Package 'neuroSCC'

July 3, 2024

Type Package

Title Estimation of SCC's for PET Neuroimaging Data

Version 0.9.0

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 neuroimage 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:

`databaseCreator` *Create a database of processed PET image data with the appropiate format*

Description

This function automates the processing of PET images based on a specified file name pattern within a working directory. It reads each file matching the pattern, processes it using `neuroSCC::neuroCleaner`, and compiles the results into a comprehensive data frame.

Usage

```
databaseCreator(pattern)
```

Arguments

`pattern`	`character`, a regular expression pattern that specifies which files to process.

Details

The function first checks if there are files matching the pattern in the working directory. If no files are found, it throws an error. Otherwise, it processes each file individually, extracts necessary data, and appends it to a growing database. The function leverages `neuroSCC::neuroCleaner` for data processing. Each file's subject number is extracted from its name using regular expressions.

Value

A `data.frame` that aggregates processed data from each image. Each row represents data from one image, including subject numbers and image data.

See Also

neuroCleaner for the underlying image processing.

Examples

```
# Set the working directory where your PET images are stored
setwd("~/GitHub/PhD-2023-Neuroimage-article-SCC-vs-SPM/PETimg_masked for simulations")

# Define the pattern for file names to process
pattern <- "^masked_swwwC\\d+_tripleNormEsp_w00_rrec_OSEM3D_32_it1.nii"

# Create the database
database_CN <- databaseCreator(pattern)
```

`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

`filename`	`character`, optional; the name of the DICOM file to read. If not provided, the function will search for the first `.img` 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")
```

`getPoints` *Extract standalone significant SCC points from a SCC comparison object*

Description

This function processes an SCC comparison object to identify and extract coordinates of points where significant differences are detected by the SCC analysis. It returns the coordinates of positive and negative differences separately, which can be used for further analysis or visualization on neuroimages.

Usage

```
getPoints(aa)
```

Arguments

`aa`	A list containing SCC comparison results, expected to have elements like `Z.band`, `ind.inside.cover`, and `scc` matrices.

Value

A list with two elements: coordinates of points with positive differences and coordinates of points with negative differences. Each element is a matrix where rows are points and columns represent coordinates.

Examples

```
points <- getPoints(SCC_COMP_1)
```

`matrixCreator` *Convert PET image data to a functional data matrix format*

Description

This function transforms PET image data, previously processed into a database format by `databaseCreator`, into a matrix format suitable for functional data analysis. Each row of the matrix represents a function, corresponding to data from one patient or control subject. The function recognizes whether the matrix is for Control or Alzheimer's group.

Usage

```
matrixCreator(database, pattern, param.z, xy)
```

Arguments

database	A data frame containing the PET image data with columns for 'CN_number', 'z', 'x', 'y', and 'pet' values, created by databaseCreator.
pattern	The regular expression pattern to match filenames in the database. This pattern should correspond to the naming conventions of the PET image files you want to process.
param.z	The specific z-coordinate slice to analyze. This parameter should be defined prior to running this function if not included in the script that calls it.
xy	The total number of data points per z-slice, calculated as the product of the dimensions x and y of the image slice. If not predefined, it must be calculated by the user based on the image dimensions.

Details

matrixCreator follows databaseCreator in the data processing workflow. It reads the data for each file specified by the matched pattern and extracts only the data for the specified z-slice. The function ensures that each row in the resulting matrix corresponds to one patient or control subject, transforming each subset of data into a format suitable for functional data analysis.

Value

A matrix where each row represents the PET data from one control subject, formatted as a continuous line of data points to simulate a function.

See Also

databaseCreator and neuroCleaner for initial data processing and image cleaning.

Examples

```
# Assuming 'database_CN', 'pattern', 'param.z', and 'xy' are already defined
SCC_CN <- matrixCreator(database_CN, pattern, param.z, xy)
```

---

meanNormalization *Mean Average Normalization for SCC Data*

Description

This function normalizes each row of the given SCC data by its mean value. It divides each element in a row by the mean of that row, ignoring NA values.

Usage

```
meanNormalization(SCC_data)
```

Arguments

SCC_data	matrix, a matrix where each row represents SCC data to be normalized.

Details

The function iterates over each row of the SCC data matrix, calculates the mean of the row, and then divides each element of the row by the calculated mean.

Value

A matrix where each row has been normalized by its mean value.

Examples

```
# Assume SCC_data is a matrix of SCC values
normalized_SCC_data <- neuroSCC::meanNormalization(SCC_data)
```

---

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 = NULL)
```

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. If not provided, only image data will be processed.

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. If the demographic dataframe is provided, it checks if it 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. If demo is not provided, it returns only the image data.

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. If `demo` is not provided, the dataframe will contain only the image data.

Author(s)

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

See Also

demoCleaner and `oro.nifti::readNIfTI()`.

`neuroContour` *Obtain and save neuroimaging contours*

Description

This function extracts contours from neuroimaging data where values change according to specified levels. It uses the `contoureR::getContourLines` function to obtain the contours and stores the coordinates in a list.

Usage

```
neuroContour(data, levels = c(0))
```

Arguments

`data`	`data.frame`, a data frame containing the neuroimaging data to process.
`levels`	`numeric`, a vector of levels at which to draw the contours. Default is `c(0)`.

Details

The function filters the contours by their `GID` and stores the coordinates of each contour in a list. It ensures that the `contoureR` package is loaded before attempting to use its functions.

Value

A `list` of data frames, where each data frame contains the x and y coordinates of a contour.

See Also

getContourLines for the underlying contour extraction.

Examples

```
# Example usage:
# Load sample data
data <- some_neuroimaging_data

# Get contours at level 0
contours <- neuroContour(data, levels = c(0))

# Plot the first contour
plot(contours[[1]])
if (length(contours) > 1) {
  for (j in 2:length(contours)) {
    points(contours[[j]]) # Holes or internal contours
  }
}
```

Index