than reading in the entire file due to the required intermediate steps.

**resamp_res** Resolution to resample the cortical data and surface to. Default: `NULL` (do not resample). If not `NULL`, the data will have to be read in with `-cifti-separate`, which is slower than `-cifti-convert -to-gifti-ext`.

**flat** Should the result be flattened into a single matrix?
If `FALSE` (default), the result will be a "xifti" object.
If `TRUE`, the result will be a $T \times G$ matrix ($T$ measurements, $G$ greyordinates not including the medial wall if it's excluded from the ROI). All below arguments will be ignored because the brain structures cannot be identified. Surfaces will not be appended. Resampling is also not possible. `flat==TRUE` is the fastest way to read in just the CIFTI data.
If `TRUE`, the greyordinates will be ordered by left cortex, right cortex, and then subcortex. Subcortical voxels will be ordered by alphabetical label. However, where each brainstructure (and subcortical structure) begins and ends cannot be determined. The medial wall locations and subcortical brain mask are also not included. The data matrix will be identical to that created by `-cifti-convert -to-gifti-ext`.

**mwall_values** If the medial wall locations are not indicated in the CIFTI, use these values to infer the medial wall mask. Default: `c(NA,NaN)`. If `NULL`, do not attempt to infer the medial wall.

**verbose** Should occasional updates be printed? Default: `FALSE`.

**...** Additional arguments to `read_cifti_convert` or `read_cifti_separate`.

**Details**
First, metadata is obtained with `info_cifti`. Then, if no resampling is requested, the `-cifti-convert -to-gifti-ext` Workbench Command is used to "flatten" the data and save it as a metric GIFTI file, which is read in and separated by brainstructure according to the metadata (`read_cifti_convert`). Otherwise, if sampling is requested, then the CIFTI is separated into its GIFTI and NIFTI components, resampled, and then re-assembled (`read_cifti_separate`). The former is much faster for large CIFTI files, so the latter is only used when necessary for resampling.
If `cifti_fname` is not provided, then only the surfaces are read in.

**Value**
If `!flat`, a "xifti" object. Otherwise, a $T \times G$ matrix ($T$ measurements, $G$ greyordinates).

**Connectome Workbench**
This function interfaces with the "-cifti-convert" Workbench command if resampling is not needed, and the "-cifti-separate" Workbench command if resampling is needed.
Label Levels

`xifti$meta$subcort$labels` is a factor with the following levels:

1. Cortex-L
2. Cortex-R
3. Accumbens-L
4. Accumbens-R
5. Amygdala-L
6. Amygdala-R
7. Brain Stem
8. Caudate-L
9. Caudate-R
10. Cerebellum-L
11. Cerebellum-R
12. Diencephalon-L
13. Diencephalon-R
14. Hippocampus-L
15. Hippocampus-R
16. Pallidum-L
17. Pallidum-R
18. Putamen-L
19. Putamen-R
20. Thalamus-L
21. Thalamus-R

These correspond to the same structures as given by `ft_read_cifti` in the cifti-matlab MATLAB toolbox. Note that the first two levels (left and right cortex) are not used.

See Also

Other commonly-used functions: `is.cifti()`, `resample_cifti()`, `smooth_cifti()`, `view_xifti_surface()`, `view_xifti_volume()`, `write_cifti()`

Other functions for reading in CIFTI or GIFTI metric data: `as.xifti()`, `info_cifti()`, `load_parc()`, `load_surf()`, `read_surf()`, `read_xifti2()`
Description

Coerce a file path to a surface GIFTI, a "gifti" object, a list with entries "pointset" and "triangle", or a "surf" to a "surf".

Usage

read_surf(surf, expected_hemisphere = NULL, resamp_res = NULL)

make_surf(surf, expected_hemisphere = NULL, resamp_res = NULL)

Arguments

surf Either a file path to a surface GIFTI; a "gifti" read by readgii; a list with entries "pointset" and "triangle"; or, a "surf" object.

expected_hemisphere The expected hemisphere ("left" or "right") of surf. If the hemisphere indicated in the GIFTI metadata is the opposite, an error is raised. If NULL (default), use the GIFTI hemisphere.

resamp_res The resolution to resample the surfaces to. If NULL (default), do not resample.

Value

The "surf": a list with components "vertices" (3D spatial locations), "faces" (defined by three vertices), and "hemisphere" ("left", "right", or NULL if unknown).

See Also

Other functions for reading in CIFTI or GIFTI metric data: as.xifti(), info_cifti(), load_parc(), load_surf(), read_cifti(), read_xifti2()

Other functions for working with GIFTI surface geometry data: is.surf(), resample_surf(), rotate_surf(), view_surf(), write_surf_gifti()

Description

Read in gifti metric files as a "xifti" object. May also include surface geometry gifti files and perform resampling.
Usage

read_xifti2(
  cortexL = NULL,
  cortexL_mwall = NULL,
  cortexR = NULL,
  cortexR_mwall = NULL,
  mwall_values = c(NA, NaN),
  surfL = NULL,
  surfR = NULL,
  resamp_res = NULL,
  col_names = NULL,
  HCP_32k_auto_mwall = TRUE,
  read_dir = NULL,
  validate = TRUE
)

Arguments

  cortexL, cortexL_mwall
  Left cortex data and ROI. Each must be a path to a GIFTI file.
  If cortexL_mwall is not provided, cortexL should have data for all vertices on
  the left cortical surface (V_LxT data matrix). There will not be a mask for the
  medial wall. Not providing the medial wall mask is appropriate for ".dlabels.nii"
  files where the medial wall may have its own label and therefore should not be
  treated as missing data.
  If cortexL_mwall is provided, cortexL should either have data for all vertices
  on the left cortical surface (V_LxT data matrix, with filler values e.g. 0
  or NaN for medial wall vertices), or have data only for non-medial wall vertices
  ((V_L - mwall_L)xT data matrix). The medial wall mask will be the 0 values
  in cortexL_mwall. The medial wall mask should be provided whenever the
  medial wall should be treated as missing data.
  Since the unmasked cortices must have the same number of vertices, V_L should
  match V_R, or resamp_res must be set.

  cortexR, cortexR_mwall
  Right cortex data and ROI. Each must be a path to a GIFTI file.
  If cortexR_mwall is not provided, cortexR should have data for all vertices on
  the right cortical surface (V_RxT data matrix, with filler values e.g. 0 or
  NaN for medial wall vertices). Not providing the medial wall mask is appropriate for ".dlabels.nii"
  files where the medial wall may have its own label and therefore should not be treated as
  missing data.
  If cortexR_mwall is provided, cortexR should either have data for all vertices
  on the right cortical surface (V_RxT data matrix, with filler values e.g. 0 or
  NaN for medial wall vertices), or have data only for non-medial wall vertices
  ((V_R - mwall_R)xT data matrix). The medial wall mask will be the 0 values
  in cortexR_mwall. The medial wall mask should be provided whenever the
  medial wall should be treated as missing data.
  Since the unmasked cortices must have the same number of vertices, V_L should
  match V_R, or resamp_res must be set.
remove_xifti

mwall_values  If cortex[L/R]_mwall was not provided, or if it was invalid (i.e. bad length or all TRUE), the medial wall mask will be inferred from rows in cortex[L/R] that are constantly one of these values. Default: c(NA,NaN). If NULL, do not attempt to infer the medial wall from the data values. NULL should be used if NA or NaN are legitimate values that non-medial wall vertices might take on.

surfL, surfR  (Optional) File path(s) to surface GIFTI(s) for the left or right cortex.

resamp_res  Resolution to resample the cortical data and surface to. Default: NULL (do not resample). If provided, the original resolutions of the cortex data and surfaces may differ.

col_names  Names of each measurement/column in the data. Overrides names indicated in the data components.

HCP_32k_auto_mwall  If left and/or right cortex data is provided, and the number of vertices matches that of the HCP 32k mesh (29696 on left, and 29716 on right), should the medial wall masks be added to the "xifti" if not provided? Default: TRUE.

read_dir  (Optional) Append a directory to all file names in the arguments. If NULL (default), do not modify file names.

validate  Validate that the result is a "xifti" object? Default: TRUE. If FALSE, the result may not be properly formatted if the inputs were invalid.

Value

The "xifti" object containing all the data in the input giftis.

See Also

Other functions for reading in CIFTI or GIFTI metric data: as.xifti(), info_cifti(), load_parc(), load_surf(), read_cifti(), read_surf()

remove_xifti  Remove a component from a "xifti"

Description

Remove a brain structure or surface from a "xifti"

Usage

remove_xifti(xifti, remove = NULL)

Arguments

xifti  A "xifti" object.
remove  A character vector containing one or more of the following: "cortex_left", "cortex_right", "subcortical", "surf_left", and "surf_right". Each of these components will be removed from the "xifti"
Value
The new "xifti" with the requested component(s) removed

See Also
Other functions for manipulating 'xifti' objects: add_surf(), apply_xifti(), combine_xifti(), convert_to_dlabel(), merge_xifti(), newdata_xifti(), select_xifti(), transform_xifti()

resample_cifti

Resample CIFTI data

Description
Performs spatial resampling of CIFTI data on the cortical surface by separating it into GIFTI and NIFTI files, resampling the GIFTIs, and then putting them together. (The subcortex is not resampled.)

Usage
resample_cifti(
  x = NULL,
  cifti_target_fname = NULL,
  surfL_original_fname = NULL,
  surfR_original_fname = NULL,
  surfL_target_fname = NULL,
  surfR_target_fname = NULL,
  resamp_res,
  write_dir = NULL,
  mwall_values = c(NA, NaN),
  verbose = TRUE
)

resampleCIfTI(
  x = NULL,
  cifti_target_fname = NULL,
  surfL_original_fname = NULL,
  surfR_original_fname = NULL,
  surfL_target_fname = NULL,
  surfR_target_fname = NULL,
  resamp_res,
  write_dir = NULL,
  mwall_values = c(NA, NaN),
  verbose = TRUE
)

resamplecii(
  x = NULL,
resample_cifti

cifti_target_fname = NULL,
surfL_original_fname = NULL,
surfR_original_fname = NULL,
surfL_target_fname = NULL,
surfR_target_fname = NULL,
resamp_res,
write_dir = NULL,
mwall_values = c(NA, NaN),
verbose = TRUE
}

resample_xifti(
  x = NULL,
cifti_target_fname = NULL,
surfL_original_fname = NULL,
surfR_original_fname = NULL,
surfL_target_fname = NULL,
surfR_target_fname = NULL,
resamp_res,
write_dir = NULL,
mwall_values = c(NA, NaN),
verbose = TRUE
)

Arguments

x
  The CIFTI file name or "xifti" object to resample. If NULL, the result will
  be a "xifti" with resampled surfaces given by surfL_original_fname and
  surfR_original_fname.
cifti_target_fname
  File name for the resampled CIFTI. Will be placed in write_dir. If NULL, will
  be written to "resampled.d*.nii". write_dir will be appended to the beginning
  of the path.
surfL_original_fname, surfR_original_fname
  (Optional) Path to a GIFTI surface geometry file representing the left/right cortex.
  One or both can be provided. These will be resampled too, and are convenient
  for visualizing the resampled data.
  If x is a "xifti" object with surfaces, these arguments will override the surfaces
  in the "xifti".
surfL_target_fname, surfR_target_fname
  (Optional) File names for the resampled GIFTI surface geometry files. Will
  be placed in write_dir. If NULL (default), will use default names created by
  resample_cifti_default_fname.
resamp_res
  Target resolution for resampling (number of cortical surface vertices per hemi-
  sphere).
write_dir
  Where to write the resampled CIFTI (and surfaces if present.) If NULL (default),
  will use the current working directory if x was a CIFTI file, and a temporary
directory if x was a "xifti" object.
Resample a CIFTI from a template

Description

Resample a CIFTI from a template, ensuring the new CIFTI’s resolution matches that of the template.

Usage

```r
resample_cifti_from_template(original_fname, template_fname, target_fname)
```

Arguments

- `original_fname`: A CIFTI file to resample.
- `template_fname`: A CIFTI file to use as the template.
- `target_fname`: The file name to save the resampled CIFTI.

Value

The `target_fname`, invisibly

details

Correctly indicating the medial wall locations is important for resampling, because the medial wall mask is taken into account during resampling calculations.

Value

A named character vector of written files: "cifti" and potentially "surfL" (if `surfL_original_fname` was provided) and/or "surfR" (if `surfR_original_fname` was provided).

Connectome Workbench

This function interfaces with the "-metric-resample", "-label-resample", and/or "-surface-resample" Workbench commands, depending on the input.

See Also

Other commonly-used functions: `is.cifti()`, `read_cifti()`, `smooth_cifti()`, `view_xifti_surface()`, `view_xifti_volume()`, `write_cifti()`
Connectome Workbench

This function interfaces with the "-cifti-resample" Workbench command.

--

**resample_gifti**  
Resample a GIFTI file (with its ROI)

---

### Description

Perform spatial resampling of GIFTI data on the cortical surface (metric and label), or of GIFTI surface geometry data itself.

### Usage

```r
resample_gifti(
  original_fname,
  target_fname,
  hemisphere = c("left", "right"),
  file_type = NULL,
  original_res = NULL,
  resamp_res,
  ROIcortex_original_fname = NULL,
  ROIcortex_target_fname = NULL,
  read_dir = NULL,
  write_dir = NULL
)
```

```r
resampleGIfTI(
  original_fname,
  target_fname,
  hemisphere,
  file_type = NULL,
  original_res = NULL,
  resamp_res,
  ROIcortex_original_fname = NULL,
  ROIcortex_target_fname = NULL,
  read_dir = NULL,
  write_dir = NULL
)
```

```r
resamplegii(
  original_fname,
  target_fname,
  hemisphere,
  file_type = NULL,
  original_res = NULL,
  resamp_res,
```
Arguments

- **original_fName** (required): The GIFTI file to resample.
- **target_fName** (optional): Where to save the resampled file.
- **hemisphere** (optional): "left" (default) or "right". An error will occur if the hemisphere indicated in the GIFTI metadata does not match.
- **file_type** (optional): "metric", "label", "surf", or NULL (default) to infer from original_fName.
- **original_res** (optional): The resolution of the original file. If NULL (default), infer from the file.
- **resamp_res** (required): Target resolution for resampling (number of cortical surface vertices per hemisphere).
- **ROIcortex_original_fName** (optional): The name of the ROI file corresponding to original_fName. Leave as NULL (default) if this doesn’t exist or shouldn’t be resampled.
- **ROIcortex_target_fName** (optional): The name of the resampled ROI file. Only applicable if ROIcortex_original_fName is provided.
- **read_dir** (optional): Directory to append to the path of every file name in original_fName and ROIcortex_original_fName. If NULL (default), do not append any directory to the path.
- **write_dir** (optional): Directory to append to the path of every file name in target_fName and ROIcortex_target_fName. If NULL (default), do not append any directory to the path.

Value

The resampled GIFTI file name, invisibly

Connectome Workbench

This function interfaces with the "-metric-resample", "-label-resample", and/or "-surface-resample" Workbench commands, depending on the input.

```
resample_surf
```

Resample a "surf" object

Description

Resample a "surf" by writing it to a GIFTI, using the Connectome Workbench to resample it, and then reading the new file.
Usage

rotate_surf(surf, r1 = 0, r2 = 0, r3 = 0, units = c("radians", "degrees"))

Arguments

surf: The "surf" object: see is.surf.

r1, r2, r3: Angle to rotate along the first, second, and third column's axis, in units (e.g. changing r1 will change the vertex positions in the second and third dimensions/columns, since the mesh is being rotated with respect to the first column's axis). Default: 0.

With view_xifti_surface and other mesh rendering functions that use rgl, these rotations seem to correspond to yaw, pitch, and roll, respectively.

units: "radians" (default) or "degrees".

Description

Rotate a "surf". Can be used to adjust the mesh orientation prior to view_xifti_surface.

Usage

rotate_surf(surf, r1 = 0, r2 = 0, r3 = 0, units = c("radians", "degrees"))
Value
The rotated "surf"

See Also
Other functions for working with GIFTI surface geometry data: `is.surf()`, `read_surf()`, `resample_surf()`, `view_surf()`, `write_surf_gifti()`

### Description
"ROY_BIG_BL", the default palette from the Connectome Workbench.

### Usage

```
ROY_BIG_BL(min = 0, max = 1, mid = NULL, half = NULL, pos_half = FALSE)
```

### Arguments

- **min**: The minimum value for the color mapping. As in the original palette, the last color (aqua) is actually placed at the bottom .5\ the minimum and maximum. Default: 0
- **max**: The maximum value for the color mapping. If this value is lower than the minimum, the color mapping will be reversed. If this is equal to the minimum, a palette with only the color black will be returned. Default: 1.
- **mid**: (Optional) The midpoint value for the color mapping. If NULL (default), the true midpoint is used.
- **half**: "positive" or "negative" to use the positive half (black -> red -> yellow) or negative half (black -> blue -> purple -> green -> aqua) only. NULL (default) or FALSE to use entire palette.
- **pos_half**: Deprecated. Use half.

### Details
Yields the landmark color hex codes and values for the "ROY_BIG_BL" palette. This is the same color palette as the default Connectome Workbench palette. Source: https://github.com/Washington-University/workbench/blob/master/src/Files/PaletteFile.cxx

### Value
A data.frame with two columns: "color" (character: color hex codes) and "value" (numeric)
run_wb_cmd  
Wrapper for Connectome Workbench Commands

Description

Runs a Connectome Workbench command that has already been formatted.

Usage

run_wb_cmd(cmd, intern = TRUE, ignore.stdout = NULL, ignore.stderr = NULL)

Arguments

cmd  The full command, beginning after the workbench path.
intern  Return printed output? If FALSE, return logical indicating success instead. Default: TRUE.
ignore.stdout, ignore.stderr  The "ignore.stdout" and "ignore.stderr" arguments to system. Should be logical or NULL. If NULL (default), messages will be controlled by ciftiToolsgetOption("suppress_msgs") and errors will not be ignored.

Value

If intern==TRUE, the printed output of the command. If intern==FALSE, a logical indicating if the command finished successfully.

select_xifti  
Select columns of a "xifti"

Description

Select column indices to keep in a "xifti". Can also be used to reorder the columns.

Usage

select_xifti(xifti, idx, add_meta = "select")

Arguments

xifti  A "xifti" object.
idx  The column indices to keep, in order.
add_meta  Add idx to xifti$meta$cifti$misc[[add_meta]] for reference. Default: "select". If NULL or an empty string, do not add a metadata entry.
separate_cifti

Value

The "xifti" with only the selected columns.

See Also

Other functions for manipulating 'xifti' objects: add_surf(), apply_xifti(), combine_xifti(), convert_to_dlabel(), merge_xifti(), newdata_xifti(), remove_xifti(), transform_xifti()

Description

Separate a CIFTI file into GIFTI files for the cortical data and NIFTI files for the subcortical data and labels. ROIs can also be written to indicate the medial wall mask (cortex) and volume mask (subcortex). This uses the Connectome Workbench command -cifti-separate.

Usage

separate_cifti(
    cifti_fname,
    brainstructures = c("left", "right"),
    cortexL_fname = NULL,
    cortexR_fname = NULL,
    subcortVol_fname = NULL,
    subcortLabs_fname = NULL,
    ROI_brainstructures = "all",
    ROIcortexL_fname = NULL,
    ROIcortexR_fname = NULL,
    ROIsubcortVol_fname = NULL,
    write_dir = NULL
)

separateCIfTI(
    cifti_fname,
    brainstructures = c("left", "right"),
    cortexL_fname = NULL,
    cortexR_fname = NULL,
    subcortVol_fname = NULL,
    subcortLabs_fname = NULL,
    ROI_brainstructures = "all",
    ROIcortexL_fname = NULL,
    ROIcortexR_fname = NULL,
    ROIsubcortVol_fname = NULL,
    write_dir = NULL
)
separate_cifti(
  cifti_fname,
  brainstructures = c("left", "right"),
  cortexL_fname = NULL,
  cortexR_fname = NULL,
  subcortVol_fname = NULL,
  subcortLabs_fname = NULL,
  ROI_brainstructures = "all",
  ROIcortexL_fname = NULL,
  ROIcortexR_fname = NULL,
  ROIsubcortVol_fname = NULL,
  write_dir = NULL
)

Arguments

cifti_fname  File path to a CIFTI file (ending in ".d*.nii").

brainstructures
Character vector indicating which brain structure(s) to obtain: "left" (left cortex), "right" (right cortex) and/or "subcortical" (subcortex and cerebellum). Can also be "all" (obtain all three brain structures). Default: c("left", "right") (cortex only).

If a brain structure is indicated but does not exist in the CIFTI file, a warning will occur and that brain structure will be skipped.

cortexL_fname, cortexR_fname
(Optional) GIFTI file names (*.func/label.gii) to save the [left/right] cortex data to. If not provided, defaults to ".[L/R].\[func/label\].gii", where * is the file name component of cifti_fname. Will be written in write_dir. dtseries and dscalar files should use "func", whereas dlabel files should use "label".

subcortVol_fname, subcortLabs_fname
(Optional) NIFTI file names to save the subcortical [volume/labels] to. If not provided, defaults to ".labels\.nii", where * is the file name component of cifti_fname. Will be written in write_dir.

ROI_brainstructures
Which ROIs should be obtained? "all" (default) to obtain ROIs for each of the brainstructures. NULL to not obtain any ROIs. This should be a subset of brainstructures.

ROIcortexL_fname, ROIcortexR_fname
(Optional) GIFTI file names (*.func/label.gii) to save the [left/right] cortex ROI to. If not provided, defaults to "*.ROI_[L/R].\[func/label\].gii", where * is the file name component of cifti_fname. The cortical ROIs typically represent the medial wall mask, with values of 1 for in-ROI (non-medial wall) vertices and 0 for out-of-ROI (medial wall) vertices. Will be written in write_dir. dtseries and dscalar files should use "func", whereas dlabel files should use "label".
**smooth_cifti**

**ROIsubcortVol_fname**

(Optional) NII file names to save the subcortical ROI to. If not provided, defaults to "*ROI.nii", where * is the file name component of cifti_fname. The subcortical ROI typically represents the volumetric mask for the entire subcortical structure, with values of 1 for in-ROI (in subcortex) voxels and 0 for out-of-ROI (not in subcortex) voxels. Will be written in write_dir.

**write_dir**

Where should the separated files be placed? NULL (default) will write them to the current working directory.

write_dir must already exist, or an error will occur.

**Details**

Time unit, start, and step (dtseries files) will not be written to the GIFTI/NIFTIs. Column names (dscalar files) will not be written to the GIFTIs, as well as label names and colors (dlabel files). (Haven’t checked the NIFTIs yet.)

ROI/medial wall behavior: If there are 32k vertices in the left cortex with 3k representing the medial wall, then both cortexL_fname and ROIcortexL_fname will have 32k entries, 3k of which having a value of 0 indicating the medial wall. The non-medial wall entries will have the data values in cortexL fname and a value of 1 in ROIcortexL_fname. Thus, exporting ROIcortexL_fname is vital if the data values include 0, because 0-valued non-medial wall vertices and medial wall vertices cannot be distinguished from one another within cortexL_fname alone.

**Value**

A named character vector with the file paths to the written NIFTI and GIFTI files

**Connectome Workbench**

This function interfaces with the "-cifti-separate" Workbench command.

**See Also**

Other functions for writing CIFTI or GIFTI metric data: write_cifti(), write_metric_gifti(), write_subcort_nifti(), write_surf_gifti()
smooth_cifti

Usage

smooth_cifti(
  x,
  cifti_target_fname = NULL,
  surf_FWHM = 5,
  vol_FWHM = 5,
  surfL_fname = NULL,
  surfR_fname = NULL,
  cerebellum_fname = NULL,
  subcortical_zeroes_as_NA = FALSE,
  cortical_zeroes_as_NA = FALSE,
  subcortical_merged = FALSE
)

smoothCIfTI(
  x,
  cifti_target_fname = NULL,
  surf_FWHM = 5,
  vol_FWHM = 5,
  surfL_fname = NULL,
  surfR_fname = NULL,
  cerebellum_fname = NULL,
  subcortical_zeroes_as_NA = FALSE,
  cortical_zeroes_as_NA = FALSE,
  subcortical_merged = FALSE
)

smoothcii(
  x,
  cifti_target_fname = NULL,
  surf_FWHM = 5,
  vol_FWHM = 5,
  surfL_fname = NULL,
  surfR_fname = NULL,
  cerebellum_fname = NULL,
  subcortical_zeroes_as_NA = FALSE,
  cortical_zeroes_as_NA = FALSE,
  subcortical_merged = FALSE
)

smooth_xifti(
  x,
  cifti_target_fname = NULL,
  surf_FWHM = 5,
  vol_FWHM = 5,
  surfL_fname = NULL,
  surfR_fname = NULL,
  cerebellum_fname = NULL,
  subcortical_zeroes_as_NA = FALSE,
  cortical_zeroes_as_NA = FALSE,
  subcortical_merged = FALSE
)
subcortical_zeroes_as_NA = FALSE,  
cortical_zeroes_as_NA = FALSE,  
subcortical_merged = FALSE 
)

Arguments

x  The CIFTI file name or "xifti" object to smooth.
cifti_target_fname  
File name for the smoothed CIFTI. If NULL, will be written to "smoothed.d*.nii"  
in the current working directory if x was a CIFTI file, and in a temporary directory if x was a "xifti" object.
surf_FWHM, vol_FWHM  
The full width at half maximum (FWHM) parameter for the gaussian surface or volume smoothing kernel, in mm. Default: 5
surfL_fname, surfR_fname  
(Required if the corresponding cortex is present) Surface GIFTI files for the left and right cortical surfaces. If not provided, the default surfaces will be used.
cerebellum_fname  
(Optional) Surface GIFTI file for the cerebellar surface
subcortical_zeroes_as_NA, cortical_zeroes_as_NA  
Should zero-values in the subcortical volume or cortex be treated as NA? Default: FALSE.
subcortical_merged  
Smooth across subcortical structure boundaries? Default: FALSE.

Details

If the CIFTI is a "dlabel" file (intent 3007), then it will be converted to a "dscalar" file because the values will no longer be integer indices. Unless the label values were ordinal, this is probably not desired so a warning will be printed.
Can accept a "xifti" object as well as a path to a CIFTI-file.
Surfaces are required for each hemisphere in the CIFTI. If they are not provided, the default inflated surfaces will be used.

Value

The cifti_target_fname, invisibly

Connectome Workbench

This function interfaces with the "-cifti-smoothing" Workbench command.

See Also

Other commonly-used functions: is.cifti(), read.cifti(), resample.cifti(), view.xifti_surface(), view.xifti_volume(), write.cifti()
smooth_gifti  

Smooth a metric GIFTI file

Description

Smooths metric GIFTI data along the cortical surface. The results are written to a new GIFTI file.

Usage

smooth_gifti(
  original_fname,
  target_fname,
  surf_fname = NULL,
  surf_FWHM = 5,
  hemisphere = c("left", "right"),
  ROI_fname = NULL,
  zeroes_as_NA = FALSE
)

smoothGIfTI(
  original_fname,
  target_fname,
  surf_fname,
  surf_FWHM = 5,
  zeroes_as_NA = FALSE
)

smoothgii(
  original_fname,
  target_fname,
  surf_fname,
  surf_FWHM = 5,
  zeroes_as_NA = FALSE
)

Arguments

original_fname  The GIFTI file to smooth.
target_fname    Where to save the smoothed file.
surf_fname      Surface GIFTI files cortical surface along which to smooth. If not provided, the default inflated surfaces will be used.
surf_FWHM       The full width at half maximum (FWHM) parameter for the gaussian surface smoothing kernel, in mm. Default: 5
hemisphere      The cortex hemisphere: "left" or "right". Only used if surf_fname is NULL.
substructure_table

ROI_fname  The ROI to limit smoothing to, as a metric file. This is used to exclude the medial wall from smoothing. If not provided (default) all the data is smoothed across the surface.

zeroes_as_NA  Should zero-values be treated as NA? Default: FALSE.

Value

The smoothed GIFTI file name, invisibly

Connectome Workbench

This function interfaces with the "-metric-smoothing" Workbench command.

---

description  Substructure table

Description

Table of labels for cortex hemispheres (left and right) and subcortical substructures. The names used by the CIFTI format and the names used by ciftiTools are given.

Usage

substructure_table()

Details

The names used by ciftiTools are based on those in FT_READ_CIFTI from the FieldTrip MATLAB toolbox.

Value

A data.frame with each substructure along the rows. The first column gives the CIFTI format name and the second column gives the ciftiTools name.
summary.surf

**Summarize a "surf" object**

**Description**

Summary method for class "surf"

**Usage**

```r
## S3 method for class 'surf'
summary(object, ...)

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

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

**Arguments**

- `object` Object of class "surf". See `is.surf` and `make_surf`.
- `...` further arguments passed to or from other methods.
- `x` Object of class "surf".

summary.xifti

**Summarize a "xifti" object**

**Description**

Summary method for class "xifti"

**Usage**

```r
## S3 method for class 'xifti'
summary(object, ...)

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

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

Arguments

object    Object of class "xifti".
...       further arguments passed to or from other methods.
x         A "xifti" object.

The NIFTI intents supported by ciftiTools

Description

Table of CIFTI file types (NIFTI intents) supported by ciftiTools.

Usage

supported_intents()

Details


Value

A data.frame with each supported file type along the rows, and column names "extension", "intent_code", "value", and "intent_name"

sys_path

Format a path for system

Description

Right now, it escapes spaces and parentheses with "\\".

Usage

sys_path(R_path)

Arguments

R_path    The name of the file. It should be properly formatted: if it exists, file.exists(R_path) should be TRUE.

Value

The name of the file
transform_xifti

Apply a univariate transformation to a "xifti" or pair of "xifti"s.

Description

Apply a univariate transformation to each value in a "xifti" or pair of "xifti"s. If a pair, they must share the same dimensions (brainstructures) and number of measurements.

Usage

transform_xifti(xifti, FUN, xifti2 = NULL, ...)

## S3 method for class 'xifti'
    xifti + xifti2

## S3 method for class 'xifti'
    xifti - xifti2

## S3 method for class 'xifti'
    xifti * xifti2

## S3 method for class 'xifti'
    xifti ^ xifti2

## S3 method for class 'xifti'
    xifti %% xifti2

## S3 method for class 'xifti'
    xifti %/% xifti2

## S3 method for class 'xifti'
    xifti / xifti2

## S3 method for class 'xifti'
    abs(x)

## S3 method for class 'xifti'
    sign(x)

## S3 method for class 'xifti'
    sqrt(x)

## S3 method for class 'xifti'
    floor(x)

## S3 method for class 'xifti'
    ceiling(x)
## S3 method for class 'xifti'
round(x, digits = 0)

## S3 method for class 'xifti'
exp(x)

## S3 method for class 'xifti'
log(x, base = exp(1))

### Arguments

- **xifti**: The xifti
- **FUN**: The function. If xifti2 is not provided, it should be a univariate function like `log` or `sqrt`. If xifti2 is provided, it should take in two arguments, like `+` or `pmax`.
- **xifti2**: The second xifti, if applicable. Otherwise, NULL (default)
- **...**: Additional arguments to FUN
- **x**: The "xifti"
- **digits**: The number of digits to round by
- **base**: The log base

### Details

If the "xifti" had the dlabel intent, and the transformation creates any value that is not a label value (e.g. a non-integer), then it is converted to a dscalar.

### Value

A "xifti" storing the result of applying FUN to the input(s). The data dimensions will be the same. The metadata of xifti will be retained, and the metadata of xifti2 will be discarded (if provided).

### See Also

Other functions for manipulating 'xifti' objects: `add_surf()`, `apply_xifti()`, `combine_xifti()`, `convert_to_dlabel()`, `merge_xifti()`, `newdata_xifti()`, `remove_xifti()`, `select_xifti()`

---

### unmask_cortex

#### Description

Get cortex data with medial wall vertices

#### Usage

unmask_cortex(cortex, mwall, mwall_fill = NA)
Arguments

cortex  V vertices x T measurements matrix
mwall  Logical vector with T TRUE values.
mwall_fill  The fill value to use for medial wall vertices.

Value

The unmasked cortex data

unmask_subcortex  Undo a volumetric mask

Description

Un-applies a mask to vectorized data to yield its volumetric representation. The mask and data should have compatible dimensions: the number of rows in dat should equal the number of locations within the mask. This is used for the subcortical CIFTI data.

Usage

unmask_subcortex(dat, mask, fill = NA)

Arguments

dat  Data matrix with locations along the rows and measurements along the columns. If only one set of measurements were made, this may be a vector.
mask  Volumetric binary mask. TRUE indicates voxels inside the mask.
fill  The value for locations outside the mask. Default: NA.

Value

The 3D or 4D unflattened volume array
**use_color_pal**

Use a color palette

Description

Applies a palette to a data vector to yield a vector of colors.

Usage

```r
use_color_pal(data_values, pal, color_NA = "white", indices = FALSE)
```

Arguments

- `data_values`: The values to map to colors
- `pal`: The palette to use to map values to colors
- `color_NA`: The color to use for `NA` values. Default: "white".
- `indices`: Return the numeric indices of colors in `pal$value` rather than the colors themselves. A value of 0 will be used for missing data. Default: `FALSE`.

Value

A character vector of color names (or integers if `indices`).

**view_surf**

View "surf" object(s)

Description

Visualize one or two "surf" objects(s), or the "surf" component(s) in a "xifti" using an interactive Open GL window made with `rgl`. The `rgl` package is required.

Usage

```r
view_surf(
  ..., 
  view = c("both", "lateral", "medial"),
  widget = NULL,
  title = NULL,
  fname = FALSE,
  cex.title = NULL,
  text_color = "black",
  bg = NULL,
  alpha = 1,
  edge_color = NULL,
  vertex_color = NULL,
)```
vertex_size = 0,
width = NULL,
height = NULL,
zoom = NULL
)

Arguments

... One of: A "surf" object; two "surf" objects; or, a "xifti" object. If a "surf" object has an empty "hemisphere" metadata entry, it will be set to the opposite side of the other’s if known; otherwise, it will be set to the left side. If both are unknown, the first will be taken as the left and the second as the right.

view Which view to display: "lateral", "medial", or "both". If NULL (default), both views will be shown. Each view will be plotted in a separate panel row.

widget Display the plot in an htmlwidget? Should be logical or NULL (default), in which case a widget will be used only if needed (length(idx)>1 & isFALSE(fname), fname is a file path to an .html file, or if rgl.useNULL()).

title Optional title(s) for the plot(s). It will be printed at the top in a separate subplot with 1/4 the height of the brain cortex subplots.

Default: NULL will not use any title if length(idx)==1. Otherwise, it will use the time index (".dtseries") or name (.dscalar or .dlabel) of each data column.

To use a custom title(s), use a length 1 character vector (same title for each plot) or length length(idx) character vector (different title for each plot). Set to NULL or an empty character to omit the title.

If the title is non-empty but does not appear, try lowering cex.title.

fname Save the plot(s) (and color legend if applicable)?

If isFALSE(fname) (default), no files will be written.

If fname is a length-1 character vector ending in ".html", an html with an interactive widget will be written.

If neither of the cases above apply, a png image will be written for each idx. If isTRUE(fname) the files will be named by the data column names (underscores will replace spaces). Set fname to a length 1 character vector to name files by this suffix followed by the fname_suffix: either the data column names ("names") or the index value ("idx"). Set this to a character vector with the same length as idx to name the files exactly.

cex.title Font size multiplier for the title. NULL (default) will use 2 for titles less than 20 characters long, and smaller sizes for increasingly longer titles.

text_color Color for text in title and colorbar legend. Default: "black".

bg Background color. NULL will not color the background (white).

alpha Transparency value for mesh coloring, between 0 and 1. Default: 1.0 (no transparency).

edge_color Outline each edge in this color. Default: NULL (do not outline the edges).

vertex_color Draw each vertex in this color. Default: "black". Vertices are only drawn if vertex_size > 0

vertex_size Draw each vertex with this size. Default: 0 (do not draw the vertices).
width

The dimensions of the RGL window, in pixels. If both are NULL (default), these dimensions depend on type of output (Open GL window or widget) and subplots (hemisphere, view, title, and slider_title) and are chosen to be the largest plot within a 1500 x 700 area (Open GL window) or 600 x 700 area (widget) that maintains a brain hemisphere subplot dimensions ratio of 10 x 7. Specifying only one will set the other to maintain this aspect ratio. Both can be specified to set the dimensions exactly. (These arguments do not affect the size of the legend, which cannot be controlled.)

height

The dimensions of the RGL window, in pixels. If both are NULL (default), these dimensions depend on type of output (Open GL window or widget) and subplots (hemisphere, view, title, and slider_title) and are chosen to be the largest plot within a 1500 x 700 area (Open GL window) or 600 x 700 area (widget) that maintains a brain hemisphere subplot dimensions ratio of 10 x 7. Specifying only one will set the other to maintain this aspect ratio. Both can be specified to set the dimensions exactly. (These arguments do not affect the size of the legend, which cannot be controlled.)

zoom

Adjust the sizes of the brain meshes. Default: NULL (will be set to 0.6 or 160\ for widget.)

Details

This function works as a wrapper to view_xifti_surface, but some arguments are not applicable (e.g. color scheme and legend). Also, instead of using the hemisphere argument, name the surface arguments surfL or surfR (see description for parameter...). Finally, the default value for param is "surf", not "xifti".

Navigating and Embedding the Interactive Plots

To navigate the interactive Open GL window and html widget, left click and drag the cursor to rotate the meshes. Use the scroll wheel or right click and drag to zoom. Press the scroll wheel and drag to change the field-of-view. For Open GL windows, execute snapshot to save the current window as a .png file, rgl.close to close the window, and rgl.viewpoint to programmatically control the perspective.

To embed an interactive plot in an R Markdown document, first execute rgl::setupKnitr() to prepare the document for embedding the widget. Then execute the plot command as you normally would to create a widget (i.e. without specifying fname, and by requesting more than one idx or by setting widget to TRUE). When the R Markdown document is knitted, the interactive widget should be displayed below the chunk in which the plot command was executed. See the vignette for an example.

Embedding the Static Plots

To embed a static plot in an R Markdown document, first execute rgl::setupKnitr() to prepare the document for embedding the snapshot of the Open GL window. Then execute the plot command as you normally would to create an Open GL window (i.e. without specifying fname, and by requesting only one idx). In the options for the chunk in which the plot command is executed, set rgl=TRUE,format="png". You can also control the image dimensions here e.g. fig.height=3.8, fig.width=5. When the R Markdown document is knitted, the static plots
should be displayed below the chunk in which the plot command was executed. See the vignette for an example.

See Also

Other functions for working with GIFTI surface geometry data: is.surf(), read_surf(), resample_surf(), rotate_surf(), write_surf_gifti()

---

view_xifti    View a "xifti" object

Description

Switch for view_xifti_surface or view_xifti_volume

Usage

view_xifti(xifti, what = NULL, ...)
view_cifti(xifti, what = NULL, ...)
viewCIfTI(xifti, what = NULL, ...)
viewcii(xifti, what = NULL, ...)

Arguments

xifti A "xifti" object.
what Either "surface" or "volume". NULL will infer based on the contents of the "xifti": if there is data, plot the surface cortex data if present, and the volumetric subcortical data otherwise. If there is no data, plot the surface geometry if present, and do nothing otherwise.
... Additional arguments to pass to either view function.

Value

The return value of view_xifti_surface or view_xifti_volume.
view_xifti_surface

View cortical surface data in a "xifti"

Description

Visualize "xifti" cortical data using an interactive Open GL window or htmlwidget made with rgl. The rmarkdown package is required for the htmlwidget functionality.

Usage

```r
view_xifti_surface(
  xifti = NULL,
  surfL = NULL,
  surfR = NULL,
  color_mode = "auto",
  zlim = NULL,
  colors = NULL,
  idx = NULL,
  hemisphere = NULL,
  view = c("both", "lateral", "medial"),
  widget = NULL,
  title = NULL,
  slider_title = "Index",
  fname = FALSE,
  fname_suffix = c("names", "idx"),
  legend_fname = "[fname]_legend",
  legend_ncol = NULL,
  legend_alllevels = FALSE,
  legend_embed = NULL,
  digits = NULL,
  cex.title = NULL,
  text_color = "black",
  bg = NULL,
  borders = FALSE,
  alpha = 1,
  edge_color = NULL,
  vertex_color = NULL,
  vertex_size = 0,
  width = NULL,
  height = NULL,
  zoom = NULL
)
```

```r
view_cifti_surface(
  xifti = NULL,
  surfL = NULL,
  surfR = NULL,  
```
color_mode = "auto",
zlim = NULL,
colors = NULL,
idx = NULL,
hemisphere = NULL,
view = c("both", "lateral", "medial"),
widget = NULL,
title = NULL,
slider_title = "Index",
fname = FALSE,
fname_suffix = c("names", "idx"),
legend_ncol = NULL,
legend_embed = NULL,
digits = NULL,
cex.title = NULL,
text_color = "black",
bg = NULL,
borders = FALSE,
alpha = 1,
edge_color = NULL,
vertex_color = NULL,
vertex_size = 0,
width = NULL,
height = NULL,
zoom = NULL
)

viewCIIfTI_surface(
  xifti = NULL,
surfL = NULL,
surfR = NULL,
color_mode = "auto",
zlim = NULL,
colors = NULL,
idx = NULL,
hemisphere = NULL,
view = c("both", "lateral", "medial"),
widget = NULL,
title = NULL,
slider_title = "Index",
fname = FALSE,
fname_suffix = c("names", "idx"),
legend_ncol = NULL,
legend_embed = NULL,
digits = NULL,
cex.title = NULL,
text_color = "black",
bg = NULL,
Arguments

`view_xifti_surface`

- **xifti**: A "xifti" object.
- **surfL, surfR**: (Optional) The brain surface model to use. Each can be a "surf" object, any valid argument to `read_surf`, or one of "very inflated", "inflated", or "midthickness". If provided, it will override `xifti$surf$cortex_left` or `xifti$surf$cortex_right` if it exists. Leave as NULL (default) to use `xifti$surf$cortex_left`
view_xifti_surface

or xifti$surf$cortex_right if it exists, or the default inflated surfaces if it
does not exist.

color_mode
(Optional) "sequential", "qualitative", "diverging", or "auto" (default).
Auto mode will use the qualitative color mode if the "xifti" object represents
a .dlabel CIFTI (intent 3007). Otherwise, it will use the diverging mode if the
data contains both positive and negative values, and the sequential mode if the
data contains >90\%make_color_pal for more details.

zlim
(Optional) Controls the mapping of values to each color in colors. If the length
is longer than one, using -Inf will set the value to the data minimum, and Inf will
set the value to the data maximum. See make_color_pal description for more
details.

colors
(Optional) "ROY_BIG_BL", vector of colors to use, the name of a ColorBrewer
palette (see RColorBrewer::brewer_pal.info and colorbrewer2.org), or the
name of a viridisLite palette. Defaults are "ROY_BIG_BL" (sequential), "Set2"
(qualitative), and "ROY_BIG_BL" (diverging). An exception to these defaults is if
the "xifti" object represents a .dlabel CIFTI (intent 3007), then the qualitative
colors in the label table will be used. See make_color_pal for more details.

idx
The time/column index of the data to display.
If its length is greater than one, and isFALSE(fname), a widget must be used
since a single OpenGL window cannot show multiple indexes. A slider will be
added to the widget to control what time/column is being displayed.

hemisphere
Which brain cortex to display: "both" (default), "left", or "right". Each will be
plotted in a separate panel column.
If a brain cortex is requested but no surface is available, a default inflated surface
will be used.
This argument can also be NULL (default). In this case, the default inflated sur-
faced included with ciftiTools will be used for each cortex with data (i.e. if
xifti$data$cortex_left and/or xifti$data$cortex_right exist).
Surfaces without data will be colored white.

view
Which view to display: "lateral", "medial", or "both". If NULL (default),
both views will be shown. Each view will be plotted in a separate panel row.

widget
Display the plot in an htmlwidget? Should be logical or NULL (default), in which
case a widget will be used only if needed (length(idx)>1 & isFALSE(fname),
fname is a file path to an .html file, or if rgl.useNULL()).

title
Optional title(s) for the plot(s). It will be printed at the top in a separate subplot
with 1/4 the height of the brain cortex subplots.
Default: NULL will not use any title if length(idx)==1. Otherwise, it will use
the time index (".dtseries") or name (.dscalar or .dlabel) of each data column.
To use a custom title(s), use a length 1 character vector (same title for each plot)
or length length(idx) character vector (different title for each plot). Set to
NULL or an empty character to omit the title.
If the title is non-empty but does not appear, try lowering cex.title.

slider_title
Text at bottom of plot that will be added if a slider is used, to provide a title for
it. Default: "Index". If NULL or an empty character, no title will be added.
fname

Save the plot(s) (and color legend if applicable)?
If isFALSE(fname) (default), no files will be written.
If fname is a length-1 character vector ending in ".html", an html with an interactive widget will be written.
If neither of the cases above apply, a png image will be written for each idx. If isTRUE(fname) the files will be named by the data column names (underscores will replace spaces). Set fname to a length 1 character vector to name files by this suffix followed by the fname_suffix: either the data column names ("names") or the index value ("idx"). Set this to a character vector with the same length as idx to name the files exactly.

fname_suffix

Save the plot(s) (and color legend if applicable)?
If isFALSE(fname) (default), no files will be written.
If fname is a length-1 character vector ending in ".html", an html with an interactive widget will be written.
If neither of the cases above apply, a png image will be written for each idx. If isTRUE(fname) the files will be named by the data column names (underscores will replace spaces). Set fname to a length 1 character vector to name files by this suffix followed by the fname_suffix: either the data column names ("names") or the index value ("idx"). Set this to a character vector with the same length as idx to name the files exactly.

legend_fname

Save the color legend? Since the legend is the same for each idx only one legend is written even if length(idx)>1. This argument can be NULL to not save the legend, an exact file path, or a length-one character vector with "[fname]\[1\]" in it, which will name the legend based on fname\[1\]. For example, if fname\[1\] is "plots/my_cifti.png" and legend_fname is "\[fname\]_legend" (default), then the legend plot will be saved to "plots/my_cifti_legend.png".

legend_ncol

Number of columns in color legend. If NULL (default), use 10 entries per row. Only applies if the color legend is used (qualitative data).

legend_alllevels

Show all label levels in the color legend? If FALSE (default), just show the levels present in the data being viewed. Only applies if the color legend is used (qualitative data).

legend_embed

Should the colorbar be embedded in the plot? It will be positioned in the bottom-left corner, in a separate subplot with 1/4 the height of the brain cortex subplots. Default: TRUE. If FALSE, print/save it separately instead. Only applies if the color bar is used (sequential or diverging data). The color legend (qualitative data) cannot be embedded at the moment.

digits

The number of digits for the colorbar legend ticks. If NULL (default), let format decide.
cex.title

Font size multiplier for the title. NULL (default) will use 2 for titles less than 20 characters long, and smaller sizes for increasingly longer titles.
text_color

Color for text in title and colorbar legend. Default: "black".
bg

Background color. NULL will not color the background (white).
borders

Only applicable if color_mode is "qualitative". Border vertices will be identified (those that share a face with at least one vertex of a different value) and
colored over. If this argument is TRUE borders will be colored in black; provide the name of a different color to use that instead. If FALSE or NULL (default), do not draw borders.

**alpha**
Transparency value for mesh coloring, between 0 and 1. Default: 1.0 (no transparency).

**edge_color**
Outline each edge in this color. Default: NULL (do not outline the edges).

**vertex_color**
Draw each vertex in this color. Default: "black". Vertices are only drawn if vertex_size > 0

**vertex_size**
Draw each vertex with this size. Default: 0 (do not draw the vertices).

**width**
The dimensions of the RGL window, in pixels. If both are NULL (default), these dimensions depend on type of output (Open GL window or widget) and subplots (hemisphere, view, title, and slider_title) and are chosen to be the largest plot within a 1500 x 700 area (Open GL window) or 600 x 700 area (widget) that maintains a brain hemisphere subplot dimensions ratio of 10 x 7. Specifying only one will set the other to maintain this aspect ratio. Both can be specified to set the dimensions exactly. (These arguments do not affect the size of the legend, which cannot be controlled.)

**height**
The dimensions of the RGL window, in pixels. If both are NULL (default), these dimensions depend on type of output (Open GL window or widget) and subplots (hemisphere, view, title, and slider_title) and are chosen to be the largest plot within a 1500 x 700 area (Open GL window) or 600 x 700 area (widget) that maintains a brain hemisphere subplot dimensions ratio of 10 x 7. Specifying only one will set the other to maintain this aspect ratio. Both can be specified to set the dimensions exactly. (These arguments do not affect the size of the legend, which cannot be controlled.)

**zoom**
Adjust the sizes of the brain meshes. Default: NULL (will be set to 0.6 or 160 for widget.)

### Value

If a png or html file(s) were written, the names of the files for each index will be returned. Otherwise, the widget itself is returned if a widget was used, and the rgl object IDs are returned if an Open GL window was used. The rgl object IDs are useful for further programmatic manipulation of the Open GL window.

### Navigating and Embedding the Interactive Plots

To navigate the interactive Open GL window and html widget, left click and drag the cursor to rotate the meshes. Use the scroll wheel or right click and drag to zoom. Press the scroll wheel and drag to change the field-of-view. For Open GL windows, execute `snapshot` to save the current window as a .png file, `rgl.close` to close the window, and `rgl.viewpoint` to programmatically control the perspective.

To embed an interactive plot in an R Markdown document, first execute `rgl::setupKnitr()` to prepare the document for embedding the widget. Then execute the plot command as you normally would to create a widget (i.e. without specifying `fname`, and by requesting more than one `idx` or by setting `widget` to `TRUE`). When the R Markdown document is knitted, the interactive widget should be displayed below the chunk in which the plot command was executed. See the vignette for an example.
Embedding the Static Plots

To embed a static plot in an R Markdown document, first execute `rgl::setupKnitr()` to prepare the document for embedding the snapshot of the Open GL window. Then execute the plot command as you normally would to create an Open GL window (i.e. without specifying `fname`, and by requesting only one `idx`). In the options for the chunk in which the plot command is executed, set `rgl=TRUE, format="png"`. You can also control the image dimensions here e.g. `fig.height=3.8, fig.width=5`. When the R Markdown document is knitted, the static plots should be displayed below the chunk in which the plot command was executed. See the vignette for an example.

See Also

Other commonly-used functions: `is.cifti()`, `read_cifti()`, `resample_cifti()`, `smooth_cifti()`, `view_xifti_volume()`, `write_cifti()`

---

**Description**

View subcortical data in a "xifti" using slices.

**Usage**

```r
view_xifti_volume(
  xifti,
  structural_img = "MNI",
  idx = 1,
  plane = c("axial", "sagittal", "coronal"),
  num.slices = 9,
  interactive = FALSE,
  zlim = NULL,
  verbose = TRUE,
  ...
)
```

```r
view_cifti_volume(
  xifti,
  structural_img = "MNI",
  idx = 1,
  plane = c("axial", "sagittal", "coronal"),
  num.slices = 9,
  interactive = FALSE,
  zlim = NULL,
  verbose = TRUE,
  ...
)
```
viewCIfTI_volume(
  xifti,
  structural_img = "MNI",
  idx = 1,
  plane = c("axial", "sagittal", "coronal"),
  num.slices = 9,
  interactive = FALSE,
  zlim = NULL,
  verbose = TRUE,
  ...
)

viewcii_volume(
  xifti,
  structural_img = "MNI",
  idx = 1,
  plane = c("axial", "sagittal", "coronal"),
  num.slices = 9,
  interactive = FALSE,
  zlim = NULL,
  verbose = TRUE,
  ...
)

Arguments

xifti A "xifti" object.

structural_img The structural MRI image on which to overlay the subcortical values. Can be a file name, "MNI" (default) to use the MNI T1-weighted template, or NULL to use a blank image.

idx The time/column index of the "xifti" data to plot. Must be a single index (length 1).

plane If interactive=FALSE, the plane to display. Default: "axial". Other options are "sagittal" and "coronal".

num.slices If interactive=FALSE, the number of slices to display. Default: 9.

interactive interactive=TRUE will use papayar to allows for interactive visualization.

zlim A length-2 numeric vector giving the minimum and maximum values to plot. Data values beyond this range will be truncated to the min/max. If NULL (default), will use the min and max of the data.

verbose Should occasional updates be printed? Default: TRUE.

... Additional arguments to pass to papayar::papaya or oro.nifti::overlay

See Also

Other commonly-used functions: is.cifti(), read_cifti(), resample_cifti(), smooth_cifti(), view_xifti_surface(), write_cifti()
**write_cifti**

Write a CIFTI file from a "xifti" object

---

**Description**

Write out a "xifti" object as a CIFTI file and (optionally) GIFTI surface files.

**Usage**

```r
write_cifti(
  xifti,
  cifti_fname,
  surfL_fname = NULL,
  surfR_fname = NULL,
  verbose = TRUE
)
```

```r
writeCIfTI(
  xifti,
  cifti_fname,
  surfL_fname = NULL,
  surfR_fname = NULL,
  verbose = TRUE
)
```

```r
writecii(
  xifti,
  cifti_fname,
  surfL_fname = NULL,
  surfR_fname = NULL,
  verbose = TRUE
)
```

```r
write_xifti(
  xifti,
  cifti_fname,
  surfL_fname = NULL,
  surfR_fname = NULL,
  verbose = TRUE
)
```

**Arguments**

- **xifti**: A "xifti" object.
- **cifti_fname**: File path to a CIFTI file (ending in ".d*.nii").
write_metric_gifti

```
surfL_fname, surfR_fname
    If the [left/right] surface is present, it will be written to a GIFTI file at this file path. If NULL (default), do not write out the surface.
verbose
    Should occasional updates be printed? Default: TRUE.
```

Value

Named character vector of the written files

Connectome Workbench

This function interfaces with the "-cifti-create-dense-timeseries","-cifti-create-dense-scalar", or "-cifti-create-label" Workbench Command, depending on the input.

See Also

Other commonly-used functions: `is.cifti()`, `read_cifti()`, `resample_cifti()`, `smooth_cifti()`, `view_xifti_surface()`, `view_xifti_volume()`

Other functions for writing CIFTI or GIFTI metric data: `separate_cifti()`, `write_metric_gifti()`, `write_subcort_nifti()`, `write_surf_gifti()`

---

**write_metric_gifti**  
Write a data matrix to a GIFTI metric file

---

**Description**

Write the data for the left or right cortex to a metric GIFTI file.

**Usage**

```
write_metric_gifti(
    x,  
    gifti_fname,  
    hemisphere = c("left", "right"),  
    intent = NULL,  
    data_type = NULL,  
    encoding = NULL,  
    endian = c("LittleEndian", "BigEndian"),  
    col_names = NULL,  
    label_table = NULL
)
```
**write_metric_gifti**

**Arguments**

- **x**: A \(V \times T\) data matrix (\(V\) vertices, \(T\) measurements). This can also be an object from `gifti::readgii`, or a length \(T\) list of length \(V\) vectors.

- **gifti_fname**: Where to write the GIFTI file.

- **hemisphere**: "left" (default) or "right". Ignored if data is already a "gifti" object.

- **intent**: "NIFTI_INTENT_*". NULL (default) will use metadata if data is a "gifti" object, or "NONE" if it cannot be inferred. If not NULL and data is a "gifti" object, it will overwrite the existing intent. See [https://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields Pages/group__NIFTI1__INTENT__CODES.html#document_view](https://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields_pages/group__NIFTI1__INTENT__CODES.html#document_view).

- **data_type**: the type of data: "NIFTI_TYPE_*" where * is "INT32" or "FLOAT32". If NULL (default), the data type will be inferred. If not NULL and data is a "gifti" object, it will overwrite the existing data type.

- **encoding**: One of "ASCII", "Base64Binary", or "GZipBase64Binary". If NULL (default), will use the metadata if data is a GIFTI object, or "ASCII" if the data_type is "NIFTI_TYPE_INT32" and "GZipBase64Binary" if the data_type is "NIFTI_TYPE_FLOAT32". If not NULL and data is a "gifti" object, it will overwrite the existing data type.

- **col_names**: The names of each data column in gii (or entries in gii$data).

- **label_table**: A data.frame with labels along rows. The row names should be the label names. The column names should be among: "Key", "Red", "Green", "Blue", and "Alpha". The "Key" column is required whereas the others are optional (but very often included). Values in the "Key" column should be non-negative integers, typically beginning with 0. The other columns should be floating-point numbers between 0 and 1. Although CIFTI files support a different label table for each data column, GIFTI files only support a single label table. So this label table should be applicable to each data column.

**Value**

Whether the GIFTI was successfully written

**See Also**

Other functions for writing CIFTI or GIFTI metric data: `separate_cifti()`, `write_cifti()`, `write_subcort_nifti()`, `write_surf_gifti()`
write_subcort_nifti  Write subcortical data to NIFTI files

Description

Write subcortical data to NIFTI files representing the data values, subcortical structure labels, and volumetric mask. The input formats of subcortVol, subcortLabs, and subcortMask correspond to the data structures of xifti$data$subcort, xifti$meta$subcort$labels, and xifti$meta$subcort$mask respectively. subcortVol and subcortLabs should be vectorized, so if they are volumes consider using RNifti::writeNIfTI.

Usage

write_subcort_nifti(
  subcortVol,
  subcortLabs,
  subcortMask,
  trans_mat = NULL,
  subcortVol_fname,
  subcortLabs_fname,
  ROIsubcortVol_fname = NULL,
  fill = 0
)

Arguments

subcortVol  A vectorized data matrix: V voxels by T measurements
subcortLabs  Numeric (0 and 3-21) or factor vector corresponding to subcortical structure labels. See substructure_table.
subcortMask  Logical volumetric mask. Values of 0 represent out-of-mask voxels (not subcortical), and values of 1 represent in-mask voxels (subcortical),
trans_mat  The TransformationMatrixIJKtoXYZ, or equivalently the desired sform matrix (srow_x, srow_y and srow_z) to write. If NULL, do not write it (all zeroes).
subcortVol_fname, subcortLabs_fname, ROIsubcortVol_fname  File path to a NIFTI to save the corresponding data. ROIsubcortVol_fname is optional but the rest is required.
fill  Values to use for out-of-mask voxels. Default: 0.

Details

All file path arguments are required except ROIsubcortVol_fname. If not provided, the volumetric mask will not be written. (It's redundant with the 0 values in subcortLabs_fname because valid labels have positive indexes.)
write_surf_gifti

Value

Named character vector with the "subcortVol", "subcortLabs", and "ROIsubcortVol" file names (if written)

Connectome Workbench

This function interfaces with the "-volume-label-import" Workbench Command.

See Also

Other functions for writing CIFTI or GIFTI metric data: separate_cifti(), write_cifti(), write_metric_gifti(), write_surf_gifti()

---

write_surf_gifti  Write a "surf" to a GIFTI surface file

Description

Write the data for the left or right surface to a surface GIFTI file.

Usage

write_surf_gifti(
  x,
  gifti_fname,
  hemisphere = c("left", "right"),
  encoding = NULL,
  endian = c("LittleEndian", "BigEndian")
)

write_surf(
  x,
  gifti_fname,
  hemisphere = c("left", "right"),
  encoding = NULL,
  endian = c("LittleEndian", "BigEndian")
)

Arguments

x  A "surf" object, an object from gifti::readgii, or a list with elements "pointset" and "triangle".
gifti_fname  Where to write the GIFTI file.
hemisphere  "left" (default) or "right". Ignored if data is already a "gfti" object, or if it is a "surf" object with the hemisphere metadata already specified.
encoding

A length-2 vector with elements chosen among "ASCII", "Base64Binary", and "GZipBase64Binary". If NULL (default), will use the metadata if data is a "gifti" object, or "GZipBase64Binary" for the "pointset" and "ASCII" for the "triangles" if data is not already a GIFTI.

Value

Whether the GIFTI was successfully written

See Also

Other functions for writing CIFTI or GIFTI metric data: separate_cifti(), write_cifti(), write_metric_gifti(), write_subcort_nifti()

Other functions for working with GIFTI surface geometry data: is.surf(), read_surf(), resample_surf(), rotate_surf(), view_surf()
Index

* common
  is.cifti, 16
  read_cifti, 28
  resample_cifti, 35
  smooth_cifti, 45
  view_xifi_surface, 59
  view_xifi_volume, 65
  write_cifti, 67

* manipulating
  add_surf, 3
  apply_xifi, 4
  combine_xifi, 12
  merge_xifi, 23
  newdata_xifi, 26
  remove_xifi, 34
  select_xifi, 42
  transform_xifi, 52

* reading
  as.xifi, 5
  info_cifti, 8
  load_parc, 20
  load_surf, 21
  read_cifti, 28
  read_surf, 32
  read_xifi, 32

* surfing
  is.surf, 18
  read_surf, 32
  resample_surf, 39
  rotate_surf, 40
  view_surf, 55
  write_surf_gifti, 71

* writing
  separate_cifti, 43
  write_cifti, 67
  write_metric_gifti, 68
  write_subcort_nifti, 70
  write_surf_gifti, 71
  *.xifi (transform_xifi), 52
  +.xifi (transform_xifi), 52
  -.xifi (transform_xifi), 52
  /.xifi (transform_xifi), 52
  %/.xifi (transform_xifi), 52
  %%.xifi (transform_xifi), 52
  ^.%xifi (transform_xifi), 52
  abs.xifi (transform_xifi), 52
  add_surf, 3, 4, 12, 24, 26, 35, 43, 53
  apply_xifi, 4, 4, 12, 24, 26, 35, 43, 53
  as.cifti (as.xifi), 5
  as.matrix.xifi, 5
  as.xifi, 5, 16, 21, 31, 32, 34
  as_cifti (as.xifi), 5
  as_xifi (as.xifi), 5
  ceiling.xifi (transform_xifi), 52
  ciftiTools, 8
  ciftiTools.files, 10
  ciftiTools.getOption, 10
  ciftiTools.listOptions, 10, 11
  ciftiTools.setOption, 11
  combine_xifi, 4, 12, 24, 26, 35, 43, 53
  convert_to_dlabel, 4, 12, 24, 26, 35, 43, 53
  dim.xifi, 12
  exp.xifi (transform_xifi), 52
  expand_color_pal, 13
  fix_xifi, 13
  floor.xifi (transform_xifi), 52
  format, 63
  get_wb_cmd_path, 14
  info_cifti, 8, 14, 21, 30–32, 34
  infoCIfTI (info_cifti), 14
  infocii (info_cifti), 14
  is.cifti, 16, 31, 37, 47, 65, 66, 68
  is.surf, 18, 32, 40, 41, 50, 58, 72
is.xifti, 16, 18, 18
is_cifti (is.cifti), 16
is_xifti (is.xifti), 18
isCIfTI (is.cifti), 16

load_parx, 8, 16, 20, 21, 31, 32, 34
load_xifti, 8, 16, 20, 21, 31, 32, 34
log.xifti (transform_xifti), 52

make_color_pal, 22, 62
make_surf, 8, 50
make_surf (read_surf), 32
merge_xifti, 4, 12, 23, 26, 35, 43, 53
move_from_mwall, 24
move_to_mwall, 25
newdata_xifti, 4, 12, 24, 26, 35, 43, 53

parcBorders, 26
plot_surf, 27
plot.xifti, 28
print.summary.surf (summary.surf), 50
print.summary.xifti (summary.xifti), 50
print.surf (summary.surf), 50
print.xifti (summary.xifti), 50

read_cifti, 8, 16, 17, 21, 28, 32, 34, 37, 47, 65, 66, 68
read_cifti_convert, 30
read_cifti_separate, 30
read_surf, 8, 16, 18, 21, 31, 32, 34, 40, 41, 58, 61, 72
read_xifti (read_cifti), 28
read_xifti2, 8, 16, 21, 31, 32, 32
readCIfTI (read_cifti), 28
readcii (read_cifti), 28
readgii, 32
remove_xifti, 4, 12, 24, 26, 34, 43, 53
resample_cifti, 17, 31, 35, 47, 65, 66, 68
resample_cifti_default_fname, 36
resample_cifti_from_template, 37
resample giftedi, 38
resample_surf, 18, 32, 39, 41, 58, 72
resample_xifti (resample_cifti), 35
resampleCIfTI (resample_cifti), 35
resamplecii (resample_cifti), 35
resampleGIFTI (resample_gifti), 38
resamplelegii (resample_gifti), 38
rgl.close, 37, 64
rgl.viewpoint, 57, 64
rotate_surf, 18, 32, 40, 40, 58, 72
round.xifti (transform_xifti), 52
ROY_BIG_BL, 41
run_wb_cmd, 42

select_xifti, 4, 12, 24, 26, 35, 42, 53
separate_cifti, 43, 68, 69, 71, 72
separateCIfTI (separate_cifti), 43
separatcii (separate_cifti), 43
sign.xifti (transform_xifti), 52
smooth_cifti, 17, 31, 37, 45, 65, 66, 68
smooth_gifti, 48
smooth_xifti (smooth_cifti), 45
smoothCIfTI (smooth_cifti), 45
smoothcii (smooth_cifti), 45
smoothGIFTI (smooth_gifti), 48
smoothgii (smooth_gifti), 48
snapshot, 57, 64
sqrt.xifti (transform_xifti), 52
substructure_table, 7, 49, 70
summary.surf, 50
summary.xifti, 50
supported_intents, 51
sys_path, 51
system, 42, 51

template_xifti, 8, 14, 16, 19
transform_xifti, 4, 12, 24, 26, 35, 43, 52

unmask_cortex, 53
unmask_subcortex, 54
use_color_pal, 55

view_cifti (view_xifti), 58
view_cifti_surface
(view_xifti_surface), 59
view_cifti_volume (view_xifti_volume), 65
view_surf, 18, 32, 40, 41, 55, 72
view_xifti, 28, 58
view_xifti_surface, 17, 27, 31, 37, 40, 47, 57, 58, 59, 66, 68
view_xifti_volume, 17, 31, 37, 47, 58, 65, 65, 68
viewCIfTI (view_xifti), 58
viewCIfTI_surface (view_xifti_surface), 59
viewCIfTI_volume (view_xifti_volume), 65
viewcii (view_xifti), 58
viewcii_surface (view_xifti_surface), 59
viewcii_volume (view_xifti_volume), 65

write_cifti, 17, 31, 37, 45, 47, 65, 66, 67, 69, 71, 72
write_metric_gifti, 45, 68, 68, 71, 72
write_subcort_nifti, 45, 68, 69, 70, 72
write_surf (write_surf_gifti), 71
write_surf_gifti, 18, 32, 40, 41, 45, 58, 68, 69, 71, 71
write_xifti (write_cifti), 67
writeCIfTI (write_cifti), 67
writecii (write_cifti), 67