

Package ‘neuroSCC’

June 20, 2024

Type Package

Title Estimation of SCC's for PET Neuroimaging Data

Version 0.2.1

Author Juan A. Arias [aut, cre]

Maintainer Juan A. Arias <iguanamarina@protonmail.com>

Description This package provides functions for the estimation of Simultaneous Confidence Corridors (SCCs) on PET neuroimaging data. It is designed to help with the replication of methods implemented in the Ph.D. Thesis ``Development of statistical methods for neuroimaging data analysis towards early diagnostic of neurodegenerative diseases" and it is best applied using the scripts available in the link below. Further information on the methodology for SCCs estimation can be found in: ``Wang Y, Wang G, Wang L, Ogden RT. Simultaneous confidence corridors for mean functions in functional data analysis of imaging data. Biometrics. 2020 Jun;76(2):427-437".

License use_mit_license()

URL <https://github.com/iguanamarina/neuroSCC>

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Suggests roxygen2 (>= 7.1.0),
knitr,
rmarkdown

Roxygen list(markdown = TRUE)

Depends R (>= 2.10)

R topics documented:

getDimensions	2
neuroCleaner	3
PETsample	4

Index	5
-------	---

`getDimensions`*Get Dimensions from a DICOM File*

Description

This function loads a DICOM image file using the `oro.nifti` package and extracts its dimensions. It provides the X and Y dimensions as well as the total number of elements ('dim') in the image data.

Usage

```
getDimensions(filename = NULL)
```

Arguments

<code>filename</code>	character, optional; the name of the DICOM file to read. If not provided, the function will search for the first <code>.img</code> file in the current working directory.
-----------------------	---

Details

If no filename is provided, the function searches the current directory for the first file with a `.img` extension. It stops with an error message if no such files are found. It is important to ensure that the specified file or files in the directory are in the DICOM format.

Value

A list containing `xDim`, `yDim`, and `dim`, representing the dimensions of the image in the X and Y axes, and the total number of elements in the image, respectively.

See Also

[readNIfTI](#) for the function used to read the DICOM files.

Examples

```
# If 'filename' is not provided, it will get the first image in the current working directory:
dimensions <- getDimensions()

# Providing a specific filename:
dimensions <- getDimensions("003_S_1059.img")
```

neuroCleaner*Cleans and loads data from NIFTI files*

Description

This function reads a NIFTI image, transforms it into a dataframe, preserves the cross-section specified by the Z axis, and organizes the data into a structured table that other functions work on subsequently.

Usage

```
neuroCleaner(name, demo)
```

Arguments

name	character, the name of the NIFTI file to read.
demo	data.frame, a dataframe containing demographic data formatted according to the demoCleaner function.

Details

The function first reads the NIFTI file using the `oro.nifti::readNIFTI` function from the package. Then, it converts the image data to a dataframe and selects only the cross-section of interest. Afterward, it checks if the demographic dataframe contains the necessary columns and extracts the data for the specified participant. Finally, it combines these data with the image data and returns the resulting dataframe for that patient.

Value

A `data.frame` that combines the NIFTI image data with the demographic data in the appropriate format. Each row represents a pixel, and the columns include demographic data and pixel intensity.

Author(s)

Juan A. Arias (<http://juan-arias.xyz>)

See Also

[demoCleaner](#) and `oro.nifti::readNIFTI()`.

PETsample

Data from DICOM Image File

Description

The 'PETsample' dataset contains the data extracted from the DICOM image "003_S_1059.img".

Usage

```
data("PETsample")
```

Format

A PET neuroimaging file with dimensions and other metadata stored in the ".hdr" file.

Source

This file has been obtained with permission from ADNI.

Index

* **datasets**

PETsample, [4](#)

demoCleaner, [3](#)

getDimensions, [2](#)

neuroCleaner, [3](#)

PETsample, [4](#)

readNIfTI, [2](#)