

Package ‘skiftiTools’

October 16, 2025

Title Tools and Operations for Reading, Writing, Viewing, and
Manipulating SKIFTI Files

Version 0.1.0

Description SKIFTI files contain brain imaging data in coordinates across Tract Based Spatial Statistics (TBSS) skeleton, which represent the brain white matter intensity values. 'skiftiTools' provides a unified environment for reading, writing, visualizing and manipulating SKIFTI-format data. It supports the ``subsetting'', ``concatenating'', and using data as data.frame for R statistical functions. The SKIFTI data is structured for convenient access to the data and metadata, and includes support for visualizations. For more information see Merisaari et al. (2024) <[doi:10.57736/87d2-0608](https://doi.org/10.57736/87d2-0608)>.

Depends R (>= 4.2.0)

License GPL-3

Encoding UTF-8

Imports RNifti, stringr, R.utils, rmarchingcubes, Rvcg, png, rgl, oce,
abind, methods, s2dv

RoxygenNote 7.3.3

URL <https://github.com/haanme/skiftiTools>

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Harri Merisaari [aut] (ORCID: <<https://orcid.org/0000-0002-8515-5399>>),
Ilkka Suuronen [cre] (ORCID: <<https://orcid.org/0009-0001-6516-4116>>)

Maintainer Ilkka Suuronen <ilksuu@utu.fi>

Repository CRAN

Date/Publication 2025-10-16 12:10:02 UTC

Contents

concat	2
Nifti2Skifti	3
readSkifti	5
save_skeleton	6
Skifti2CSV	7
Skifti2Nifti	8
subset	9
writeSkifti	9
Index	12

concat	<i>Concatenate Skifti data</i>
--------	--------------------------------

Description

Concatenate Skifti data

Usage

concat(Skifti_data1, Skifti_data2)

Arguments

Skifti_data1 Skifti data object1
Skifti_data2 Skifti data object2

Value

concatenated Skifti data object

Examples

```
library(RNifti)
data<-array(0,dim=list(10,10,10,10))
for(t in 1:10) {
  for(x in 1:10) {
    for(y in 1:10) {
      for(z in 1:10) {
        data[x,y,z,t]<-t+x
      }
    }
  }
}
data_Nifti<-RNifti::retrieveNifti(data)
RNifti::writeNifti(data_Nifti, "data_Nifti.nii.gz", template = NULL, datatype = "auto")

data_skeleton<-array(0,dim=list(10,10,10))
```

```

data_skeleton[5,5,5]<-1
data_skeleton[6,6,6]<-1
data_skeleton[7,7,7]<-1
data_skeleton_Nifti<-RNifti::retrieveNifti(data_skeleton)
RNifti::writeNifti(data_skeleton_Nifti, "data_skeleton_Nifti.nii.gz", datatype = "auto")

data_Skifti<-Nifti2Skifti(Nifti_data="data_Nifti.nii.gz",
                        Nifti_skeleton="data_skeleton_Nifti.nii.gz",
                        selected_volumes=1:10,
                        Nifti_labels=NULL,
                        write_coordinates=TRUE,
                        verbose=FALSE)

data_Skifti_subset<-subset(data_Skifti, c(1,5,10))
m<-matrix(c(6,10,15,7,11,16,8,12,17), nrow=3, ncol=3)
rownames(m)<-c("vol1", "vol5", "vol10")

data_Skifti_subset1<-subset(data_Skifti, c(1,5))
data_Skifti_subset2<-subset(data_Skifti, c(10))
data_Skifti_concat<-concat(data_Skifti_subset1, data_Skifti_subset2)
m<-matrix(c(6,10,15,7,11,16,8,12,17), nrow=3, ncol=3)
rownames(m)<-c("vol1", "vol5", "")

```

Nifti2Skifti

Create a SKIFTI file from fsl TBSS skeleton data

Description

Skeleton mask and corresponding image intensity data must be in Nifti format. The skeleton mask is used to determine the coordinates of intensity data. If optional label file is given, that is used to label the voxels.

Usage

```

Nifti2Skifti(
  Nifti_data = NULL,
  Nifti_skeleton = NULL,
  selected_volumes = NULL,
  Nifti_labels = NULL,
  write_coordinates = FALSE,
  verbose = FALSE
)

```

Arguments

Nifti_data Intensity data in Nifti format (required)

Nifti_skeleton Skeleton at same imaging space as the data, in Nifti format (required)

selected_volumes Selected volume indexes starting from 1 (default==NULL, selecting all)

readSkifti	<i>Read Skifti data</i>
------------	-------------------------

Description

Read Skifti data

Usage

```
readSkifti(filename, verbose = FALSE)
```

Arguments

filename	file to read
verbose	TRUE/FALSE(default), for verbosity

Value

Skifti data object

Examples

```
#source('.../R/Skifti2Nifti.R')
#source('.../R/Nifti2Skifti.R')
library(RNifti)
data<-array(0,dim=list(10,10,10,10))
for(t in 1:10) {
  for(x in 1:10) {
    for(y in 1:10) {
      for(z in 1:10) {
        data[x,y,z,t]<-t+x
      }
    }
  }
}
data_Nifti<-RNifti::retrieveNifti(data)
RNifti::writeNifti(data_Nifti, "data_Nifti.nii.gz", template = NULL, datatype = "auto")

data_skeleton<-array(0,dim=list(10,10,10))
data_skeleton[5,5,5]<-1
data_skeleton[6,6,6]<-1
data_skeleton[7,7,7]<-1
data_skeleton_Nifti<-RNifti::retrieveNifti(data_skeleton)
RNifti::writeNifti(data_skeleton_Nifti, "data_skeleton_Nifti.nii.gz", datatype = "auto")

data_Skifti<-Nifti2Skifti(Nifti_data="data_Nifti.nii.gz",
                          Nifti_skeleton="data_skeleton_Nifti.nii.gz",
                          selected_volumes=c(1),
                          Nifti_labels=NULL,
```

```

write_coordinates=TRUE,
verbose=FALSE)

# Create Skifti
data_Nifti2<-Skifti2Nifti(data_Skifti)
RNifti::writeNifti(data_Nifti2[[1]], "data_Nifti.nii.gz", datatype = "auto")
data_Nifti2<-RNifti::readNifti("data_Nifti.nii.gz", internal = TRUE, volumes = NULL)

```

save_skeleton

Create png from mask and data in Nifti format

Description

Skeleton mask and corresponding image intensity data must be in Nifti format. The skeleton mask is used to determine the coordinates of intensity data.

Usage

```

save_skeleton(
  mask,
  data,
  img_hdr,
  output,
  legend_title,
  scale,
  keep_temp = FALSE,
  palette = "lajolla",
  verbose = FALSE
)

```

Arguments

mask	Intensity data in Nifti object format
data	Skeleton at same imaging space as the data, in Nifti format
img_hdr	Nifti header object
output	Output PNG filename
legend_title	Title to be shown
scale	scaling for intensity values, tune for better color depth
keep_temp	TRUE/FALSE(default) to keep temporary png images
palette	color palette
verbose	TRUE/FALSE(default), for verbosity

Value

No output, as results are saved to a png file

Skifti2CSV

*Create a Nifti file from Skifti data***Description**

Skeleton mask and corresponding image intensity data in Nifti format. The skeleton mask is used to determine the coordinates of intensity data. If optional label file is given, that is used to label the voxels.

Usage

```
Skifti2CSV(Skifti_data, filename, overwrite = FALSE, sep = ";")
```

Arguments

Skifti_data	Intensity data in Nifti format
filename	file to read'
overwrite	TRUE/FALSE(default) to overwrite existing data
sep	file separator to be written default ';'.

Value

CSV filename

Examples

```
library(RNifti)
data<-array(0,dim=list(10,10,10,10))
for(t in 1:10) {
  for(x in 1:10) {
    for(y in 1:10) {
      for(z in 1:10) {
        data[x,y,z,t]<-t+x
      }
    }
  }
}
data_Nifti<-RNifti::retrieveNifti(data)
RNifti::writeNifti(data_Nifti, "data_Nifti.nii.gz", template = NULL, datatype = "auto")

data_skeleton<-array(0,dim=list(10,10,10))
data_skeleton[5,5,5]<-1
data_skeleton[6,6,6]<-1
data_skeleton[7,7,7]<-1
data_skeleton_Nifti<-RNifti::retrieveNifti(data_skeleton)
RNifti::writeNifti(data_skeleton_Nifti, "data_skeleton_Nifti.nii.gz", datatype = "auto")

data_Skifti<-Nifti2Skifti(Nifti_data="data_Nifti.nii.gz",
```

```

Nifti_skeleton="data_skeleton_Nifti.nii.gz",
selected_volumes=1:10,
Nifti_labels=NULL,
write_coordinates=TRUE,
verbose=FALSE)

Skifti2CSV(data_Skifti, "data_Skifti.csv", overwrite=TRUE, sep=';')
data_csv<-read.csv2("data_Skifti.csv", ';', header = FALSE, row.names = NULL)

```

Skifti2Nifti

Create a Nifti file from Skifti data

Description

Skeleton mask and corresponding image intensity data in Nifti format. The skeleton mask is used to determine the coordinates of intensity data. If optional label file is given, that is used to label the voxels.

Usage

```
Skifti2Nifti(Skifti_data)
```

Arguments

Skifti_data Intensity data in Nifti format

Value

Nifti skeleton file for Skifti data

Examples

```

#source('.../R/Skifti2Nifti.R')
#source('.../R/Nifti2Skifti.R')
library(RNifti)
data<-array(0,dim=list(10,10,10,10))
for(t in 1:10) {
  for(x in 1:10) {
    for(y in 1:10) {
      for(z in 1:10) {
        data[x,y,z,t]<-t+x
      }
    }
  }
}
data_Nifti<-RNifti::retrieveNifti(data)
RNifti::writeNifti(data_Nifti, "data_Nifti.nii.gz", template = NULL, datatype = "auto")

data_skeleton<-array(0,dim=list(10,10,10))
data_skeleton[5,5,5]<-1

```



```
data_skeleton[6,6,6]<-1
data_skeleton[7,7,7]<-1
data_skeleton_Nifti<-RNifti::retrieveNifti(data_skeleton)
RNifti::writeNifti(data_skeleton_Nifti, "data_skeleton_Nifti.nii.gz", datatype = "auto")

data_Skifti<-Nifti2Skifti(Nifti_data="data_Nifti.nii.gz",
                        Nifti_skeleton="data_skeleton_Nifti.nii.gz",
                        selected_volumes=c(1),
                        Nifti_labels=NULL,
                        write_coordinates=TRUE,
                        verbose=FALSE)

# Create Skifti
data_Nifti2<-Skifti2Nifti(data_Skifti)
RNifti::writeNifti(data_Nifti2[[1]], "data_Nifti.nii.gz", datatype = "auto")
data_Nifti2<-RNifti::readNifti("data_Nifti.nii.gz", internal = TRUE, volumes = NULL)
```

subset	<i>Get subset of Skifti data</i>
--------	----------------------------------

Description

Get subset of Skifti data

Usage

```
subset(Skifti_data, volumes)
```

Arguments

Skifti_data	Skifti data object
volumes	selection

Value

Skifti data object of subset

writeSkifti	<i>Write Skifti data</i>
-------------	--------------------------

Description

Write Skifti data

Usage

```
writeSkifti(
  Skifti_data,
  basename,
  overwrite = FALSE,
  compress = "none",
  verbose = FALSE
)
```

Arguments

Skifti_data	Skifti data object
basename	basename to write without suffix
overwrite	TRUE/FALSE(default) to overwrite existing data
compress	bz2/zip/none(default) to select compression method
verbose	TRUE/FALSE(default), for verbosity

Value

filename where Skifti data was written

Examples

```
#source('.../R/Skifti2Nifti.R')
#source('.../R/Nifti2Skifti.R')
library(RNifti)
data<-array(0,dim=list(10,10,10,10))
for(t in 1:10) {
  for(x in 1:10) {
    for(y in 1:10) {
      for(z in 1:10) {
        data[x,y,z,t]<-t+x
      }
    }
  }
}
data_Nifti<-RNifti::retrieveNifti(data)
RNifti::writeNifti(data_Nifti, "data_Nifti.nii.gz", template = NULL, datatype = "auto")

data_skeleton<-array(0,dim=list(10,10,10))
data_skeleton[5,5,5]<-1
data_skeleton[6,6,6]<-1
data_skeleton[7,7,7]<-1
data_skeleton_Nifti<-RNifti::retrieveNifti(data_skeleton)
RNifti::writeNifti(data_skeleton_Nifti, "data_skeleton_Nifti.nii.gz", datatype = "auto")

data_Skifti<-Nifti2Skifti(Nifti_data="data_Nifti.nii.gz",
  Nifti_skeleton="data_skeleton_Nifti.nii.gz",
  selected_volumes=c(1),
  Nifti_labels=NULL,
```

```
        write_coordinates=TRUE,  
        verbose=FALSE)  
  
# Create Skifti  
data_Nifti2<-Skifti2Nifti(data_Skifti)  
RNifti::writeNifti(data_Nifti2[[1]], "data_Nifti.nii.gz", datatype = "auto")  
data_Nifti2<-RNifti::readNifti("data_Nifti.nii.gz", internal = TRUE, volumes = NULL)
```

Index

concat, [2](#)

Nifti2Skifti, [3](#)

readSkifti, [5](#)

save_skeleton, [6](#)

Skifti2CSV, [7](#)

Skifti2Nifti, [8](#)

subset, [9](#)

writeSkifti, [9](#)