Package ‘mand’

July 22, 2025

Type Package

Title Multivariate Analysis for Neuroimaging Data

Version 2.0

Date 2023-09-14

Maintainer Atsushi Kawaguchi <kawa_a24@yahoo.co.jp>

Depends R (>= 3.5), msma

Imports oro.nifti, oro.dicom, imager, caret

Description Several functions can be used to analyze neuroimaging data using multivariate methods based on the 'msma' package. The functions used in the book entitled ``Multivariate Analysis for Neuroimaging Data'' (2021, ISBN-13: 978-0367255329) are contained.

License GPL (>= 2)

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.3

NeedsCompilation no

Author Atsushi Kawaguchi [aut, cre]

Repository CRAN

Date/Publication 2023-09-12 22:10:06 UTC

Contents

`mand-package` *Multivariate Analysis for Neuroimaging Data Package*

Description

A Package for implementation of multivariate data analysis for neuroimaging data.

Author(s)

Atsushi Kawaguchi. `<kawa_a24@yahoo.co.jp>`

References

Kawaguchi, A. (2021). Multivariate Analysis for Neuroimaging Data. CRC Press.

`atlas` *Atlas set*

Description

The data is the atlas image data. An image whose element is "ROIid" is stored for each atlas.

Usage

```
data(atlas)
```

Format

A list of array

`atlasdatasets` *Atlas data set*

Description

The data is the atlas data. Various atlases are stored. Each matrix has "ROIid" and "ROIname" as column names.

Usage

```
data(atlasdatasets)
```

Format

A list of matrix

`atlastable` *Result report with atlas data*

Description

This function refers to the results obtained by the analysis in an atlas image, and reports a summary of the results for each anatomical region.

Usage

```
atlastable(x, y, atlasdataset = NULL, ROIids = NULL, ...)

## S3 method for class 'atlastable'
print(x, ...)
```

Arguments

`x`	an array for the atlas image.
`y`	an array for the result image.
`atlasdataset`	a matrix or data.frame. The colnames should include "ROIid" and "ROIname".
`ROIids`	a vector indicating ROI id shown in the result.
`...`	further arguments passed to or from other methods.

Details

`atlastable` requires the atlas image and data frame including the ROI id and the name.

Examples

```
data(diffimg)
data(atlasdatasets)
data(atlas)
atlasname = "aal3"
atlasdataset = atlasdatasets[[atlasname]]
tmpatlas = atlas[[atlasname]]
atlastable(tmpatlas, diffimg, atlasdataset=atlasdataset, ROIids = c(1:2, 41:44))
```

baseimg *Base Brain Data*

Description

The data is the base brain data. This is an average image of a healthy person, and is used when generating artificial data.

Usage

```
data(baseimg)
```

Format

A array

basisprod *Product Radial Basis Function*

Description

This is a function to product the output for the rbfunc function with data matrix for a dimension reduction.

Usage

```
basisprod(A, B)
```

Arguments

`A`	a list or a matrix correponding to the output for the `rbfunc` function with the argument hispec=FALSE or data matrix, respectivey.
`B`	a list or a matrix.

Details

`basisprod` requires one list and one matrix.

Examples

```
imagedim1=c(10,10,10)

B1 = rbfunc(imagedim=imagedim1, seppix=4, hispec=TRUE)
B2 = rbfunc(imagedim=imagedim1, seppix=4, hispec=FALSE)

n = 50
S = matrix(rnorm(n*prod(imagedim1)), nrow = n, ncol = prod(imagedim1))

SB1 = S %*% B1
SB12 = tcrossprod(S, t(B1))
all(SB1-SB12 == 0)

SB2 = basisprod(S, B2)
all(SB1-SB2 == 0)

BS1 = t(B1) %*% t(S)
BS2 = basisprod(B2, S)
all(BS1-t(BS2) == 0)
```

coat *Coat Function*

Description

This is a function for plotting an image. The analysis result can be overcoated on the template.

Usage

```
coat(
  x,
  y = NULL,
  pseq = NULL,
  xyz = NULL,
  col.x = gray(0:64/64),
  col.y = NULL,
  breaks.y = NULL,
  zlim.x = NULL,
  zlim.y = NULL,
  rownum = 5,
  colnum = NULL,
  plane = c("axial", "coronal", "sagittal", "all")[1],
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  paron = TRUE,
  cross.hair = FALSE,
  chxy = NULL,
  color.bar = TRUE,
  regionplot = FALSE,
  atlasdataset = NULL,
  regionname = c("atlas", "stat")[1],
  regionlegend = FALSE,
  atlasname = "",
  ROIids = 1:9,
  ...
)
```

Arguments

`x`	image1. Base image.
`y`	image2 to be overcoated.
`pseq`	a vector plot sequence.
`xyz`	a vector position to be plotted.
`col.x`	a color vector for image1.
`col.y`	a color vector for image2.
`breaks.y`	a vector breaks value for y.
`zlim.x`	a vector plot limitation values for z of x.
`zlim.y`	a vector plot limitation values for z of y.
`rownum`	a numeric, the number of row for the plot.
`colnum`	a numeric, the number of colnum for the plot.
`plane`	a vector plot sequence.
`xlab`	a character for a label in the x axis.
`ylab`	a character for a label in the y axis.
`axes`	a logical. TRUE presents the axes.
`oma`	a vector for outer margin area.
`mar`	a vector for margin.
`bg`	a character for color of background.
`paron`	a logical. TRUE means par is used.
`cross.hair`	a logical.
`chxy`	a vector cross hair position to be plotted.
`color.bar`	a logical.
`regionplot`	a logical.
`atlasdataset`	a matrix or data.frame. colnames shold include "ROIid" and "ROIname".
`regionname`	a character.

`regionlegend`	a logical.
`atlasname`	a character.
`ROIids`	a vector
`...`	further arguments passed to or from other methods.

Details

`coat` requires a image array.

Examples

```
data(exbrain)
coat(exbrain)
```

`diffimg` *Difference Brain Data*

Description

The data is the difference brain data. This represents the difference between the average images of healthy subjects and patients with Alzheimer's disease, and is used when generating artificial data.

Usage

```
data(diffimg)
```

Format

A array

`exbrain` *Example Brain Data*

Description

The data are from a MRI gray matter brain data for one subject.

Usage

```
data(exbrain)
```

Format

A array

imgdatamat *Creat Data Matrix Function*

Description

This is a function that creates a data matrix for analysis from a file saved in image format.

Usage

```
imgdatamat(
  imgfnames,
  mask = NULL,
  ROI = FALSE,
  atlas = NULL,
  atlasdataset = NULL,
  ROIids = NULL,
  zeromask = FALSE,
  schange = FALSE,
  ...
)
```

Arguments

`imgfnames`	a vector for (nifti) file names to be used.
`mask`	a vector for brain mask data.
`ROI`	a logical for roi data set.
`atlas`	an array for the atlas.
`atlasdataset`	a matrix or data.frame. colnames shold include "ROIid" and "ROIname".
`ROIids`	a vector
`zeromask`	a logical for masking voxel with all zeros.
`schange`	a logical for change dimension.
`...`	further arguments passed to or from other methods.

Details

`imgdatamat` requires image file names.

Value

`S`	data matrix
`brainpos`	binary brain position.
`imagedim`	three dimensional vector for image dimension

Examples

```
# imgfnames1 = c("img1.nii", "img2.nii")
# imgdata = imgdatamat(imgfnames1)
```

---

mask *Brain Mask*

Description

The data is the brain mask. This is used to exclude extra-brain regions from the analysis.

Usage

```
data(mask)
```

Format

A array

---

multicoat *Multi Coat Function*

Description

This is a function for plotting an image. The analysis result can be overcoated on the template.

Usage

```
multicoat(imgs, y = NULL, row4imp = 6, col4imp = 1, trm = NULL, ...)
```

Arguments

`imgs`	list of images. Base images.
`y`	image2 to be overcoated.
`row4imp`	the number of rows per a image
`col4imp`	the number of columns per a image
`trm`	the index to trim the top and bottom of the slice
`...`	further arguments passed to or from other methods.

Details

`multicoat` requires a image array.

Examples

```
data(exbrain)
```

multicompplot *Multi components plot*

Description

This is a function that plots the vectorized image returned to its original dimensions by the multirec function.

Usage

```
multicompplot(
  object,
  x,
  comps = NULL,
  row4comp = 6,
  col4comp = 1,
  pseq4comp = NULL,
  ...
)
```

Arguments

`object`	an object of class "`multirec`." Usually, a result of a call to `multirec`
`x`	template image
`comps`	a component sequence to be plotted.
`row4comp`	the number of rows per a component
`col4comp`	the number of columns per a component
`pseq4comp`	the number of images per a component
`...`	further arguments passed to or from other methods.

Details

`multicompplot` requires the output result of `msma` function.

Examples

```
data(baseimg)
data(diffimg)
data(mask)
data(template)
img1 = simbrain(baseimg = baseimg, diffimg = diffimg, mask=mask)
B1 = rbfunc(imagedim=img1$imagedim, seppix=2, hispec=FALSE, mask=img1$brainpos)
SB1 = basisprod(img1$S, B1)
fit111 = msma(SB1, comp=2)
ws = multirec(fit111, imagedim=img1$imagedim, B=B1, mask=img1$brainpos)
multicompplot(ws, template)
```

multirec *Multi components reconstruction*

Description

This is a function that returns the weight vector of multiple components obtained by the msma function applied after dimension reduction by the radial basis function to the same dimension as the original image.

Usage

```
multirec(
  object,
  imagedim,
  B = NULL,
  mask = NULL,
  midx = 1,
  comps = NULL,
  XY = c(”X”, ”Y”, ”XY”)[1],
  signflip = FALSE
)
```

Arguments

object	an object of class msma. Usually, a result of a call to msma
imagedim	a vector for original dimension.
B	a list or a matrix.
mask	a list or a matrix.
midx	a block number.
comps	a component sequence to be plotted.
XY	a character, indicating "X" or "Y". "XY" for the scatter plots using X and Y scores from msma.
signflip	a logical if the sign in the block is flipped to pose the super as possitive.

Details

multirec requires the output result of msma function.

Examples

```
data(baseimg)
data(diffimg)
data(mask)
img1 = simbrain(baseimg = baseimg, diffimg = diffimg, mask=mask)
B1 = rbfunc(imagedim=img1$imagedim, seppix=2, hispec=FALSE, mask=img1$brainpos)
SB1 = basisprod(img1$S, B1)
```

```
fit111 = msma(SB1, comp=2)
ws = multirec(fit111, imagedim=img1$imagedim, B=B1, mask=img1$brainpos)
```

ptest *Prediction Model Function*

Description

This is the function that creates and evaluates the predictive model.

Usage

```
ptest(
  object,
  Z = Z,
  newdata = NULL,
  testZ = NULL,
  regmethod = "glm",
  methods1 = c("boot", "boot632", "cv", "repeatedcv", "LOOCV", "LGOCV")[4],
  metric = "ROC",
  number1 = 10,
  repeats1 = 5,
  params = NULL
)
```

Arguments

`object` a matrix indicating the explanatory variable(s), or an object of class `msma`, which is a result of a call to `msma` .

`Z` a vector, response variable(s) for the construction of the prediction model. The length of Z is the number of subjects for the training.

`newdata` a matrix for the prediction.

`testZ` a vector, response variable(s) for the prediction evaluation. The length of testZ is the number of subjects for the validation.

`regmethod` a character for the name of the prediction model. This corresponds to the `method` argument of the `train` function in the `caret` package.

`methods1` a character for the name of the evaluation method.

`metric` a character for the name of summary metric to select the optimal model.

`number1` a number of folds or number of resampling iterations

`repeats1` a number of repeats for the repeated cross-validation

`params` a data frame with possible tuning values.

Details

`ptest` requires the output result of `msma` function.

Value

object	an object of class "msma", usually, a result of a call to msma

trainout	a predictive model output from the train function in the caret package with scores computed by the msma function as predictors

scorecvroc	the training evaluation measure and values of the tuning parameters

evalmeasure	evaluation measures and information criterion for the msma model

traincnfmat	a confusion matrix in training data

predcnfmat	a confusion matrix in test data

Examples

```
data(baseimg)
data(diffimg)
data(mask)
data(template)
img1 = simbrain(baseimg = baseimg, diffimg = diffimg, mask=mask)
B1 = rbfunc(imagedim=img1$imagedim, seppix=2, hispec=FALSE, mask=img1$brainpos)
SB1 = basisprod(img1$S, B1)
fit111 = msma(SB1, comp=2)
predmodel = ptest(fit111, Z=img1$Z)
```

rbfunc *Radial Basis Function*

Description

This makes a radial basis function.

Usage

```
rbfunc(imagedim, seppix, hispec = FALSE, mask = NULL, brainpos = NULL)
```

Arguments

imagedim	a vector indicating image three dimension.

seppix	a numeric. distance between knots.

hispec	a logical. TRUE produces a matrix output. FALSE produces a list output to reduce the data memorry.

mask	a vector.

brainpos	a logical vector.

Details

rbfunc requires the dimensions of the original image to be applied and the knot interval. The output is obtained as a matrix, with the number of rows corresponding to the number of voxels in the original image and the number of columns determined by the knot spacing. By setting hispec = TRUE, you can get the output in list format with a smaller memory.

Examples

```
imagedim1=c(10,10,10)

B1 = rbfunc(imagedim=imagedim1, seppix=4, hispec=TRUE)
B2 = rbfunc(imagedim=imagedim1, seppix=4, hispec=FALSE)
```

rec *Reconstruction*

Description

This is a function that restores the vectorized image to its original dimensions, reduced in dimension by the radial basis function.

Usage

```
rec(Q, imagedim, B = NULL, mask = NULL)
```

Arguments

Q	a vector for reduced data.
imagedim	a vector for original dimension.
B	a list or a matrix indicating the basis function used in the dimension reduction.
mask	a list or a matrix indicating the mask image used in the dimension reduction.

Details

rec requires a vector to be converted to a array.

Examples

```
imagedim1=c(10,10,10)
recvec = rec(rnorm(prod(imagedim1)), imagedim1)
```

`sdevimg` *Standard Deviation Brain Data*

Description

The data is the standard deviation brain data. This represents the common standard deviation between the average images of healthy subjects and patients with Alzheimer's disease, and is used when generating artificial data.

Usage

```
data(sdevimg)
```

Format

A array

`simbrain` *Generate simulation data Function*

Description

This is a function for simulation data based on the real base brain image data and difference in brain between healty and disease groups.

Usage

```
simbrain(
  baseimg,
  diffimg,
  sdevimg = NULL,
  mask = NULL,
  n0 = 10,
  c1 = 0.5,
  sd1 = 0.01,
  rho = NULL,
  zeromask = FALSE,
  reduce = c("no", "rd1", "rd2")[1],
  output = c("rdata", "nifti")[1],
  seed = 1
)
```

Arguments

`baseimg`	an array for the basis image.
`diffimg`	an array for the difference image.
`sdevimg`	an array for the standard deviation image.
`mask`	an array for the mask image.
`n0`	a numeric, which is a sample size per group.
`c1`	a numeric, the strength of the difference
`sd1`	a numeric, standard deviation for the individual variation.
`rho`	a numeric, correlation coefficient in the noize
`zeromask`	a logical, whether mask the position with zero values for all subjects.
`reduce`	a vector.
`output`	a vector.
`seed`	a numeric for seed for random variables.

Details

`simbrain` requires a base brain image data and mean difference image data.

Value

`S`	data matrix
`Z`	binary group variable
`brainpos`	binary brain position.
`imagedim`	three dimensional vector for image dimension

Examples

```
data(baseimg)
data(diffimg)
sim1 = simbrain(baseimg = baseimg, diffimg = diffimg)
```

`sizechange` *Size change Function*

Description

TThis is a function that changes the resolution of the image.

Usage

```
sizechange(img1, simscale = NULL, refsize = NULL, ...)
```

Arguments

`img1`	a array or nifti class, which is a image data to be changed the size.
`simscale`	a numeric.
`refsize`	a vector with length 3, which is a size to be changed.
`...`	further arguments passed to or from other methods.

Details

`sizechange` requires the array data.

Examples

```
data(exbrain)
exbrain2 = sizechange(exbrain, simscale=1/2)
```

`template` *Brain Template*

Description

The data is the brain tempalte. This is an average brain image, and is mainly used for overlaying analysis results.

Usage

```
data(template)
```

Format

A array

Index