Package ‘rsurfer’

October 30, 2017

Version 0.2
Date 2017-10-30
Title Manipulating 'Freesurfer' Generated Data
Maintainer Alexander Luke Spedding <alexspedding271@gmail.com>
Depends R (>= 3.3.3), stringr (>= 1.1.0), gdata (>= 2.17.0)
Description The software suite, 'Freesurfer', is a open-source software suite involving the segmentation of brain MRIs (see <http://freesurfer.net/> for more information). This package provides functionality to import the data generated by 'Freesurfer'; functions to easily manipulate the data; and provides brain specific normalisation commonly used when studying structural brain MRIs. This package has been designed using an installation of and data generated from 'Freesurfer' version 5.3.
License MIT + file LICENSE
BugReports https://github.com/AlexDiru/rsurfer/issues
RoxygenNote 6.0.1
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Alexander Luke Spedding [aut, cre], Fabio Cappello [aut], Giuseppe di Fatta [aut]
Repository CRAN
Date/Publication 2017-10-30 22:49:46 UTC

R topics documented:

rsurfer-package ......................................................... 3
addrandomdiagnosis ................................................. 3
addrandomgender .................................................... 4
adni.mergewithfreesurferoutput .................................. 5
adni.printfilelocations ............................................ 5
adni.setfiles ....................................................... 6
### R topics documented:

- `caddementia.mergewithfreesurferoutput` ........................................ 7
- `caddementia.printfilelocations` .................................................. 8
- `caddementia.setfiles` ................................................................. 8
- `eliminateabnormalities` ............................................................. 9
- `eliminateabnormalities.cols` ...................................................... 9
- `eliminateabnormalities.rows` ..................................................... 10
- `export.forKNIME` ....................................................................... 11
- `extract.brain.features` ............................................................... 11
- `extract.byname` .......................................................................... 12
- `extract.cortical` ......................................................................... 13
- `extract.corticalsurfaceareas` ....................................................... 14
- `extract.corticalthicknesses` ......................................................... 14
- `extract.corticalthicknesssstddevs` ............................................... 15
- `extract.corticalvolumes` .............................................................. 16
- `extract.hippocampalvolumes` ......................................................... 17
- `extract.subcorticalvolumes` ........................................................... 17
- `extract.volumes` ........................................................................... 18
- `fsdirectorycheck` .......................................................................... 19
- `fsimport` ..................................................................................... 19
- `fsimport.listfields` ...................................................................... 20
- `fsimport.serialise` ...................................................................... 21
- `generaterandomsubjects` .............................................................. 22
- `get.hemisphere.side` ................................................................... 23
- `get.opposite.hemisphere.measurement` .......................................... 23
- `getfieldgroup` ............................................................................. 24
- `getfshome` ................................................................................... 25
- `getfsversion` .............................................................................. 25
- `getnamesofareas` ......................................................................... 26
- `getnamesofthicknesses` ............................................................... 27
- `getnamesofvolumes` ..................................................................... 27
- `is.cortical` ................................................................................... 28
- `is.corticalarea` ............................................................................ 29
- `is.corticalthickness` ..................................................................... 29
- `is.corticalthicknessssdtdevs` ....................................................... 30
- `is.corticalvolume` ........................................................................ 31
- `is.hippocampalvolume` ................................................................. 31
- `is.subcorticalvolume` ................................................................... 32
- `isfshome` ..................................................................................... 33
- `ixi.mergewithfreesurferoutput` .................................................... 33
- `ixi.setfile` ................................................................................... 34
- `normalise` .................................................................................... 34
- `normalise.listfieldgroups` ........................................................... 36
- `normalise.listmethods` ............................................................... 36
- `removeabnormalrowsandcols` ....................................................... 37
- `searchforabnormalities.cols` ....................................................... 37
- `searchforabnormalities.rows` ...................................................... 38
- `setfshome` ................................................................................... 39
- `subjectDistributionTable` ............................................................ 39
rsurfer-package

Description

A package for interfacing between R and 'Freesurfer'

Details

This package provides functionality for importing MRI data which has been processed with 'Freesurfer' into R. It also provides functions for manipulating this data within R such as functionality to perform intracranial volume normalisation and field manipulation.

Author(s)

Maintainer: Alexander Luke Spedding <alexspedding71@gmail.com>

Authors:

• Fabio Cappello
• Giuseppe di Fatta

See Also

Useful links:

• Report bugs at https://github.com/AlexDiru/rsurfer/issues

addrandomdiagnosis

Description

This function will add a random diagnosis to every subject. It works by appending a Diagnosis column to the input data frame 'all'. Half of the diagnosis will be 'HC' (healthy control) and the other half will be 'AD' (Alzheimer's Disease).

Usage

addrandomdiagnosis(all)
Arguments

all The data frame to add a Diagnosis column to

Value

The input data frame with a Diagnosis column added

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

data <- generaterandomsubjects()
addrandomdiagnosis(data)

Description

This function will add a random gender to every subject. It works by appending a Gender column to the input data frame `all`. Half of the diagnosis will be Male and the other half will be Female.

Usage

addrandomgender(all)

Arguments

all The data frame to add a gender column to

Value

The input data frame with a Gender column added

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

data <- generaterandomsubjects()
addrandomgender(data)
adni.mergewithfreesurferoutput

ADNI Merge With 'Freesurfer' Output

Description

Merges the (baseline) Diagnosis, Age, Gender and MMSE for ADNI subjects with the post-processed 'Freesurfer' data, note that the rownames of the input data frame must be the Subject IDs i.e. 141_S_4232 of the ADNI database

Usage

adni.mergewithfreesurferoutput(df)

Arguments

df
     The data frame of data imported using fsimport()

Value

The input data frame merged with Age, Gender, Diagnosis and MMSE

Author(s)

Alexander Luke Spedding, <alexspeeding271@gmail.com>

Examples

## Not run:
setfshome("/Applications/freesurfer")
df <- fsimport.serialise("/~CADDementiaSubjects/", "~/CADSubjects.rds", verbose = T)
adni.setfiles("DXSUM_PDXCONV_ADNIALL.csv", "ADNIMERGE.csv")
df <- adni.mergewithfreesurferoutput(df)

## End(Not run)

adni.printfilelocations

ADNI Print File Locations

Description

Prints the location of two files required to merge the (baseline) Diagnosis, Age, Gender and MMSE for ADNI subjects with the post-processed 'Freesurfer' data
Usage

adni.printfilelocations()

Author(s)

Alexander Luke Spedding, <alexspeeding271@gmail.com>

Examples

adni.setfiles("DXSUM_PDXCONV_ADNIALL.csv", "ADNIMERGE.csv")
adni.printfilelocations()

Description

Points the location of two files required to merge the (baseline) Diagnosis, Age, Gender and MMSE for ADNI subjects with the post-processed 'Freesurfer' data

Usage

adni.setfiles(dxsumLocation, adnimergeLocation)

Arguments

dxsumLocation    The filepath to DXSUM_PDXCONV_ADNIALL.csv
adnimergeLocation    The filepath to ADNIMERGE.csv

Author(s)

Alexander Luke Spedding, <alexspeeding271@gmail.com>

Examples

adni.setfiles("DXSUM_PDXCONV_ADNIALL.csv", "ADNIMERGE.csv")
Description

Merges the Diagnosis, Age, and Gender for CAD Dementia subjects with the post-processed 'Freesurfer' data, note that the rownames of the input data frame must be the Subject IDs i.e. train_vumc_011 of the CAD Dementia data.

Usage

caddementia.mergewithfreesurferoutput(df)

Arguments

df
    The data frame of data imported using fsimport()

Details

Test data Diagnoses are returned as NAs

Value

The input data frame merged with Age, Gender and Diagnosis

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```r
## Not run:
setfshome("/Applications/freesurfer")
df <- fsimport.serialise("~/CADDementiaSubjects/", "~/CADSubjects.rds", verbose = T)
caddementia.setfiles("train.txt", "test.txt")
df <- caddementia.mergewithfreesurferoutput(df)

## End(Not run)
```
caddementia.printfilelocations

_CAD Dementia Print File Locations_

**Description**

Prints the location of two files required to merge the Diagnosis, Age, and Gender for CAD Dementia subjects with the post-processed 'Freesurfer' data

**Usage**

```r
caddementia.printfilelocations()
```

**Author(s)**

Alexander Luke Spedding, <alexspedding271@gmail.com>

**Examples**

```r
caddementia.setfiles("train.txt", "test.txt")
caddementia.printfilelocations()
```

caddementia.setfiles

_CAD Dementia Set Files_

**Description**

Points to the location of the two files required to merged the Diagnosis, Age and Gender for CAD Dementia subjects with the post-processed CAD Dementia data

**Usage**

```r
caddementia.setfiles(trainLocation, testLocation)
```

**Arguments**

- `trainLocation` - The filepath to train.txt
- `testLocation` - The filepath to test.txt

```r
caddementia.setfiles("train.txt","test.txt")
```

**Details**

Data can be accessed: [https://caddementia.grand-challenge.org/](https://caddementia.grand-challenge.org/)

**Author(s)**

Alexander Luke Spedding, <alexspedding271@gmail.com>
elminateabnormalities

Eliminate Abnormal Rows and Columns

Description
Will remove rows from a data frame of data generated by 'Freesurfer' that are abnormal - such as they have values of NA in them; will also remove columns.

Usage
eliminateabnormalities(data, verbose = F)

Arguments
data The data frame to remove abnormalities from
verbose Whether to print a log of what was removed

Value
The data frame with abnormalities removed

Author(s)
Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples
data <- generaterandomsubjects()
eliminateabnormalities.rows(data)

elminateabnormalities.cols

Eliminate Abnormal Columns

Description
Will remove columns from a data frame of data generated by 'Freesurfer' that are abnormal - columns where all values are zero.

Usage
eliminateabnormalities.cols(data, verbose = T)
eliminateabnormalities.rows

Arguments

- data: The data frame to remove abnormal columns from
- verbose: Whether to print a log of what was removed

Value

The data frame with abnormal columns removed

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```
data <- generaterandomsubjects() 
eliminateabnormalities.cols(data)
```

eliminateabnormalities.rows

`Eliminate Abnormal Rows`

Description

Will remove rows from a data frame of data generated by 'Freesurfer' that are abnormal - such as they have values of NA in them

Usage

```
eliminateabnormalities.rows(data, verbose = T)
```

Arguments

- data: The data frame to remove abnormal rows from
- verbose: Whether to print a log of what was removed

Value

The data frame with abnormal rows removed

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```
data <- generaterandomsubjects(10)  
eliminateabnormalities.rows(data)
```
Description

Exports the 'Freesurfer' imported data frame to a CSV readable by the software KNIME, it will assign two extra rows to the input dataframe: field_group_1 which classifies the columns as $S=$ hippocampal, subcortical, corticalthicknessstds, corticalareas, corticalthicknesses, corticalvolumes field_group_2 which classifies the columns as $S=$ volume, area, thickness, thicknessstd And the data for the columns will be in folder/field_group_*/$S.csv

Usage

```
export.forKNIME(df, folder, additionalFields = c("Gender", "Age", "Diagnosis"))
```

Arguments

- `df`: The data frame to export
- `folder`: The folder to export to
- `additionalFields`: Vector of column names which should be added to each individual file created

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```
## Not run:
export.forKNIME(df, "/Users/alex/KNIMEData/", c("Age","MMSE"))
## End(Not run)
```

---

### extract.brain.features

#### Extract All MRI Brain Features

Description

This command takes a data frame as input and extracts all the features segmented by 'Freesurfer'. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.
extract.bname

Usage

extract.bname(data, additionalFields = c())

Arguments

data: The subject data to extract from
additionalFields: Any additional fields to extract data from

Value

The MRI brain features

Author(s)

Fabio Cappello

Examples

data <- generaterandomsubjects()
extract.bname(data)

extract.bname Extract Features By Group Name

Description

Extracts features in group specified by a string, groups that can be used are in normalise.listfieldgroups()

Usage

extract.bname(data, fieldGroupName, additionalFields = c())

Arguments

data: Subject data to extract from
fieldGroupName: The group field name to extract
additionalFields: Any additional fields to extract

Value

The extracted fields

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>
**Examples**

```r
data <- generatorandomsubjects()
extract.bynam(data, "subcortical")
```

**extract.cortical**  
**Extract Cortical Fields**

**Description**

This command takes a data frame as input and extracts all the cortical fields (cortical volumes, cortical thicknesses, cortical surface areas, standard deviations of cortical thicknesses) from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

**Usage**

```r
extract.cortical(data, additionalFields = c())
```

**Arguments**

- `data`  
The subject data to extract from
- `additionalFields`  
  Any additional fields to extract data from

**Value**

The cortical fields

**Author(s)**

Alexander Luke Spedding, <alexspeeding271@gmail.com>

**Examples**

```r
data <- generatorandomsubjects()
extract.cortical(data)
```
**extract.corticalthicknesses**

---

**extract.corticalsurfaceareas**

*Extract Cortical Surface Areas*

**Description**

This command takes a data frame as input and extracts all the cortical surface areas from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

**Usage**

```r
extract.corticalsurfaceareas(data, additionalFields = c())
```

**Arguments**

- `data` The subject data to extract from
- `additionalFields` Any additional fields to extract data from

**Value**

The cortical surface areas

**Author(s)**

Alexander Luke Spedding, <alexspedding271@gmail.com>

**Examples**

```r
data <- generaterandomsubjects()
exttract.corticalsurfaceareas(data)
```

---

**extract.corticalthicknesses**

*Extract Cortical Thicknesses*

**Description**

This command takes a data frame as input and extracts all the cortical thicknesses from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

**Usage**

```r
extract.corticalthicknesses(data, additionalFields = c())
```
extract.corticalthicknessstddevs

Arguments

data The subject data to extract from
additionalFields Any additional fields to extract data from

Value

The cortical thicknesses

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

data <- generaterrandomsubjects()
extract.corticalthicknesses(data)

extract.corticalthicknessstddevs

Extract Cortical Thickness Standard Deviations

Description

This command takes a data frame as input and extracts all the cortical thickness standard deviations from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

Usage

extract.corticalthicknessstddevs(data, additionalFields = c())

Arguments

data The subject data to extract from
additionalFields Any additional fields to extract data from

Value

The cortical thickness standard deviations

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>
extract.corticalvolumes

Extract Cortical Volumes

Description

This command takes a data frame as input and extracts all the cortical volumes from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

Usage

extract.corticalvolumes(data, additionalFields = c())

Arguments

data               The subject data to extract from
additionalFields   Any additional fields to extract data from

Value

The cortical volumes

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

data <- generaterandomsubjects()
extract.corticalvolumes(data)
extract.hippocampalvolumes

*Extract Hippocampal Volumes*

**Description**

This command takes a data frame as input and extracts all the hippocampal volumes from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

**Usage**

```r
extract.hippocampalvolumes(data, additionalFields = c())
```

**Arguments**

- `data`: The subject data to extract from
- `additionalFields`: Any additional fields to extract data from

**Value**

The hippocampal volumes

**Author(s)**

Alexander Luke Spedding, <alexspeeding271@gmail.com>

**Examples**

```r
data <- generateRandomSubjects()
extraict.hippocampalvolumes(data)
```

---

extract.subcorticalvolumes

*Extract Subcortical Volumes*

**Description**

This command takes a data frame as input and extracts all the subcortical volumes from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

**Usage**

```r
extract.subcorticalvolumes(data, additionalFields = c())
```
extract.volumes

Arguments

  data  The subject data to extract from

  additionalFields  Any additional fields to extract data from

Value

  The subcortical volumes

Author(s)

  Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

  data <- generatereandomsubjects()
  extract.subcorticalvolumes(data)

Description

  This command takes a data frame as input and extracts all the cortical thicknesses from this data frame and any other fields specified by the user. Note that the MRIs must be processed with the 'hippocampal-subfields' flag when 'Freesurfer' is invoked.

Usage

  extract.volumes(data, additionalFields = c())

Arguments

  data  The subject data to extract from

  additionalFields  Any additional fields to extract data from

Value

  The cortical thicknesses

Author(s)

  Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

  data <- generatereandomsubjects()
  extract.corticalthicknesses(data)
fsdirectorycheck

`Freesurfer' Subject Directory Check

Description

Crawls through the subdirectory of subjects looking for missing file that would affect the fsimport() process.

Usage

```r
fsdirectorycheck(subjectDir, checkForHippocampalSubfieldsError = T)
```

Arguments

- `subjectDir` The directory containing the subject subdirectories
- `checkForHippocampalSubfieldsError` Whether to check to missing files which are to do with the "hippo-subfields" flag being called

Value

The number of errors found

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

fsimport

`Freesurfer’ Import (Serialise)

Description

This function assumes all the subdirectories of subjectDir are subjects who have been processed in 'Freesurfer'. In then runs various 'Freesurfer’ scripts to extract the specified fields into an R data frame.

Usage

```r
fsimport(subjectDir, fields = fsimport.listfields(), verbose = F)
```

Arguments

- `subjectDir` The directory containing the subject subdirectories
- `fields` The field groups to use, see fsimport() for more details
- `verbose` Whether to log the 'Freesurfer’ scripts to the R console
Details

The field groups which can be imported from the subject are: The specified fields can be:

lh.cortical.thickness = left hemisphere cortical thicknesses
rh.cortical.thickness = right hemisphere cortical thicknesses
lh.cortical.volume = left hemisphere cortical volumes
rh.cortical.volume = right hemisphere cortical volumes
lh.cortical.thickness.std = left hemisphere cortical thickness standard deviations
rh.cortical.thickness.std = right hemisphere cortical thickness standard deviations
lh.cortical.area = left hemisphere cortical surface areas
rh.cortical.area = right hemisphere cortical surface areas
lh.subcortical = left hemisphere subcortical volumes
rh.subcortical = right hemisphere subcortical volumes
lh.hippocampal = left hemisphere hippocampal volumes
rh.hippocampal = right hemisphere hippocampal volumes

By default all of the above fields are included. For the hippocampal volumes, the subjects must have been processed with the "hippo-subfields" when 'Freesurfer' was invoked.

Value

The subject data processed from 'Freesurfer'

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```r
## Not run:
setfshome("/Applications/freesurfer")
fsimport("/Users/alex/Desktop/Subjects")
## End(Not run)
```

Description

This function assumes all the subdirectories of subjectDir are subjects who have been processed in 'Freesurfer'. In then runs various 'Freesurfer' scripts to extract the specified fields into an R data frame.

Usage

```r
fsimport.listfields()
```
Details

Lists the field groups which can be imported from the subject are: The specified fields can be:

- lh.cortical.thickness = left hemisphere cortical thicknesses
- rh.cortical.thickness = right hemisphere cortical thicknesses
- lh.cortical.volume = left hemisphere cortical volumes
- rh.cortical.volume = right hemisphere cortical volumes
- lh.cortical.thickness.std = left hemisphere cortical thickness standard deviations
- rh.cortical.thickness.std = right hemisphere cortical thickness standard deviations
- lh.cortical.area = left hemisphere cortical surface areas
- rh.cortical.area = right hemisphere cortical surface areas
- lh.subcortical = left hemisphere subcortical volumes
- rh.subcortical = right hemisphere subcortical volumes
- lh.hippocampal = left hemisphere hippocampal volumes
- rh.hippocampal = right hemisphere hippocampal volumes

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

fsimport.listfields()

---

**Description**

Calls `fsimport()` and once that data frame is created it is serialised to the file specified by `serialisationLocation`. However, if this file already exists when the function is run, then it will unserialise the file instead of calling `fsimport()`. This saves constant running of `Freesurfer` scripts when running code thus saving execution time for code to run.

**Usage**

```r
fsimport.serialise(subjectDir, serialisationLocation, fields, verbose)
```

**Arguments**

- `subjectDir`: The directory containing the subject subdirectories
- `serialisationLocation`: The location where the serialised file is saved to and loaded from
- `fields`: The field groups to use, see `fsimport()` for more details
- `verbose`: Whether to log the `Freesurfer` scripts to the R console
generaterandomsubjects

Value
The subject data processed from 'Freesurfer'

Author(s)
Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples
```
# Not run:
setfshome("/Applications/freesurfer")
fsimport.serialise("~/Subjects", serialisationLocation = "~/data.rds")
```
```
# End(Not run)
```

---

**Description**

Generate a data frame of random subjects whose fields match what would be imported from 'Freesurfer'. This function is used for testing.

**Usage**

`generaterandomsubjects(num = 40)`

**Arguments**

- `num` The number of subjects to generate

**Value**

The generated subjects

**Author(s)**

Alexander Luke Spedding, <alexspedding271@gmail.com>

**Examples**

`generaterandomsubjects(num = 500)`
get.hemisphere.side  Get Hemisphere Side

Description

Given the name of a feature, will return a string as to whether it belongs to the left or the right hemisphere. If it belongs to neither, it is assumed that the feature is central.

Usage

get.hemisphere.side(name)

Arguments

name The name of the feature to return the hemisphere of

Value

The side of the hemisphere the feature belongs to ("left" or "right"). If it belongs to neither of these, "central" is returned.

Author(s)

Alexander Luke Spedding, <alexspeeding271@gmail.com>

Examples

get.hemisphere.side("Right.vessel")
get.hemisphere.side("lhCortexVol")

get.opposite.hemisphere.measurement

Get Opposite Hemisphere Measurement

Description

Given a left hemisphere measurement, will return the equivalent measure on the right hemisphere. If there is no equivalent feature, NULL will be returned.

Usage

get.opposite.hemisphere.measurement(name, verbose = F, verbose_warn = F)
Arguments

name The name of the feature to return the corresponding measurement of the opposite hemisphere of
verbose If a corresponding feature doesn’t exist, a message will be printed if this is true
verbose_warn If a corresponding feature doesn’t exist, a message will be printed as a warning if this is true

Value

Name of the feature on the other hemisphere (or NULL if the feature does not exist)

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

g.get.opposite.hemisphere.measurement("Right.vessel")
g.get.opposite.hemisphere.measurement("1hCortexVol")

---

getfieldgroup Get Field Group of Feature

Description

Given the name of a feature, this function gets what type of field group it belongs to, i.e. subcortical volume

Usage

g.getfieldgroup(name, method = 1)

Arguments

name Name of the feature
method The type of field groups that are returned method = 1: hippocampal, subcortical, corticalthicknessstuds, corticalareas, corticalthicknesses, corticalvolumes method = 2: volume, area, thickness, thicknessstd

Value

The field group the name belongs to

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>
getfshome

Examples

gffieldgroup("left.CA1", method = 1)
gffieldgroup("left.CA1", method = 2)

Description
This command is used to get the base directory where 'Freesurfer' is installed as set using the command setfshome().

Usage

getfshome()

Value
The directory 'Freesurfer' is installed

Author(s)
Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

sfsthome("/Applications/freesurfer/")
getfshome()

getfsversion

Get 'Freesurfer' Version

Description
This command is used to get the version number of 'Freesurfer' which a certain subject was processed with

Usage

getfsversion(subjectDir)

Arguments

subjectDir The directory of the subject to get the 'Freesurfer' version of
getnamesofareas

Value

The version number

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

## Not run:
getfsversion("/Users/alex/Subjects/002_S_0413/")

## End(Not run)

getnamesofareas Get Names of Areas

Description

Gets the names of all the features that are areas

Usage

getnamesofareas(data = NULL, excludeFields = NULL)

Arguments

data Your subject data frame, if you have removed any columns from your data frame, the function will only return the areas in this data frame. If this parameter is NULL then a new data frame will be randomly generated

excludeFields A vector of areas names to exclude, to exclude nothing set this parameter to NULL

Value

A vector of the names of all the features which are areas

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

gGetnamesofareas()
gGetnamesofareas(NULL, NULL)
getnamesofthicknesses  

Get Names of Thicknesses

Description

Gets the names of all the features that are thicknesses

Usage

getnamesofthicknesses(data = NULL, excludeFields = NULL)

Arguments

data  Your subject data frame, if you have removed any columns from your data frame, the function will only return the thicknesses in this data frame. If this parameter is NULL then a new data frame will be randomly generated

excludeFields  A vector of thickness names to exclude, to exclude nothing set this parameter to NULL

Value

A vector of the names of all the features which are thicknesses

Author(s)

Alexander Luke Spedding, <alexspeeding271@gmail.com>

Examples

getnamesofthicknesses()
getnamesofthicknesses(NULL, NULL)

getnamesofvolumes  

Get Names of Volumes

Description

Gets the names of all the features that are volumes

Usage

getnamesofvolumes(data = NULL, excludeFields = c("EstimatedTotalIntraCranialVol"))
Arguments

data: Your subject data frame, if you have removed any columns from your data frame, the function will only return the volumes in this data frame. If this parameter is NULL then a new data frame will be randomly generated.

excludeFields: A vector of volume names to exclude, to exclude nothing set this parameter to NULL.

Value

A vector of the names of all the features which are volumes.

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

genamesofvolumes()
genamesofvolumes(NULL, NULL)
genamesofvolumes(NULL, excludeFields = c("left.CA1", "EstimatedTotalIntraCranialVol"))

is.cortical (Is Feature Cortical)

Description

Checks whether a feature is a cortical measurement.

Usage

is.cortical(fieldName)

Arguments

fieldName: The field name of the feature to check is a cortical measurement.

Value

Whether the feature is cortical.

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

is.cortical("lh.bankssts.thickness")
is.corticalarea  

**Is Feature Cortical Area**

**Description**
Checks whether a feature is a cortical area

**Usage**
`is.corticalarea(field_name)`

**Arguments**
- **fieldName**  
The field name of the feature to check is a cortical area

**Value**
Whether the feature is a cortical area

**Author(s)**
Alexander Luke Spedding, <alexspedding271@gmail.com>

**Examples**
`is.corticalarea("lh.bankssts.area")`

---

is.corticalthickness  

**Is Feature Cortical Thickness**

**Description**
Checks whether a feature is a cortical thickness

**Usage**
`is.corticalthickness(field_name)`

**Arguments**
- **fieldName**  
The field name of the feature to check is a cortical thickness

**Value**
Whether the feature is a cortical thickness
is.corticalthicknessstd

Is Feature Cortical Thickness Standard Deviation

Description
Checks whether a feature is a cortical thickness standard deviation

Usage
is.corticalthicknessstd(fieldName)

Arguments
fieldName The field name of the feature to check is a cortical thickness standard deviation

Value
Whether the feature is a cortical thickness standard deviation

Author(s)
Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples
is.corticalthicknessstd("lh.bankssts.thicknessstd")
is.corticalvolume  

Description
Checks whether a feature is a cortical volume

Usage
is.corticalvolume(fieldNname)

Arguments

fieldName  
The field name of the feature to check is a cortical volume

Value

Whether the feature is a cortical volume

Author(s)
Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

is.corticalvolume("lh.bankssts.volume")

---

is.hippocampalvolume  

Description
Checks whether a feature is a hippocampal volume

Usage
is.hippocampalvolume(fieldNname)

Arguments

fieldName  
The field name of the feature to check is a hippocampal volume

Value

Whether the feature is a hippocampal volume
is.subcorticalvolume

**Author(s)**

Alexander Luke Spedding, <alexspeedding271@gmail.com>

**Examples**

```r
is.hippocampalvolume("right.fimbria")
```

---

| **is.subcorticalvolume** | *Is Feature Subcortical Volume* |

**Description**

Checks whether a feature is a subcortical volume

**Usage**

```r
is.subcorticalvolume(fieldNName)
```

**Arguments**

- `fieldName` The field name of the feature to check is a subcortical volume

**Value**

Whether the feature is a subcortical volume

**Author(s)**

Alexander Luke Spedding, <alexspeedding271@gmail.com>

**Examples**

```r
is.subcorticalvolume("Brain.Stem")
```
isfsfeature  

*Is Feature Created by 'Freesurfer'*

**Description**  
Given the name of a feature, this function returns whether it was a feature generated by 'Freesurfer' processing stream.

**Usage**  
`isfsfeature(name)`

**Arguments**  
- `name`  
  Name of the feature

**Value**  
Whether the feature was created by 'Freesurfer'

**Author(s)**  
Alexander Luke Spedding, <alexspedding271@gmail.com>

**Examples**  
`isfsfeature("left.CA1")`
`isfsfeature("Age")`

ixi.mergewithfreesurferoutput  

*'IXI’ Merge Study Information With 'Freesurfer’ Output*

**Description**  
Merges the external 'IXI’ study data (specified with 'IXI’ Set Files) with the data generated by 'Freesurfer'.

**Usage**  
`ixi.mergewithfreesurferoutput(ixi.mri.data, verbose = T)`

**Arguments**  
- `ixi.mri.data`  
  The data frame containing the data generated by 'Freesurfer' of the 'IXI' subjects
- `verbose`  
  Whether to log information to the console
Details

Data can be accessed: http://brain-development.org/ixi-dataset/
Requires a modification of IXI.xls

Author(s)

Fabio Cappello, Alexander Luke Spedding, <alexspedding271@gmail.com>

---

ixi.setfile | IXI Set File

Description

Points to the location of the two files required to merge the Diagnosis, Age and Gender for IXI subjects with the post-processed IXI data

Usage

ixi.setfile(location)

Arguments

location | The filepath to IXI.csv

---

normalise | Normalise

Description

Performs ICV (intracranial volume) normalisation on a data frame of imported subjects in data.

Usage

normalise(data, normalisationFunction, fieldType = "all", trainData = NULL, testData = NULL)
normalise  

Arguments  

- **data**  
  The subject data to normalise
- **normalisationFunction**  
  The normalisation function to use, see details for further information.
- **fieldType**  
  The field set to normalise, see details for further information.
- **trainData**  
  Data to train on, required for 'hconly' normalisation methods
- **testData**  
  Unseen data, required for 'hconly' normalisation methods

Details  

Performs ICV (intracranial volume) normalisation on a data frame of imported subjects in `data`. The `normalisationFunction` specifies which normalisation method to use:

- **normalisation.proportional** = proportional ICV normalisation, the volumes of each subject are divided by their ICV
- **normalisation.residual** = residual ICV normalisation, a linear regression model is built for each volume using the ICV as a predictor
- **normalisation.residualgender** = residual ICV normalisation with a gender split, similar to residual ICV normalisation, except a separate linear regression model is built for Males and Females
- **normalisation.residualhconly** = residual ICV normalisation creating a regression model based on healthy control patients only

The **fieldType** can be:

- **corticalvolumes** = Normalise cortical volumes by ICV
- **subcortical** = Normalise subcortical volumes by ICV
- **hippocampal** = Normalise hippocampal volumes by ICV
- **corticalareas** = Normalise cortical areas by ICV
- **corticalthicknesses** = Normalise cortical thicknesses by ICV
- **corticalthicknessesstds** = Normalise cortical thicknesses standard deviations by ICV
- **corticalareastsa** = Normalise cortical areas by total surface area
- **corticalthicknessessmct** = Normalise cortical thicknesses by mean cortical thickness

Value  

The normalised data

Author(s)  

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples  

```r  
data <- generateRandomSubjects()  
addRandomGender(data)  ```
normalise.listfieldgroups  
*List Normalisation Field Groups*

**Description**
Lists all the available field groups that the normalisation can operate on

**Usage**
`normalise.listfieldgroups()`

**Value**
A list of the normalisation field groups

**Author(s)**
Alexander Luke Spedding, <alexspedding271@gmail.com>

**Examples**
`normalise.listfieldgroups()`

---

normalise.listmethods  
*List Normalisation Methods*

**Description**
Lists all the available normalisation methods

**Usage**
`normalise.listmethods()`

**Value**
A list of the normalisation methods

**Author(s)**
Alexander Luke Spedding, <alexspedding271@gmail.com>

**Examples**
`normalise.listmethods()`
removeabnormalrowsandcols

 Remove Abnormalities

Description

Removes columns and rows which have been exported from 'Freesurfer' and may cause classification problems

Usage

removeabnormalrowsandcols(df, verbose)

Arguments

df The data frame imported using fsimport
verbose Whether the print debug information

Value

The data frame with abnormal rows and columns removed

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

data <- generaterandomsubjects()
data <- removeabnormalrowsandcols(data, T)

searchforabnormalities.cols

 Search For Abnormalities (Columns)

Description

Looks for columns which have been exported from 'Freesurfer' and may cause classification problems, for example, my personal abnormal columns are Left.WM.hypointensities, Right.WM.hypointensities, Left.non.WM.hypointensities, Right.WM.hypointensities only have values of zero

Usage

searchforabnormalities.cols(data, verbose = T)
Arguments

data          The data frame imported using fsimport
verbose      Whether the print debug information

Value

Indices of abnormal columns

Author(s)

Alexander Luke Spedding, <alexspeeding271@gmail.com>

Examples

data <- generatorandomsubjects()
searchforabnormalities.cols(data)

searchforabnormalities.rows

Search For Abnormalities (Rows)

Description

Looks for rows (subjects) which have been exported from ‘Freesurfer’ and may cause classification problems, for example, rows with NAs in

Usage

searchforabnormalities.rows(data, verbose = T)

Arguments

data          The data frame imported using fsimport
verbose      Whether the print debug information

Value

Indices of abnormal rows

Author(s)

Alexander Luke Spedding, <alexspeeding271@gmail.com>

Examples

data <- generatorandomsubjects()
searchforabnormalities.rows(data)
setfshome  

**Set 'Freesurfer' Home**

**Description**

This command is used to set the base directory where 'Freesurfer' is installed. My installation of 'Freesurfer' is located in: "~/Applications/freesurfer"; thus for my installation location setfshome("~/Applications/freesurfer") would be used.

**Usage**

```r
setfshome(freesurferDirectory)
```

**Arguments**

- `freesurferDirectory`
  
  The directory 'Freesurfer' is installed to

**Value**

None

**Author(s)**

Alexander Luke Spedding, <alexspeeding271@gmail.com>

**Examples**

```r
setfshome("~/Applications/freesurfer")
```

---

**subjectDistributionTable**  

*Create Subject Distribution Table*

**Description**

Creates a data frame with the distributions of the subjects age and gender grouped by the field 'targetClassName', providing an overview of the subjects. Note: requires an 'Age' and 'Gender' column.

**Usage**

```r
subjectDistributionTable(data, targetClassName)
```
Arguments

- `data`: The subject data to create the table from
- `targetClassName`: The name of the field to group the data by

Value

The subject distribution table in a data frame

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```r
all <- generateRandomSubjects(1000)
all$Age <- stats::runif(1000, 50, 80)
all <- addRandomGender(all)
all <- addRandomDiagnosis(all)
subjectDistributionTable(all, "Diagnosis")
```

---

**subjectDistributionTableToLatex**

_Convert Subject Distribution Table To LaTeX_

Description

Converts a subject distribution table created using `subjectDistributionTable()` into text which can be used in the typesetting language LaTeX. The table created can have its caption and label specified using the respective function arguments. The decimal point rounding can be specified by the function argument `roundDP`.

Usage

```r
subjectDistributionTableToLatex(subjectDistributionTable, 
caption = "Placeholder Caption", label = "table:SubjectDistributionTable", 
roundDP = 1)
```

Arguments

- `subjectDistributionTable`: The subject distribution table created using `subjectDistributionTable()`
- `caption`: The caption to give the table in LaTeX
- `label`: The label to give the table in LaTeX
- `roundDP`: The number of decimal places to round the numbers to on the table
Value

The LaTeX code representing the subject distribution table

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```r
call <- generaterandomsubjects(1000)
call$Age <- stats::runif(1000, 50, 80)
call <- addrandomgender(all)
call <- addrandomdiagnosis(all)
sdt <- subjectDistributionTable(all, "Diagnosis")
subjectDistributionTableToLatex(subjectDistributionTable = sdt,
    caption="Subject Distribution Table",
    label="table:SDT", roundDP=1)
```

Description

Tests the field extraction functions are working correctly on imported data from subjectDir. Note: requires the subjects to be processed with the "hippocampal-subfields" flag.

Usage

```r
test.fieldextraction(subjectTestDir)
```

Arguments

- `subjectTestDir` The directory containing the subject subdirectories

Author(s)

Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

```r
## Not run:
test.fieldextraction("/Users/alex/Desktop/Subjects")

## End(Not run)```
test.importing  Test Importing

Description
Calls fsimport() with different parameters on the subjects in subjectDir to test it is working correctly. Note: requires the subjects to be processed with the "hippocampal-subfields" flag.

Usage

test.importing(subjectTestDir)

Arguments

subjectTestDir  The directory containing the subject subdirectories

Author(s)
Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

## Not run:
test.importing("/Users/alex/Desktop/Subjects")

## End(Not run)

test.normalisation  Test Normalisation

Description
Tests the normalisation functions are working correctly on randomly generated data.

Usage

test.normalisation()

Author(s)
Alexander Luke Spedding, <alexspedding271@gmail.com>

Examples

## Not run:
test.normalisation()

## End(Not run)
Index

addrandomdiagnosis, 3
addrandomgender, 4
adni.mergewithfreesurferoutput, 5
adni.printfilelocations, 5
adni.setfiles, 6
caddementia.mergewithfreesurferoutput, 7
caddementia.printfilelocations, 8
caddementia.setfiles, 8
eliminateabnormalities, 9
eliminateabnormalities.cols, 9
eliminateabnormalities.rows, 10
export.forKNIME, 11
extract.brain.features, 11
extract.byname, 12
extract.cortical, 13
extract.corticalsurfaceareas, 14
extract.corticalthicknesses, 14
extract.corticalthicknessstddevs, 15
extract.corticalvolumes, 16
extract.hippocampalvolumes, 17
extract.subcorticalvolumes, 17
extract.volumes, 18
fsdirectorycheck, 19
fsimport, 19
fsimport.listfields, 20
fsimport.serialise, 21
generaterandomsubjects, 22
generate.hemisphere.side, 23
generate.opposite.hemisphere.measurement, 23
generatefieldgroup, 24
gengetfshome, 25
getfsversion, 25
getnamesofareas, 26
getnamesofthicknesses, 27
getnamesofvolumes, 27
is.cortical, 28
is.corticalarea, 29
is.corticalthickness, 29
is.corticalthicknessstd, 30
is.corticalvolume, 31
is.hippocampalvolume, 31
is.subcorticalvolume, 32
isfsfeature, 33
ixi.mergewithfreesurferoutput, 33
ixi.setfile, 34
normalise, 34
normalise.listfieldgroups, 36
normalise.listmethods, 36
removeabnormalrowsandcols, 37
rsurfer (rsurfer-package), 3
rsurfer-package, 3
searchforabnormalities.cols, 37
searchforabnormalities.rows, 38
setfshome, 39
subjectDistributionTable, 39
subjectDistributionTableToLatex, 40
test.fieldextraction, 41
test.importing, 42
test.normalisation, 42
fsimport.listmethods