

# Package ‘ciftiTools’

June 17, 2021

**Type** Package

**Title** Tools for Reading and Visualizing CIFTI Brain Files

**Version** 0.3.1

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

**Description** CIFTI files contain brain imaging data in “gray-ordinates”, which represent the gray matter as cortical surface vertices (left and right) and subcortical voxels (cerebellum, basal ganglia, and other deep gray matter). ‘ciftiTools’ uses the Connectome Workbench to read CIFTI files into R and apply common pre-processing steps (e.g. smoothing, resampling). It also provides tools for visualizing the cortical surface with GIFTI files, and for visualizing the subcortical volume.

**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>>)

**Repository** CRAN

**Date/Publication** 2021-06-17 04:10:02 UTC

**R topics documented:**

add_surf . . . . .	3
as.matrix.xifti . . . . .	4
as.xifti . . . . .	4
ciftiTools . . . . .	7
ciftiTools.getOption . . . . .	8
ciftiTools.listOptions . . . . .	8
ciftiTools.setOption . . . . .	9
combine_xifti . . . . .	9
demo_files . . . . .	10
dim.xifti . . . . .	10
expand_color_pal . . . . .	11
get_wb_cmd_path . . . . .	11
info_cifti . . . . .	12
is.cifti . . . . .	14
is.surf . . . . .	15
is.xifti . . . . .	16
make_color_pal . . . . .	17
make_surf . . . . .	19
merge_xifti . . . . .	19
newdata_xifti . . . . .	20
parc_borders . . . . .	21
plot.surf . . . . .	21
plot.xifti . . . . .	22
read_cifti . . . . .	22
read_xifti2 . . . . .	25
remove_xifti . . . . .	27
resample_cifti . . . . .	27
resample_cifti_from_template . . . . .	30
resample_gifti . . . . .	31
resample_surf . . . . .	32
rotate_surf . . . . .	33
ROY_BIG_BL . . . . .	33
run_wb_cmd . . . . .	34
select_xifti . . . . .	35
separate_cifti . . . . .	35
smooth_cifti . . . . .	38
smooth_gifti . . . . .	40
substructure_table . . . . .	41
summary.surf . . . . .	42
summary.xifti . . . . .	42
supported_intents . . . . .	43
sys_path . . . . .	43
transform_xifti . . . . .	44
unmask_cortex . . . . .	45
unmask_vol . . . . .	46
use_color_pal . . . . .	46

<i>add_surf</i>	3
view_surf . . . . .	47
view_xifti . . . . .	49
view_xifti_surface . . . . .	50
view_xifti_volume . . . . .	56
write_cifti . . . . .	58
write_metric_gifti . . . . .	59
write_subcort_nifti . . . . .	60
write_surf_gifti . . . . .	61
<b>Index</b>	<b>63</b>

---

<code>add_surf</code>	<i>Add surface(s) to a "xifti" object</i>
-----------------------	---

---

### Description

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

### Usage

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

### Arguments

<code>xifti</code>	Object of class "xifti". See <a href="#">is.xifti</a> and <a href="#">make_xifti</a> .
<code>surfL</code>	(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.
<code>surfR</code>	(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.

---

as.matrix.xifti	<i>Convert a "xifti" to a matrix</i>
-----------------	--------------------------------------

---

**Description**

Converts a "xifti" to a matrix. Surfaces and metadata are discarded.

**Usage**

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

**Arguments**

x	The "xifti" object
...	Unused

**Value**

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

---

as.xifti	<i>Assemble a "xifti" object from data</i>
----------	--

---

**Description**

Assembles cortical data, subcortical data, and/or surface geometry to form a "xifti" object. The inputs must be data objects (vectors, matrices or arrays, depending on the argument). `as.xifti` can be used to combine the files written by `separate_cifti`, or read individual components independent of any CIFTI file.

**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,  
validate = TRUE  
)  
  
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  
)  
  
as_cifti(  
  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  
)  
  
as_cifti(  
  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  
)
```

### 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 ( $V_L \times T$  data matrix, with filler values e.g.  $\emptyset$  or NaN for medial wall vertices), or have data only for non-medial wall vertices ( $(V_L - mwall_L) \times T$  data matrix). The medial wall mask will be the  $\emptyset$  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 ( $V_R \times T$  data matrix, with filler values e.g.  $\emptyset$  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 ( $V_R \times T$  data matrix, with filler values e.g.  $\emptyset$  or NaN for medial wall vertices), or have data only for non-medial wall vertices ( $(V_R - mwall_R) \times T$  data matrix). The medial wall mask will be the  $\emptyset$  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 ( $ixjkk \times T$ ), 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 ( $ixjkk$ ) 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 ( $ixjkk$ ) where TRUE values indicate subcortical voxels (in-mask). If it is not provided, the mask will be inferred from voxels with labels  $\emptyset$ , 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 <a href="#">make_surf</a> 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" object? 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" object

---

ciftiTools

*ciftiTools: Tools for Reading and Visualizing CIFTI Brain Files*

---

### Description

Below is a list of "xifti"-related functions. (See `library(help="ciftiTools")` for a complete list of user functions.)

### Details

- `make_xifti`: Create one from data.
- `newdata_xifti`: Replace the data.
- `read_xifti`: Read in CIFTI data as a "xifti".
- `resample_xifti`: Resample the data.
- `smooth_xifti`: Smooth the data.
- `transform_xifti`: Apply univariate or bivariate functions over the data.
- `view_xifti_surface`: Plot the cortical data.
- `view_xifti_volume`: Plot the subcortical data.
- `write_xifti`: Write a "xifti" to a CIFTI file.

ciftiTools.getOption *Get 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 ciftiTools option*

---

**Description**

Sets an R option (with prefix "ciftiTools\_"). See [ciftiTools.listOptions](#).

**Usage**

```
ciftiTools.setOption(opt, val)
```

**Arguments**

<code>opt</code>	The option.
<code>val</code>	The value to set the option as.

**Value**

The new value, `val`

---

`combine_xifti` *Combine multiple "xifti" objects into one*

---

**Description**

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

**Usage**

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

**Arguments**

<code>...</code>	The "xifti" objects
<code>xii_list</code>	Alternatively, a list of "xifti" objects. Will ignore any other arguments from ...
<code>meta</code>	"first" (default) to just use the metadata from the first argument, or "all" to include the other metadata in a list.

**Value**

A "xifti" object

---

demo_files	<i>Get example files</i>
------------	--------------------------

---

**Description**

Get the file paths of the example CIFTI and surface GIFTI files included with `ciftiTools`.

**Usage**

```
demo_files()
```

**Details**

These files are from NITRC: `cifti-2_test_data-1.2.zip` at [https://www.nitrc.org/frs/?group\\_id=454](https://www.nitrc.org/frs/?group_id=454)

**Value**

A list with CIFTI file names in the first entry, and surface file names in the second.

---

dim.xifti	<i>Dimensions of a "xifti"</i>
-----------	--------------------------------

---

**Description**

Returns the number of rows and columns in the "xifti" data. Doesn't bother to validate the input.

**Usage**

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

**Arguments**

x	The "xifti" object
---	--------------------

**Value**

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

---

expand_color_pal	<i>Interpolates between entries in the input palette to make a larger palette with COLOR_RES entries.</i>
------------------	---

---

**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)

---

get_wb_cmd_path	<i>Get the Connectome Workbench command path</i>
-----------------	--

---

**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)
```

```
infoCIFTI(cifti_fname)
```

```
infocii(cifti_fname)
```

### Arguments

`cifti_fname`      File path of CIFTI-format data (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  $L \times 5$  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

**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.

**Connectome Workbench Requirement**

This function uses a system wrapper for the `'wb_command'` executable. The user must first download and install the Connectome Workbench, available from <https://www.humanconnectome.org/software/get-connectome-workbench>.

---

 is.cifti

 Validate a "xifti" object
 

---

### Description

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

### Usage

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

```
is_cifti(x, messages = TRUE)
```

```
isCIfti(x, messages = TRUE)
```

### 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.

---

<code>is.surf</code>	<i>Validate a surface (vertices + faces)</i>
----------------------	--

---

### 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

<code>x</code>	The putative surface.
----------------	-----------------------

### Details

This is a helper function for [is.xifti](#).

Requirements: the surface 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 surface?

---

is.xifti	<i>Validate a "xifti" object.</i>
----------	-----------------------------------

---

**Description**

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

**Usage**

```
is.xifti(x, messages = TRUE)
```

```
is_xifti(x, messages = TRUE)
```

**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.

---

`make_color_pal`      *Make a color palette.*

---

### Description

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

### Usage

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

### Arguments

<code>colors</code>	(Optional) "ROY_BIG_BL", the name of a ColorBrewer palette (see <code>RColorBrewer::brewer.pal.info</code> and <code>colorbrewer2.org</code> ), the name of a <code>viridisLite</code> palette, or a character vector of colors. NULL (default) will use "ROY_BIG_BL" if <code>color_mode</code> is "sequential" or "diverging", and "Set2" if <code>color_mode</code> is "qualitative". See the description for more details.
<code>color_mode</code>	(Optional) "sequential", "qualitative", or "diverging". Default: "sequential". See the description for more details.
<code>zlim</code>	(Optional) Controls the mapping of values to each color in <code>colors</code> . 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/F>). 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)

---

make_surf	<i>Convert input to a "surf" object</i>
-----------	---

---

**Description**

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

**Usage**

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

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

**Arguments**

**surf** Either a file path to a surface GIFTI; a "gifti" object 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.

**Value**

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

---

merge_xifti	<i>Concatenate "xifti" objects</i>
-------------	------------------------------------

---

**Description**

Concatenate "xifti" objects along the columns. They must have the same brainstructures. 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" objects to concatenate, OR the single argument xifti\_list which should be a list of "xifti" objects. (If xifti\_list is provided all other inputs will be ignored.)

**Value**

The concatenated "xifti" object.

---

newdata_xifti	<i>Replace the data in a "xifti"</i>
---------------	--------------------------------------

---

**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).
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"

parc\_borders *Parcellation borders*

**Description**

Identify vertices which lie on the border of different parcels.

**Usage**

```
parc_borders(parcel, surf = NULL, hemisphere = c("left", "right"))
```

**Arguments**

- parcel 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 `parcel` indicating if the vertex lies on a border.

plot.surf *S3 method: plot surface*

**Description**

Visualize a single surface

**Usage**

```
## 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	<i>S3 method: use view_xifti to plot a "xifti" object</i>
------------	---

---

**Description**

S3 method: use view\_xifti to plot a "xifti" object

**Usage**

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

**Arguments**

x	Object of class "xifti". See <a href="#">is.xifti</a> and <a href="#">make_xifti</a> .
...	Additional arguments to <a href="#">view_xifti</a> , except what, which will be set to NULL.

---

read_cifti	<i>Read a CIFTI file</i>
------------	--------------------------

---

**Description**

Read a CIFTI file as a "xifti" object.

**Usage**

```
read_cifti(
  cifti_fname = NULL,
  surfL_fname = NULL,
  surfR_fname = NULL,
  brainstructures = c("left", "right"),
  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"),
  resamp_res = NULL,
  flat = FALSE,
```

```

    mwall_values = c(NA, NaN),
    verbose = FALSE,
    ...
)

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

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

```

### Arguments

<code>cifti_fname</code>	File path of CIFTI-format data (ending in ".d*.nii").
<code>surfL_fname</code>	(Optional) File path of GIFTI surface geometry file representing the left cortex.
<code>surfR_fname</code>	(Optional) File path of GIFTI surface geometry file representing the right cortex.
<code>brainstructures</code>	Character vector indicating which brain structure(s) to obtain: "left" (left cortical surface), "right" (right cortical surface) and/or "subcortical" (subcortical and cerebellar gray matter). Can also be "all" (obtain all three brain structures). Default: <code>c("left", "right")</code> (cortical surface only). If a brain structure is indicated but does not exist, a warning will be raised and that brain structure will be skipped.
<code>resamp_res</code>	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 <code>-cifti-separate</code> , which is slower than <code>-cifti-convert -to-gifti-ext</code> .
<code>flat</code>	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. <code>flat==TRUE</code> 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 <code>-cifti-convert -to-gifti-ext</code> .
<code>mwall_values</code>	If the medial wall locations are not indicated in the CIFTI, use these values to infer the medial wall mask. Default: <code>c(NA, NaN)</code> . If NULL, do not attempt to infer the medial wall.
<code>verbose</code>	Should occasional updates be printed? Default: FALSE.
<code>...</code>	Additional arguments to <code>read_cifti_convert</code> or <code>read_cifti_separate</code> .

### 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 Requirement

This function uses a system wrapper for the 'wb\_command' executable. The user must first download and install the Connectome Workbench, available from <https://www.humanconnectome.org/software/get-connectome-workbench>.

### 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.

---

<code>read_xifti2</code>	<i>Read gifti file(s) as a xifti object</i>
--------------------------	---

---

### **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 ( $V_L x T$  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 ( $V_L x T$  data matrix, with filler values e.g. 0 or NaN for medial wall vertices), or have data only for non-medial wall vertices ( $(V_L - mwall_L) x T$  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, 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 ( $V_R x T$  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 cortexR\_mwall is provided, cortexR should either have data for all vertices on the right cortical surface ( $V_R x T$  data matrix, with filler values e.g. 0 or NaN for medial wall vertices), or have data only for non-medial wall vertices ( $(V_R - mwall_R) x T$  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, or resamp\_res must be set.

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.

---

remove_xifti	<i>Remove "xifti" component</i>
--------------	---------------------------------

---

**Description**

Remove a data or surface component from a "xifti"

**Usage**

```
remove_xifti(xifti, remove = NULL)
```

**Arguments**

xifti	Object of class "xifti". See <a href="#">is.xifti</a> and <a href="#">make_xifti</a> .
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 components removed

---

resample_cifti	<i>Resample CIFTI Data</i>
----------------	----------------------------

---

**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,  
  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 hemisphere).
- 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.
- 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.  
Correctly indicating the medial wall locations is important for resampling, because the medial wall mask is taken into account during resampling calculations.
- verbose** Should occasional updates be printed? Default: TRUE.

## Details

Can accept a "xifti" object as well as a path to a CIFTI-file.

## 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 Requirement**

This function uses a system wrapper for the 'wb\_command' executable. The user must first download and install the Connectome Workbench, available from <https://www.humanconnectome.org/software/get-connectome-workbench>.

---

`resample_cifti_from_template`*Resample a CIFTI from a template*

---

**Description**

Resample a CIFTI from a template CIFTI using the `-cifti-resample` Connectome Workbench command.

**Usage**

```
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

**Connectome Workbench Requirement**

This function uses a system wrapper for the 'wb\_command' executable. The user must first download and install the Connectome Workbench, available from <https://www.humanconnectome.org/software/get-connectome-workbench>.

---

resample_gifti	<i>Resample a metric GIFTI file (with its ROI)</i>
----------------	--

---

**Description**

Performs spatial resampling of GIFTI data on the cortical surface.

**Usage**

```
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  
)
```

```
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  
)
```

```
resamplegii(  
  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  
)
```

**Arguments**

original_fname	The GIFTI file to resample.
target_fname	Where to save the resampled file.
hemisphere	"left" (default) or "right". An error will occur if the hemisphere indicated in the GIFTI metadata does not match.
file_type	"metric", "label", "surf", or NULL (default) to infer from original_fname.
original_res	The resolution of the original file. If NULL (default), infer from the file.
resamp_res	Target resolution for resampling (number of cortical surface vertices per hemisphere).
ROIcortex_original_fname	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	The name of the resampled ROI file. Only applicable if ROIcortex_original_fname is provided.
read_dir	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	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

---

resample_surf	<i>Resample a "surf" object</i>
---------------	---------------------------------

---

**Description**

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

**Usage**

```
resample_surf(surf, resamp_res, hemisphere = c("left", "right"))
```

**Arguments**

surf	A "surf" object
resamp_res	The desired resolution
hemisphere	"left" or "right". Only used if not indicated by surf\$hemisphere. An error will be raised if it does not match the hemisphere indicated in the intermediate written GIFTI.



**Value**

The new surface

---

rotate_surf	<i>Rotate a "surface" object</i>
-------------	----------------------------------

---

**Description**

Rotate a "surface". 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"))
```

**Arguments**

surf	The "surface" object: see <a href="#">is.surf</a> .
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 <a href="#">view_xifti_surface</a> and other mesh rendering functions that use <code>rgl</code> , these rotations seem to correspond to yaw, pitch, and roll, respectively.
units	"radians" (default) or "degrees".

**Value**

The rotated "surface".

---

ROY_BIG_BL	<i>"ROY_BIG_BL" color palette</i>
------------	-----------------------------------

---

**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 <code>system</code> . Should be logical or NULL. If NULL (default), messages will be controlled by <code>ciftiTools.getOption("suppress_msgs")</code> 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	<i>Select columns of a "xifti" object</i>
--------------	---

---

**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	The "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.

**Value**

The "xifti" with only the selected columns.

---

separate_cifti	<i>Separate a CIFTI file</i>
----------------	------------------------------

---

**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
)

separatecii(
  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 of CIFTI-format data (ending in ".d\*.nii").

**brainstructures** Character vector indicating which brain structure(s) to obtain: "left" (left cortical surface), "right" (right cortical surface) and/or "subcortical" (subcortical and cerebellar gray matter). Can also be "all" (obtain all three brain structures). Default: c("left", "right") (cortical surface only).  
If a brain structure is indicated but does not exist, a warning will be raised 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.

ROIsubcortVol\_fname  
 (Optional) NIFTI 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 Requirement**

This function uses a system wrapper for the 'wb\_command' executable. The user must first download and install the Connectome Workbench, available from <https://www.humanconnectome.org/software/get-connectome-workbench>.

---

 smooth\_cifti

*Smooth a CIFTI*


---

### Description

Smooth CIFTI data. This uses the `-cifti-smoothing` command from Connectome Workbench.

### 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
)

```

### 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 very inflated 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 inflated surfaces included in "ciftiTools" will be used.

### Value

The cifti\_target\_fname, invisibly

### Connectome Workbench Requirement

This function uses a system wrapper for the 'wb\_command' executable. The user must first download and install the Connectome Workbench, available from <https://www.humanconnectome.org/software/get-connectome-workbench>.

---

smooth\_gifti

*Smooth a metric GIFTI*

---

### Description

Smooths metric GIFTI data on the cortical surface. This uses the `-metric-smoothing` command from Connectome Workbench.

### 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 very 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.
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

---

substructure_table	<i>Substructure table</i>
--------------------	---------------------------

---

**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	<i>Summarize cifti objects</i>
--------------	--------------------------------

---

### Description

Summary method for class "surf"

### Usage

```
## 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 <a href="#">is.surf</a> and <a href="#">make_surf</a> .
...	further arguments passed to or from other methods.
x	bject of class "surf".

---

summary.xifti	<i>Summarize cifti objects</i>
---------------	--------------------------------

---

### Description

Summary method for class "xifti"

### Usage

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

**Arguments**

object      Object of class "xifti".  
 ...        further arguments passed to or from other methods.  
 x            Object of class "xifti". See [is.xifti](#) and [make\\_xifti](#).

---

supported\_intents      *The NIFTI intents supported by ciftiTools*

---

**Description**

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

**Usage**

supported\_intents()

**Details**

See [https://www.nitrc.org/forum/attachment.php?attachid=334&group\\_id=454&forum\\_id=1955](https://www.nitrc.org/forum/attachment.php?attachid=334&group_id=454&forum_id=1955) for information about the different NIFTI intents.

**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 brainstructures and data dimensions.

**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)
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).

---

unmask_cortex	<i>Unmask cortex</i>
---------------	----------------------

---

### 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_vol	<i>Undo a volumetric mask</i>
------------	-------------------------------

---

**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.

**Usage**

```
unmask_vol(dat, mask, fill = NA)
```

**Arguments**

<code>dat</code>	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.
<code>mask</code>	Volumetric binary mask. TRUE indicates voxels inside the mask.
<code>fill</code>	The value for locations outside the mask. Default: NA.

**Value**

The 3D or 4D unflattened volume array

---

use_color_pal	<i>Use a color palette</i>
---------------	----------------------------

---

**Description**

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

**Usage**

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

**Arguments**

<code>data_values</code>	The values to map to colors
<code>pal</code>	The palette to use to map values to colors
<code>color_NA</code>	The color to use for NA values. Default: "white".
<code>indices</code>	Return the numeric indices of colors in <code>pal\$value</code> 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	<i>View "surf" object(s)</i>
-----------	------------------------------

---

**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**

```
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`.

<code>fname</code>	<p>Save the plot(s) (and color legend if applicable)?</p> <p>If <code>isFALSE(fname)</code> (default), no files will be written.</p> <p>If <code>fname</code> is a length-1 character vector ending in ".html", an html with an interactive widget will be written.</p> <p>If neither of the cases above apply, a png image will be written for each <code>idx</code>. If <code>isTRUE(fname)</code> the files will be named by the data column names (underscores will replace spaces). Set <code>fname</code> to a length 1 character vector to name files by this suffix followed by the <code>fname_suffix</code>: either the data column names ("names") or the index value ("idx"). Set this to a character vector with the same length as <code>idx</code> to name the files exactly.</p> <p>If a separate color legend exists and <code>!isFALSE(fname)</code>, it will be saved to "[first_surf_fname]_legend.png".</p>
<code>cex.title</code>	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.
<code>text_color</code>	Color for text in title and colorbar legend. Default: "black".
<code>bg</code>	Background color. NULL will not color the background (white).
<code>alpha</code>	Transparency value for mesh coloring, between 0 and 1. Default: 1.0 (no transparency).
<code>edge_color</code>	Outline each edge in this color. Default: NULL (do not outline the edges).
<code>vertex_color</code>	Draw each vertex in this color. Default: "black". Vertices are only drawn if <code>vertex_size &gt; 0</code>
<code>vertex_size</code>	Draw each vertex with this size. Default: 0 (do not draw the vertices).
<code>width</code>	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 ( <code>hemisphere</code> , <code>view</code> , <code>title</code> , and <code>slider_title</code> ) 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.)
<code>height</code>	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 ( <code>hemisphere</code> , <code>view</code> , <code>title</code> , and <code>slider_title</code> ) 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.)
<code>zoom</code>	Adjust the sizes of the brain meshes. Default: NULL (will be set to 0.6 or 160\ 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

Here are instructions for using the interactive Open GL window and html widget:

To navigate the plot, left click and drag the cursor to rotate. 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.

See the vignette for examples on embedding snapshots of the Open GL window or interactive html widgets in an R Markdown document.

## Embedding the Static Plots

If `save==TRUE`, the plot(s) is written to a .png file. (For `view_xifti_surface`, if `length(idx) > 1`, each `idx` will be written to a separate image file.) You can use [include\\_graphics](#) to embed an image file in an R Markdown document. If `close_after_save==TRUE`, the return value of this function call is the name(s) of the image file(s) that were written, so it can be used directly to display the image.

There's an additional way to embed an image of this plot without writing a .png file: use `save==FALSE` and set the chunk options `rgl=TRUE, format="png"`. You will probably need to tweak the image dimensions e.g. `fig.width=8, fig.height=5` in the chunk options, because it uses the defaults from RMarkdown/Knitr instead of what makes sense based on the dimensions of the Open GL window.

---

view_xifti	<i>View a "xifti" object</i>
------------	------------------------------

---

## 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	Object of class "xifti". See <a href="#">is.xifti</a> and <a href="#">make_xifti</a> .
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`.

---

<code>view_xifti_surface</code>	<i>View cortical surface data</i>
---------------------------------	-----------------------------------

---

**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**

```
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
)

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
)

viewCifTI_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  
)
```

```
viewcii_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
)

```

## Arguments

xifti	Object of class "xifti". See <a href="#">is.xifti</a> and <a href="#">make_xifti</a> .
surfL, surfR	(Optional) The brain surface model to use. Each can be a file path for a GIFTI, a file read by <code>gifti::readgii</code> , or a list with components "vertices" and "faces". If provided, they will override <code>xifti\$surf\$cortex_left</code> and <code>xifti\$surf\$cortex_right</code> if those exist. Leave as NULL (default) to use <code>xifti\$surf\$cortex_left</code> and <code>xifti\$surf\$cortex_right</code> if those exist, or the default inflated surfaces if those do 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 $>90\%$ <a href="#">make_color_pal</a> for more details.
zlim	(Optional) Controls the mapping of values to each color in colors. If the length is longer than one, using <code>-Inf</code> will set the value to the data minimum, and <code>Inf</code> will set the value to the data maximum. See <a href="#">make_color_pal</a> description for more details.
colors	(Optional) "ROY_BIG_BL", vector of colors to use, the name of a ColorBrewer palette (see <code>RColorBrewer::brewer.pal.info</code> and <a href="#">colorbrewer2.org</a> ), 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 <a href="#">make_color_pal</a> for more details.
idx	The time/column index of the data to display. If its length is greater than one, and <code>isFALSE(fname)</code> , 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 surface included with <code>ciftiTools</code> will be used for each cortex with data (i.e. if <code>xifti\$data\$cortex_left</code> and/or <code>xifti\$data\$cortex_right</code> 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 <code>htmlwidget</code> ? Should be logical or NULL (default), in which case a widget will be used only if needed ( <code>length(idx)&gt;1 &amp; isFALSE(fname)</code> ), <code>fname</code> is a file path to an .html file, or if <code>rgl.useNULL()</code> .

title	<p>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.</p> <p>Default: NULL will not use any title if <code>length(idx)==1</code>. Otherwise, it will use the time index (<code>".dtseries"</code>) or name (<code>".dscalar</code> or <code>".dlabel</code>) of each data column.</p> <p>To use a custom title(s), use a length 1 character vector (same title for each plot) or length <code>length(idx)</code> character vector (different title for each plot). Set to NULL or an empty character to omit the title.</p> <p>If the title is non-empty but does not appear, try lowering <code>cex.title</code>.</p>
slider_title	<p>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.</p>
fname	<p>Save the plot(s) (and color legend if applicable)?</p> <p>If <code>isFALSE(fname)</code> (default), no files will be written.</p> <p>If <code>fname</code> is a length-1 character vector ending in <code>".html"</code>, an html with an interactive widget will be written.</p> <p>If neither of the cases above apply, a png image will be written for each <code>idx</code>. If <code>isTRUE(fname)</code> the files will be named by the data column names (underscores will replace spaces). Set <code>fname</code> to a length 1 character vector to name files by this suffix followed by the <code>fname_suffix</code>: either the data column names (<code>"names"</code>) or the index value (<code>"idx"</code>). Set this to a character vector with the same length as <code>idx</code> to name the files exactly.</p> <p>If a separate color legend exists and <code>!isFALSE(fname)</code>, it will be saved to <code>"[first_surf_fname]_legend.png"</code>.</p>
fname_suffix	<p>Save the plot(s) (and color legend if applicable)?</p> <p>If <code>isFALSE(fname)</code> (default), no files will be written.</p> <p>If <code>fname</code> is a length-1 character vector ending in <code>".html"</code>, an html with an interactive widget will be written.</p> <p>If neither of the cases above apply, a png image will be written for each <code>idx</code>. If <code>isTRUE(fname)</code> the files will be named by the data column names (underscores will replace spaces). Set <code>fname</code> to a length 1 character vector to name files by this suffix followed by the <code>fname_suffix</code>: either the data column names (<code>"names"</code>) or the index value (<code>"idx"</code>). Set this to a character vector with the same length as <code>idx</code> to name the files exactly.</p> <p>If a separate color legend exists and <code>!isFALSE(fname)</code>, it will be saved to <code>"[first_surf_fname]_legend.png"</code>.</p>
legend_fname	<p>Save the color legend? Since the legend is the same for each <code>idx</code> only one legend is written even if <code>length(idx)&gt;1</code>. This argument can be NULL to not save the legend, an exact file path, or a length-one character vector with <code>"[fname]"</code> in it, which will name the legend based on <code>fname\[1\]</code>. For example, if <code>fname\[1\]</code> is <code>"plots/my_cifti.png"</code> and <code>legend_fname</code> is <code>"\[fname\]_legend"</code> (default), then the legend plot will be saved to <code>"plots/my_cifti_legend.png"</code>.</p>
legend_ncol	<p>Number of columns in color legend. If NULL (default), use 10 entries per row. Only applies if the color legend is used (qualitative data).</p>
legend_alllevels	<p>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).</p>

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 <code>format</code> 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 <code>color_mode</code> 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 <code>vertex_size &gt; 0</code>
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 ( <code>hemisphere</code> , <code>view</code> , <code>title</code> , and <code>slider_title</code> ) 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 ( <code>hemisphere</code> , <code>view</code> , <code>title</code> , and <code>slider_title</code> ) 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\ 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

Here are instructions for using the interactive Open GL window and html widget:

To navigate the plot, left click and drag the cursor to rotate. 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.

See the vignette for examples on embedding snapshots of the Open GL window or interactive html widgets in an R Markdown document.

## Embedding the Static Plots

If `save==TRUE`, the plot(s) is written to a .png file. (For `view_xifti_surface`, if `length(idx) > 1`, each `idx` will be written to a separate image file.) You can use `include_graphics` to embed an image file in an R Markdown document. If `close_after_save==TRUE`, the return value of this function call is the name(s) of the image file(s) that were written, so it can be used directly to display the image.

There's an additional way to embed an image of this plot without writing a .png file: use `save==FALSE` and set the chunk options `rgl=TRUE, format="png"`. You will probably need to tweak the image dimensions e.g. `fig.width=8, fig.height=5` in the chunk options, because it uses the defaults from RMarkdown/Knitr instead of what makes sense based on the dimensions of the Open GL window.

---

view\_xifti\_volume      *View subcortex*

---

## Description

Visualize subcortex of a "xifti" object

## Usage

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

```
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

<code>xifti</code>	Object of class "xifti". See <a href="#">is.xifti</a> and <a href="#">make_xifti</a> .
<code>structural_img</code>	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.
<code>idx</code>	The time/column index of the "xifti" data to plot. Must be a single index (length 1).
<code>plane</code>	If <code>interactive=FALSE</code> , the plane to display. Default: "axial". Other options are "sagittal" and "coronal".
<code>num.slices</code>	If <code>interactive=FALSE</code> , the number of slices to display. Default: 9.
<code>interactive</code>	<code>interactive=TRUE</code> will use <code>papayar</code> to allow for interactive visualization.
<code>zlim</code>	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

---

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

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

writeCifTI(
  xifti,
  cifti_fname,
  surfL_fname = NULL,
  surfR_fname = NULL,
  verbose = TRUE
)

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

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

**Arguments**

xifti	Object of class "xifti". See <a href="#">is.xifti</a> and <a href="#">make_xifti</a> .
cifti_fname	File path of CIFTI-format data (ending in ".d*.nii").
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 Requirement**

This function uses a system wrapper for the 'wb\_command' executable. The user must first download and install the Connectome Workbench, available from <https://www.humanconnectome.org/software/get-connectome-workbench>.

---

write\_metric\_gifti      *Write CIFTI cortex data to GIFTI*

---

**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
)
```

**Arguments**

x	A $V \times T$ data matrix ( $V$ vertices, $T$ measurements). This can also be an object from <code>gifti::readgii</code> , 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 <a href="https://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields_pages/group__NIFTI1__INTENT__CODES.html/document_">https://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields_pages/group__NIFTI1__INTENT__CODES.html/document_</a>
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.
endian	"LittleEndian" (default) or "BigEndian". If data is a "gifti" object, it will overwrite the existing endian.
col_names	The names of each data column in <code>gii</code> (or entries in <code>gii\$data</code> ).
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

---

`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 <a href="#">substructure_table</a> .
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.)

### Value

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

---

write_surf_gifti	<i>Write CIFTI surface data to GIFTI</i>
------------------	--

---

### 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")  
)
```

**Arguments**

x	A "surf" object, an object from <code>gifti::readgii</code> , 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 "gifti" 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.
endian	"LittleEndian" (default) or "BigEndian".

**Value**

Whether the GIFTI was successfully written

# Index

`*.xifti` (`transform_xifti`), 44  
`+.xifti` (`transform_xifti`), 44  
`-.xifti` (`transform_xifti`), 44  
`/.xifti` (`transform_xifti`), 44  
`%/.xifti` (`transform_xifti`), 44  
`%%.xifti` (`transform_xifti`), 44  
`^.xifti` (`transform_xifti`), 44

`abs.xifti` (`transform_xifti`), 44  
`add_surf`, 3  
`as.cifti` (`as.xifti`), 4  
`as.matrix.xifti`, 4  
`as.xifti`, 4  
`as_cifti` (`as.xifti`), 4  
`as_xifti` (`as.xifti`), 4

`ceiling.xifti` (`transform_xifti`), 44  
`ciftiTools`, 7  
`ciftiTools.getOption`, 8  
`ciftiTools.listOptions`, 8, 8, 9  
`ciftiTools.setOption`, 9  
`combine_xifti`, 9

`demo_files`, 10  
`dim.xifti`, 10

`exp.xifti` (`transform_xifti`), 44  
`expand_color_pal`, 11

`floor.xifti` (`transform_xifti`), 44  
`format`, 55

`get_wb_cmd_path`, 11

`include_graphics`, 49, 56  
`info_cifti`, 12, 24  
`infoCIFTI` (`info_cifti`), 12  
`infocii` (`info_cifti`), 12  
`is.cifti`, 14  
`is_surf`, 15, 33, 42

`is.xifti`, 3, 14, 15, 16, 22, 27, 43, 50, 53, 57, 59  
`is_cifti` (`is.cifti`), 14  
`is_xifti` (`is.xifti`), 16  
`isCIFTI` (`is.cifti`), 14

`log.xifti` (`transform_xifti`), 44

`make_color_pal`, 17, 53  
`make_surf`, 7, 19, 42  
`make_xifti`, 3, 22, 27, 43, 50, 53, 57, 59  
`merge_xifti`, 19

`newdata_xifti`, 20

`parc_borders`, 21  
`plot_surf`, 21  
`plot_xifti`, 22  
`print.summary_surf` (`summary_surf`), 42  
`print.summary_xifti` (`summary_xifti`), 42  
`print_surf` (`summary_surf`), 42  
`print_xifti` (`summary_xifti`), 42

`read_cifti`, 22  
`read_cifti_convert`, 24  
`read_cifti_separate`, 24  
`read_surf` (`make_surf`), 19  
`read_xifti` (`read_cifti`), 22  
`read_xifti2`, 25  
`readCIFTI` (`read_cifti`), 22  
`readcii` (`read_cifti`), 22  
`readgii`, 19  
`remove_xifti`, 27  
`resample_cifti`, 27  
`resample_cifti_default_fname`, 29  
`resample_cifti_from_template`, 30  
`resample_gifti`, 31  
`resample_surf`, 32  
`resample_xifti` (`resample_cifti`), 27  
`resampleCIFTI` (`resample_cifti`), 27  
`resamplecii` (`resample_cifti`), 27

resampleGIfTI (resample\_gifti), 31  
 resamplegii (resample\_gifti), 31  
 rgl.close, 49, 56  
 rgl.viewpoint, 49, 56  
 rotate\_surf, 33  
 round.xifti (transform\_xifti), 44  
 ROY\_BIG\_BL, 33  
 run\_wb\_cmd, 34  
  
 select\_xifti, 35  
 separate\_cifti, 4, 35  
 separateCIfTI (separate\_cifti), 35  
 separatecii (separate\_cifti), 35  
 sign.xifti (transform\_xifti), 44  
 smooth\_cifti, 38  
 smooth\_gifti, 40  
 smooth\_xifti (smooth\_cifti), 38  
 smoothCIfTI (smooth\_cifti), 38  
 smoothcii (smooth\_cifti), 38  
 smoothGIfTI (smooth\_gifti), 40  
 smoothgii (smooth\_gifti), 40  
 snapshot, 49, 56  
 sqrt.xifti (transform\_xifti), 44  
 substructure\_table, 6, 41, 61  
 summary\_surf, 42  
 summary\_xifti, 42  
 supported\_intents, 43  
 sys\_path, 43  
 system, 34, 43  
  
 template\_xifti, 7, 12, 14, 16  
 transform\_xifti, 44  
  
 unmask\_cortex, 45  
 unmask\_vol, 46  
 use\_color\_pal, 46  
  
 view\_cifti (view\_xifti), 49  
 view\_cifti\_surface  
     (view\_xifti\_surface), 50  
 view\_cifti\_volume (view\_xifti\_volume),  
     56  
 view\_surf, 47  
 view\_xifti, 22, 49  
 view\_xifti\_surface, 21, 33, 49, 50  
 view\_xifti\_volume, 49, 56  
 viewCIfTI (view\_xifti), 49  
 viewCIfTI\_surface (view\_xifti\_surface),  
     50  
  
 viewCIfTI\_volume (view\_xifti\_volume), 56  
 viewcii (view\_xifti), 49  
 viewcii\_surface (view\_xifti\_surface), 50  
 viewcii\_volume (view\_xifti\_volume), 56  
  
 write\_cifti, 58  
 write\_metric\_gifti, 59  
 write\_subcort\_nifti, 60  
 write\_surf\_gifti, 61  
 write\_xifti (write\_cifti), 58  
 writeCIfTI (write\_cifti), 58  
 writecii (write\_cifti), 58