Package 'EZFragility'

April 10, 2025

Title Compute Neural Fragility for Ictal iEEG Time Series

Version 1.0.3

Description Provides tools to compute the neural fragility matrix from intracranial electrocortico-
graphic (iEEG) recordings, enabling the analysis of brain dynamics during seizures. The pack-
age implements the method de-
scribed by Li et al. (2017) <doi:10.23919/ACC.2017.7963378> and includes func-
tions for data preprocessing (`Epoch`), fragility computation (`calcAdjFrag`), and visualization.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

LazyData true

Imports stats, methods, ggplot2 (>= 3.4.0), viridis, ggtext, glue,
rlang, foreach, progress, ramify, reshape2

Suggests knitr, rmarkdown, testthat (>= 3.0.0), doSNOW

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Jiefei Wang [aut, cre] (<https://orcid.org/0000-0002-2709-5332>),
Anne-Cecile Lesage [aut] (<https://orcid.org/0000-0002-9528-4899>),
Ioannis Malagaris [aut] (<https://orcid.org/0000-0001-5126-2068>),
Oliver Zhou [ctb],
Liliana Camarillo Rodriguez [ctb]
(<https://orcid.org/0000-0001-8288-6885>),
Sean O'Leary [ctb] (<https://orcid.org/0000-0003-3650-705X>),
Patrick Karas [ctb]

Maintainer Jiefei Wang <szwjf08@gmail.com>

Repository CRAN

Date/Publication 2025-04-10 14:40:09 UTC

Contents

---

calcAdjFrag	*Calculate adjacency matrices and fragility matrix from iEEG recording*

Description

The function calculates the neural fragility column from an adjacency matrix in each time window

Usage

```
calcAdjFrag(
  epoch,
  window,
  step,
  lambda = NULL,
  nSearch = 100L,
  progress = FALSE,
  parallel = FALSE
)
```

Arguments

`epoch`	Matrix or Epoch object. iEEG data matrix or Epoch object. If matrix, the row names are the electrode names and the column names are the time points
`window`	Integer. The number of time points to use in each window
`step`	Integer. The number of time points to move the window each time
`lambda`	Numeric. The lambda value for regularization to use in the ridge regression. If NULL, the lambda will be chosen automatically ensuring that ensuring that the adjacent matrix is stable (see details)
`nSearch`	Integer. Number of instable eigenvalues with norm=1 to search for the minimum norm perturbation. This parameter is used only when the lambda is NULL
`progress`	Logical. If TRUE, print progress information. If `parallel` is TRUE, this option only support the `doSNOW` backend.
`parallel`	Logical. If TRUE, use parallel computing. Users must register a parallel backend with the foreach package

Details

1/ For each time window i, a discrete stable Linear time system (adjacency matrix) is computed named A_i such that $A_i x(t) = x(t+1)$. The 'lambda' option is the regularization parameter for the ridge regression. `lambda=NULL`(default) will find a lambda value that ensures the stability of the estimated A_i.

2/For each stable estimated A_i, the minimum norm perturbation Γ_{ik} (k index of the electrodes) for column perturbation is computed. Each column is normalized $\frac{max(\Gamma_i) - \Gamma_{ik}}{max(\Gamma_i)}$

Value

A Fragility object

Source

Recreation of the method described in Li A, Huynh C, Fitzgerald Z, Cajigas I, Brusko D, Jagid J, et al. Neural fragility as an EEG marker of the seizure onset zone. Nat Neurosci. 2021 Oct;24(10):1465–74 (pubmed). We have found solutions to fill up missing details in the paper method description

Examples

```
## A dummy example with 5 electrodes and 20 time points
data <- matrix(rnorm(100), nrow = 5)
## create an Epoch object
epoch <- Epoch(data)
windowNum <- 10
step <- 5
lambda <- 0.1
calcAdjFrag(
    epoch = epoch, window = windowNum,
    step = step, lambda = lambda, progress = TRUE
```

```
)

## A more realistic example with parallel computing

if (requireNamespace("doSNOW")) {
    ## Register a SNOW backend with 4 workers
    library(parallel)
    library(doSNOW)
    cl <- makeCluster(4, type = "SOCK")
    registerDoSNOW(cl)

    data("pt01EcoG")
    epoch <- Epoch(pt01EcoG)
    window <- 250
    step <- 125
    title <- "PT01 seizure 1"
    calcAdjFrag(
        epoch = epoch, window = window,
        step = step, parallel = TRUE, progress = TRUE
    )

    ## stop the parallel backend
    stopCluster(cl)
}
```

---

checkIndex   *Check and keep valid index only*

Description

Check and keep valid index only

Usage

```
checkIndex(indices, names)
```

Arguments

indices	Numeric or character index to check
names	Character. All names corresponding to the indices

Epoch   *Constructor for Epoch class*

Description

Constructor for Epoch class

Usage

```
Epoch(data, electrodes = NULL, timeRanges = NULL, times = NULL)
```

Arguments

data	Matrix containing epoch data (rows=electrodes, columns=time points)
electrodes	Optional character vector for electrode names, if not provided, column names of data are used. If both are NULL, electrodes are named E1, E2, ...
timeRanges	Optional numeric vector of 2 containing start and end time points. Only one of times or timeRanges can be non-null
times	Optional numeric vector of time points. Only one of times or timeRanges can be non-null

Value

An Epoch object

Epoch-class   *Epoch Class*

Description

S4 class to handle epoch data with electrodes and time points

Slots

data a tibble containing epoch data (columns=time points, rows=electrodes)

times Numeric vector containing time range

| estimateSOZ | *Find Serzure Onset Zone* |

Description

The function estimates the seizure onset zone (SOZ). For each row, it calculates the maximum, minimum, or mean of row. The rows with the highest values are considered as the SOZ.

Usage

```
estimateSOZ(x, method = c("mean", "max", "min"), proportion = 0.1, ...)
```

Arguments

x	Fragility object
method	Character. The method to use to find the onset zone. Must be one of 'max', 'min', or "mean"
proportion	Numeric. The proportion of electrodes to consider as the onset zone. The electrode number will be rounded to the nearest integer.
...	Additional arguments

Value

A vector of electrode names, or indices if the electrode names are NULL

| fragilityRow | *Compute the normalized fragility row for adjacency matrix A* |

Description

The matrix A is used for the regression: $A * x(t) = x(t+1)$

Usage

```
fragilityRow(A, nSearch = 100, normalize = TRUE)
```

Arguments

A	Numeric. Adjacency Matrix
nSearch	Integer. Number of eigenvalues tried to find the minimum norm vector
normalize	Logical. If TRUE, the fragility row is normalized

---

fragStat	*Compute quantiles, mean and standard deviation for two electrodes group marked as soz non marked as soz*

Description

Compute quantiles, mean and standard deviation for two electrodes group marked as soz non marked as soz

Usage

```
fragStat(frag, sozIndex)
```

Arguments

frag	Matrix or Fragility object. Either a matrix with row as Electrode names and Column as fragility index, or a Fragility object from `calcAdjFrag`
sozIndex	Integer. Vector soz electrodes (for good electrodes)

Value

list of 5 items with quantile matrix, mean and sdv from both electrodes groups

Examples

```
data("pt01Frag")
data("pt01EcoG")
sozIndex <- attr(pt01EcoG, "sozIndex")
pt01fragstat <- fragStat(frag = pt01Frag, sozIndex = sozIndex)
```

---

nrow,Fragility-method   *Get the number of rows or columns of a Fragility object*

Description

Get the number of rows or columns of a Fragility object

Usage

```
## S4 method for signature 'Fragility'
nrow(x)

## S4 method for signature 'Fragility'
ncol(x)
```

Arguments

x   A Fragility object

Value

- `nrow(x)`: The number of rows (electrodes) in the fragility matrix.

- `ncol(x)`: The number of columns (time points) in the fragility matrix.

- `dim(x)`: A vector of length 2 containing the number of rows and columns in the fragility matrix.

---

plotFragHeatmap   *Visualization functions (raw signal, fragility matrix)*

Description

`plotFragHeatmap`: plot fragility heatmaps with electrodes marked as soz colored

`plotFragQuantile`: Plot Fragility time quantiles for two electrodes group marked as SOZ and reference

`plotFragQuantile`: Plot Fragility time distribution for two electrodes group marked as SOZ and reference

Usage

```
plotFragHeatmap(frag, sozIndex = NULL)

plotFragQuantile(frag, sozIndex = NULL)

plotFragDistribution(frag, sozIndex = NULL)
```

Arguments

frag   Fragility object from `calcAdjFrag`

sozIndex   Integer or string. A group of electrodes to mark as in the Seizure Onset Zone (SOZ)

Value

A ggplot object

Examples

```
data("pt01EcoG")

## sozIndex is the index of the electrodes we assume are in the SOZ
sozIndex <- attr(pt01EcoG, "sozIndex")

## precomputed fragility object
data("pt01Frag")

## plot the fragility heatmap
plotFragHeatmap(frag = pt01Frag, sozIndex = sozIndex)

## plot the fragility quantiles
plotFragQuantile(frag = pt01Frag, sozIndex = sozIndex)

## plot the fragility distribution
plotFragDistribution(frag = pt01Frag, sozIndex = sozIndex)
```

---

pt01EcoG   *Pt01 seizure 1 around seizure onset*

Description

This data corresponds to the first seizure of patient from the Fragility Data Set. EcoG recording gathered in collaboration with the National Institute of Health. The data contains only the good channels. It has been notch filtered and common average referenced in RAVE. The time range for full data is (-10:10s). Due to the size limit of the package, The full data has been epoched -1:2s around the seizure onset. The acquisition frequency is 1000 Hz

Usage

```
## EEG data
data(pt01EcoG)
```

Format

pt01EcoG: A Matrix with 84 rows (electrodes) and 3000 columns (time points)

pt01Frag: A fragility object results of applying the main function `calcAdjFrag` to pt01EcoG with window = 250 and step = 125

Source

Fragility Multi-Center Retrospective Study ([OpenNeuro](#))

---

ridge	*fit a generalized linear model to compute adjacency matrix A*

Description

A x(t) = x(t+1)

Usage

```
ridge(xt, xtp1, lambda)
```

Arguments

xt	matrix. iEEG time series for a given window, with electrodes names as rows and time points as columns
xtp1	matrix. the iEEG time serie at the next time point, with electrodes names as rows and time points as columns
lambda	Numeric Vector. A user supplied lambda sequence.

Value

adjacency matrix A

---

ridgeR2	*computes R2*

Description

computes R2

Usage

```
ridgeR2(xt, xtp1, A)
```

Arguments

xt	matrix. iEEG time series for a given window, with electrodes names as rows and time points as columns
xtp1	matrix. the iEEG time serie at the next time point, with electrodes names as rows and time points as columns
A	adjacency matrix

---

ridgeSearch *Ridge Regression for Electrode Readings*

Description

Ridge regression to compute matrix adjancency matrix A such as A xt = xtpt1 the lambda parmeter is found by dichotomy such that A is stable (all eigenvalues have a norm less than one)

Usage

```
ridgeSearch(xt, xtp1, lambda = NULL)
```

Arguments

xt	matrix. iEEG time series for a given window, with electrodes names as rows and time points as columns
xtp1	matrix. the iEEG time serie at the next time point, with electrodes names as rows and time points as columns
lambda	Numeric Vector. A user supplied lambda sequence.

Value

adjacency matrix Afin with lambda as attribute

---

show,Fragility-method *Print the Fragility object*

Description

Print the Fragility object

Usage

```
## S4 method for signature 'Fragility'
show(object)
```

Arguments

object	A Fragility object

Value

the object itself

`show,FragStat-method`   *Print the FragStat object*

Description

Print the FragStat object

Usage

```
## S4 method for signature 'FragStat'
show(object)
```

Arguments

object   A FragStat object

Value

the object itself

`visuIEEGData`   *Visualization of ictal iEEG*

Description

Visualization of ictal iEEG

Usage

```
visuIEEGData(epoch)
```

Arguments

epoch   Matrix or Epoch object. iEEG data matrix or Epoch object. If matrix, the row
               names are the electrode names and the column names are the time points

Value

A ggplot object

Examples

```
data("pt01EcoG")

## Visualize a subject of electrodes
sozIndex <- attr(pt01EcoG, "sozIndex")
display <- c(sozIndex, 77:80)

epoch <- Epoch(pt01EcoG)
visuIEEGData(epoch = epoch[display, ])
```

[,Fragility-method   *Subset a Fragility object*

Description

Subset a Fragility object

Usage

```
## S4 method for signature 'Fragility'
x[i, j, ..., drop = FALSE]
```

Arguments

x	A Fragility object
i	A logical vector or a numeric vector of indices to subset the electrodes
j	A logical vector or a numeric vector of indices to subset the time windows
...	Additional arguments (not used)
drop	Additional arguments (not used)

Value

A new Fragility object with the subsetted data

$,Epoch-method   *Epoch Methods*

Description

$electrodes: Get or set electrode names $times: Get or set time points $timeRange: Get time range if time points are defined $data: Get or set data matrix

[: Subset an Epoch object using matrix indexing syntax

nrow, ncol, colnames, rownames, names: Getting the data properties, similar to base R functions.

truncateTime: Truncating time range

Usage

```
## S4 method for signature 'Epoch'
x$name

## S4 replacement method for signature 'Epoch'
x$name <- value

## S4 method for signature 'Epoch'
x[i, j]

## S4 method for signature 'Epoch'
nrow(x)

## S4 method for signature 'Epoch'
ncol(x)

## S4 method for signature 'Epoch'
colnames(x)

## S4 replacement method for signature 'Epoch'
colnames(x) <- value

## S4 method for signature 'Epoch'
rownames(x)

## S4 replacement method for signature 'Epoch'
rownames(x) <- value

## S4 method for signature 'Epoch'
names(x)

## S4 replacement method for signature 'Epoch'
names(x) <- value

## S4 method for signature 'Epoch'
as.matrix(x)

## S4 method for signature 'Epoch'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

truncateTime(x, from, to)

## S4 method for signature 'Epoch'
truncateTime(x, from, to)

## S4 method for signature 'Epoch'
show(object)
```

Arguments

x	Epoch object
name	a value name, must be one of 'electrodes', 'times', 'timeRange', 'data'
value	Value to set
i	Row (electrode) indices
j	Column (time) indices
row.names	NULL or a character vector giving the row names for the data frame. Missing values are not allowed. See `base::data.frame` for more details.
optional	Logical. If TRUE, setting row names is optional. See base::data.frame for more details.
...	additional arguments
from	Numeric value specifying start of new time range
to	Numeric value specifying end of new time range
object	Epoch object

Value

nrow: Number of rows in the data

ncol: Number of columns in the data

colnames: electrode names of the data

rownames: time points of the data

names: Return all available properties for an Epoch object

truncateTime: Truncated object

`$,FragStat-method`   *Getters and Setters for S4 object*

Description

Getters and Setters for S4 object

Usage

```
## S4 method for signature 'FragStat'
x$name

## S4 replacement method for signature 'FragStat'
x$name <- value

## S4 method for signature 'Fragility'
x$name

## S4 replacement method for signature 'Fragility'
x$name <- value
```

Arguments

x	S4 object
name	Slot name
value	Value to set

Value

S4 object itself or slot value

Index