

Package 'UnitEvents'

June 30, 2017

Type Package

Title Multiple Tests Based on a Gaussian Approximation of the Unitary Events Method with Delayed Coincidence Count (MTGAUE) and Permutation Method for Statistical Tests in Neuroscience

Version 0.0.1

Date

Author Melisande Albert <melisande.albert@math.univ-toulouse.fr>,
Yann Bouret <yann.bouret@gmail.com>,
Magalie Fromont <magalie.fromont@univ-rennes1.fr>,
Franck Grammont <grammont@unice.fr>,
Thomas Laloë <laloë@unice.fr>,
Patricia Reynaud-Bouret <reynaudb@unice.fr>,
Amel Rouis <rouis.a@chu-nice.fr>,
Christine Tuleau-Malot <malot@unice.fr>

→ Julien Chevallier

Maintainer Gilles Scarella <gscarella@unice.fr>

Description Multiple Tests based on a Gaussian Approximation of the Unitary Events method with delayed coincidence count (MTGAUE) and Permutation method. Performs statistical independence tests between two neurons based on coincidence counts. Using C++ neuro-stat code.

License GPL (>=2)

URL <https://github.com/ybouret/neuro-stat>,
<http://math.unice.fr/~malot/liste-MTGAUE.html>

) le nouveau site non ?

R topics documented:

BH	2
BHbN	3
BHtest	4
check_input	5
compute_pvalues	5
compute_time_windows	6
data_treatment	7
DNeur	8

draw_windows	9
findUE	10
get_nspath	11
isequalUE	11
load_neurostatpath	12
maxtime	12
spikes.plot	13
UE.plot	14
UnitEvents	15

Index

17

BH

This function performs Benjamini-Hochberg procedure. *on the p-values*

Description

This function allows to identify the time windows detected by the MTGAUE or permutation procedure.

Usage

BH(alpha, p, plot)

Arguments

alpha double; Array of alpha values or maximum values for the false positive. Could be a vector (in case of several tests)
 p double; Array of p-values. Needs to be ordered.
 plot bool; To plot the result or not. Default value is FALSE.

Value

k The index obtained by Benjamini-Hochberg procedure. Could be an array

Examples

```
p <- sort(runif(10))
alpha <- 0.05
BH(alpha, p, FALSE)
```

*J'aime bcp le plot !!
 la fonction à la fin me sera
 très utile !!*

Si on veut que ce soit facilement utilisable par tout le monde

en entrée : les valeurs non rangées

en sortie : le k ← optimal (comme sort & i)

mais avoir la liste des indices des pval détectées et ça tout le temps !!

on donne les sorted p-values

BHbN

on peut être malheureux ??

ça c'est pas compréhensible³

BHbN

BHbN performs Benjamini-Hochberg procedure through functions of neuro-stat code. wink_coincmat function is used here. See https://github.com/ybouret/neuro-stat

Description

BHbN function allows to identify the time windows detected by the test. It calls wink_coincmat function of neuro-stat.

with synchronization (détection de dépendance)

Usage

BHbN(TW, level, delay, DataNeur, Rtest, neurostatpath, iperm=FALSE, B=10000, num_threads=1, statistical_value="H")

→ si cette fonction n'est jamais décrite dans le package, on peut pas en parler

peut-être que toi par le véif tu vas avoir deux

Arguments

- TW double; Time window matrix with two rows. Each column is associated to a time window and A[,i] contains the bounds of the ith time window.
- level double; level is the value used in ~~p-value computation~~ ^{alpha} BH procedure
- delay double; delay is the value used in coincidence detection for delay.
- DataNeur A DataNeur e.g. the output of DNeur function. Contains the matrices of spike times. ~~Renvoie exactement ce que c'est le copie~~ ^{the coincidence count.}
- Rtest Rtest allows to define the type of the test. Possible values are "all", "symmetric", "upper" and "lower". ^{c'est quoi ??}
- neurostatpath It allows to define the path of neuro-stat code if necessary (string). It could be a relative or an absolute path.
- iperm Logical to select permutation method or not. Default is FALSE. ^{it means no permutation, MTGAUE method is chosen, it performs more rapidly}
- B integer; ~~Number of bootstraps.~~ ^{size of Monte Carlo sample for Permutation method} Default is 10000. Only used by the permutation method.
- num_threads integer; Number of threads for neuro-stat computation. Default is 1.
- statistical_value string; Only possible values are "T" or "H". Default is "H". Corresponds to a centered computation or not.

mais c'est pas ici qu'il faut le dire. Ça s'est dans l'autre pdf. je suppose que tu as pas besoin des autres infos. Permutation method takes more time but is more labor de à mettre)

Value

The output is a list with three elements:

- ndw: number of detected time windows
- prange: a matrix which identifies the detected time windows. The first two rows of this matrix are 0, 1 and -1, 0 meaning that the time window is not detected, -1 meaning that the time window is detected and that the average coincidence count is smaller than the expected one, 1 meaning that the time window is detected and that the average coincidence count is bigger than the expected one (in case of independency).

ce qu'on pouvait faire e'et

si MTGAUE → on renvoie stat MTGAUE
si perm C → on renvoie la
si perm U → on renvoie U (c'est à dire)

under

under independency

ref articles

TW non?

- pvalue: the vector of p-values associated to each test or time window# 0???
- ~~the~~ the numerator of each GAUE symmetric statistic test

See Also

BHtest

?? ~~mais~~ à quoi ça sert ???
 mais pourquoi pas... mieux que compute pvalue en tout cas

Il mettre un exemple complet! → of possible laptop

BHtest

This function is a wrapper to call Benjamini-Hochberg procedure in a general case

Description

Benjamini-Hochberg procedure. It calls the function BHbN.

Usage

```
BHtest(TW, level, delay, DataNeur, neurostatpath = "",
Rtest="all", iperm=FALSE, B=10000, num_threads=1, statistical_value='H')
```

Arguments

- TW** TW is a matrix for the definition of time windows. The matrix has two rows, each column corresponds to a small time window. The number of columns of TW is equal to the number of time windows.
- level** double; level is the value used in p-value computation, level is the maximum value for the false positive.
- delay** double; delay is the value used in coincidence detection.
- DataNeur** A DataNeur e.g. the output of DNeur function. Contains the matrices of spike times
- neurostatpath** string; It allows to define the path of neuro-stat code if necessary. It could be a relative or an absolute path. Default value is empty.
- Rtest** string; Rtest allows to define the type of the test. Only possible values are "symmetric", "upper", "lower".
- iperm** Logical to select permutation method or not. Default is FALSE - it means no permutation and MTGAUE method is chosen
- B** integer; Number of bootstraps used in permutation method. Default is 10000.
- num_threads** integer; Number of threads for neuro-stat computation. Default is 1.
- statistical_value** string; Only possible values are "T" or "H". Default is "H". Corresponds to a centered computation or not.

Value

The detected time windows after Benjamini-Hochberg procedure.

Je compare
 la différence
 à vivre?

See Also

BHbN

check_input *To check if input spike files exist. Returns a list of names of existing spike files and a chain of words as a figure title*

Description

To check if input spike files exist. Returns a list of names of existing spike files and a chain of words as a figure title. If the first neuron file does not exist, the default file is chosen.

Usage

check_input(neur1file, neur2file)

Arguments

neur1file string; Input file name given by the user containing spikes for the first neuron
 neur2file string; Input file name given by the user containing spikes for the second neuron

Value

The output is a list with 2 elements:

- ndf: list of strings; The list contains the names of existing neuron files (after verification). The first item is equal to neur1file if it exists
- titlename: string; It will be used as the title of some figures

Examples

```
out <- check_input('UnitEvents/data/Neur1_c13.txt', 'UnitEvents/data/Neur2_c13.txt')
```

compute_pvalues *Computation of the p-values associated to the GAUE symmetric test*

Description

This function computes the p-values associated to the GAUE symmetric test and the numerator of this test.

Usage

compute_pvalues(C, ntrials, a, b, delay, test)

je comprends pas
cf le
code
d'essai

⚠ il ya
des conventions
qui sont juste
dues à ces
données particulières.

↳ pas
reproductible
↳ ce n'est
pas
dans le
package

je vois pas
l'intérêt
si que GAUE c'est pour
les tests mais du coup
c'est inclus
dans le reste!!

Arguments

C	matrix with 3 rows. It corresponds to the output of the function wink_coincmat
ntrials	int; the number of trials of the datasets used to compute the average coincidence count with delay
a	lower bound of the time window
b	upper bound of the time window
delay	double; value used in coincidence detection
test	string; To define the test; only possible values are "all", "symmetric", "upper" and "lower". Default value is "all" (upper values appear in red, while lower ones appear in blue)

Value

The output of this function is a list with two elements, the first one is the p-value associated to the GAUE symmetric test and the second one is the numerator of the statistic test.

Examples

```
a <- 0 # lowest time
b <- 2 # highest time
ntrials <- 1
delay <- 0.02
C <- matrix(ncol=1,c(0,1,0.5))
test <- "symmetric"
compute_pvalues(C, ntrials, a, b, delay, test)
```

compute_time_windows *To create time windows*

Description

To create time windows of the same duration spanning the whole interval of observation

Usage

compute_time_windows(a, b, height, spacing)

Arguments

a	Starting time of observation— <i>Beginning of experiment recordings</i>
b	End time of observation <i>of experiment recordings</i>
height	Length of a time window. Identical for every time window.
spacing	Value to define a shift. If spacing=height, there is no overlap.

duration between time windows



`data_treatment`

7

Value

Returns a list containing time windows.

- A: Matrix of time windows with two rows and (len-1) columns. Each column is a time window
- L: Time step or spacing between the starts of two consecutive time windows. Equal to input argument spacing.
- len: length of the sequence. Equal to the number of time windows plus one

ça sert à quoi??

Examples

```
# L = 0.01, no overlapping
TW1 <- compute_time_windows(a=0, b=1, height=0.01, spacing=0.01)

# with overlapping
TW2 <- compute_time_windows(a=0, b=1, height=0.1, spacing=0.05)
```

essai avec 0,2,0.1,0.05
la dernière fenêtre dépasse
→ il faut corriger le bug
et vérifier toutes

<code>data_treatment</code>	<i>Treatment of the input spike data, possibly modified.</i>
Description	
Treatment of the input spike data, possibly modified (multiplied by a factor). Matrices of spike times are obtained from input files.	
Usage	
<code>data_treatment(neufiles)</code>	
Arguments	
<code>neufiles</code>	List of input spike files
Value	
The output is a list with 3 elements:	
<ul style="list-style-type: none"> • ntrials: number of trials • N1: First neuron. Contains the number of spikes per trial and spike data. • N2: Second neuron. Contains number of spikes per trial and spike data. 	
See Also	
<code>check_input</code>	
Examples	
<pre>neur1file = "UnitEvents/data/Neur1_c13.txt"; neur2file = "UnitEvents/data/Neur2_c13.txt" I = check_input(neur1file, neur2file) # to check input files neufiles = I[[1]] T = data_treatment(neufiles)</pre>	

les fenêtres qui dépassent;
avec 0,2,2,0.5 on voit bien qu'il y en a trop et il faut un message d'alerte si duration > b-a
y a aussi un truc bizarre si spacing > duration

je comprends pas à quoi ça sert puisqu'on a Data

DNeur

To create DataNeur from input neuron files.

Description

To create DataNeur from input neuron files. A DataNeur contains matrices of size ntrials-by-(nmax+1) containing spike times. nmax is the maximum number of spikes over the trials for each neuron. The first column of the matrix is the number of spikes for each trial. For a missing value, 0 is used to fill the matrix.

Usage

DNeur(neur1file, neur2file)

Arguments

neur1file	string; Input file name given by the user containing spike times for the first neuron
neur2file	string; Input file name given by the user containing spike times for the second neuron.

Value

The output is a list with four elements:

- DN: list of two matrices, one for each neuron. Each matrix contains spike times of the corresponding neuron. Each matrix has a size ntrials-by-(nmax+1) where nmax is the maximum number of spikes over the trials, specific to each neuron. The first column of the matrix is the number of spikes for each trial.
- titlename: string; It will be used as a title of figures
- xrange: Time interval - typically minimum and maximum values of spike times
- ntrials: Number of trials. Should be the same for each neuron

See Also

data_treatment

Examples

out <- DNeur('UnitEvents/data/Neur1_c13.txt', 'UnitEvents/data/Neur2_c13.txt')

ici les données sont en liste ms (je vois) mais se dépend de la matrice par d'échelle et par défaut pas de scale

dire que tu extrais juste le nom du fichier (et envoie convention même à regarder)

c'est pas là qu'il faut le dire.

mais ça peut être fait d'autre manière c'est un autre point ce serait pour l'entrée

A = DNeur (" c13 " ; " c40 ")
 => peut être le débouillage pour message d'erreur plus explicite genre pas le nb d'échantillons...

draw_windows

*To draw time windows. To highlight detected windows after test.***Description**

To draw time windows. Called after Benjamin-Hochberg procedure. Fill and border colors and density can be modified (see polygon function). Default values are "yellow" for fill color, "black" for border and no density.

Usage

```
draw_windows(A1t, xrange, yrange, col="yellow", density="", border="black")
```

Arguments

A	Matrix	double; Time window matrix (with two rows). For example it is the matrix of detected windows after Benjamini-Hochberg procedure
xrange		Double; Numeric vector of length two to define the x-coordinate range in the plot.
yrange		Double; Numeric vector of length two to define the y-coordinate range in the plot
col		string; Fill color in the call to polygon function.. Default value is "yellow".
density		string; Density in the call to polygon function. No density by default.
border		string; Border color in the call to polygon function. Default value is "black".

Examples

```
par(mfrow=c(1,2))
s = seq(0.001, 0.9, 0.1)
A = matrix(ncol=length(s), nrow=2)
A[1,] = s
A[2,] = s+0.1
xrange = c(0,max(A))
yrange = c(-25,0)
draw_windows(A, xrange, yrange)
draw_windows(A, xrange, yrange, col="lightgreen", border=NA, density=c(11))
```

je comprends pas vraiment → dire que c'est pour mettre les fenêtres sous le dessin → bordures neg?

Tu as un pb d'empilement de plot
 a priori, on dirait que ça marche comme lines (rug)
 cod après un dessin, on le rajoute → xrange fixe
 yrange à renommer pour dire que c'est sans le dessin
 → peut être possible de le mettre à une valeur par défaut.

findUE

*To find Unitary Events in detected time windows with a given delay***Description**

To find Unitary Events in detected time windows with a given delay

Usage

findUE(A, DataNeur, delay)

Arguments

A	Matrix of detected time windows. Contains two rows. It should contain detected time windows after Benjamini-Hochberg procedure.
DataNeur	A DataNeur e.g. the output of DNeur function. List which contains spike time matrices
delay	double; delay is the value for delay used in coincidence detection.

Value

The output is a list with 2 elements:

- N1: Detected indices for the first neuron
- N2: Detected indices for the second neuron

← peut être que

renvoyer les temps
pouraient être
plus utile -
ou les deux
(comme sort)

Examples

```
TW = compute_time_windows(a=1e-3, b=2.1, height=1e-1, spacing=0.05)
neur1file = "UnitEvents/data/Neur1_c13.txt"
neur2file = "UnitEvents/data/Neur2_c13.txt"
DataNeur = DNeur(neur1file, neur2file)
delay = 0.02
A = matrix(nrow = 3, ncol=0)
A[1:2,] = TW$A
T = findUE(A, DataNeur, delay)
```

à l'air
bien

mais
voir la
page
pour exemple.

get_nspath

To get the correct path of the neuro-stat code checking if files exist.

Description

To get the correct path of the neuro-stat code. Checks if the dynamic library already exists, otherwise neuro-stat is cloned again.

Usage

get_nspath(neurostatpath)

Arguments

neurostatpath string; It contains the path of the neuro-stat code. For example, 'NEURO/neuro-stat' (on Linux or Mac)

Value

Returns a string containing the path of neuro-stat code

Examples

rpath <- get_nspath('neuro-stat')

ça bugge sur mon MAC

→ la pointer au lieu

isequalUE

To check if two UEs are equal.

Description

To check if two UEs are equal. To be equal, they should have the same detected time windows.

Usage

isequalUE(xns, xpaf)

Arguments

xns First set of Unitary Events. Should be a list with fields A and UE.

xpaf Second set of Unitary Events. Should be a list with fields A and UE.

Value

The output is a logical = TRUE if equality false elsewhere.

je pense que ça c'est une fonction pour toi mais pas dans le package non?

ou sinon tu vois une utilisation que je ne vois pas ...

il manque un
rebe

load_neurostatpath This function allows to the dynamic library of neuro-stat code and source the required R file of neuro-stat.

Description

It loads the dynamic library of neuro-stat code and source the required R file of neuro-stat.

Usage

load_neurostatpath(neurostatpath)

Arguments

neurostatpath string; It contains the path of the neuro-stat code. For example, 'NEURO/neuro-stat' (on Linux or Mac)

maxtime

Function to compute the maximum time value for the plot. Specific user function for Neur1_c13.txt example. of a given Experiment

Description

Function to compute the maximum time value for the plot. Specific user function for Neur1_c13.txt example. Here the value is in the last column of the neuron data-files (in seconds)

of a given experiment. Especially useful

Usage

maxtime(NeuDataFiles)

ça veut dire quoi ??
Si c'est pas récupérer
une donnée d'un fichier
spécifique, on peut pas le faire...

pour plotting
à définir
the windows
to test on.

Arguments

NeuDataFiles List of filenames for neuron data. Each file only contains the number of spikes and spike values.

Value

double; Maximum time value for the plot

Examples

maxtime(list('UnitEvents/data/Neur1_c13.txt', 'UnitEvents/data/Neur2_c13.txt'))

on permet de mettre
un DN (X) X
diagnostique

pour continuer
on peut
lui faire
prendre
en entrée
les codes
de Data Neu
ou des
mode de base

je
comprends
pas
pourquoi
y a ça!
si avec
library(UnitEvent)
on a tout
ça à quoi ça sert ??

spikes.plot	To plot spikes of each neuron
-------------	-------------------------------

Description

To plot spikes of each neuron. Spikes of the first neuron are represented at the bottom in grey color, while spikes of the second neuron are represented at the top in green color. There is a red dividing line.

Usage

spikes.plot(DataNeur, xrange, yrange, titlename)

Arguments

- DataNeur A DataNeur e.g. the output of DNeur function - Contains the matrices of spike times
- xrange Double; Numeric vector of length two to define the x-coordinate range in the plot.
- yrange Double; Numeric vector of length two to define the y-coordinate range in the plot
- titlename string; Title of the figure. Default value is an empty string.

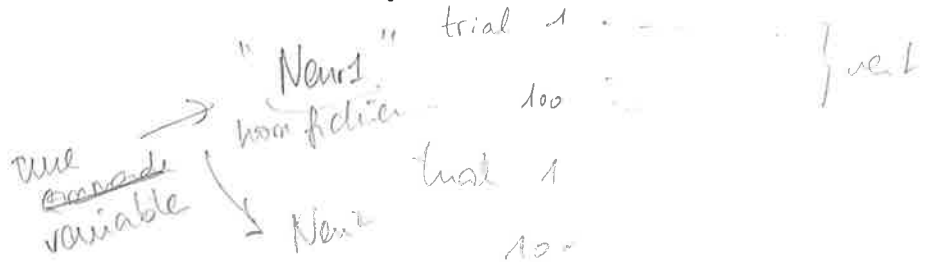
Examples

```
# creation of spikes
lambda <- 50 #firing rate
a <- 0; b <- 2 # times
p=lambda*(b-a)
trials <- 10; N <- 20
ntops <- rpois(N,p)
M <- matrix(data=0,nrow=N,ncol=max(ntops)+1)
M[,1] <- ntops
for (i in 1:N) # for each trial
{
  r <- ntops[i] # depending on ntops
  if (r!=0)
  { x <- runif(r,a,b) # uniform law
    M[i,2:(r+1)] <- sort(x) # sorting
  }
}
F1 <- M[1:trials,]
F2 <- M[(trials+1):(2*trials),]
xrange <- c(0,max(F1[,-1],F2[,-1])) #maximum spike value
yrange <- c(-50,2*nrow(F1[,-1])+2) # number of trials
DNeur <- list(DN=list(F1, F2), ntrials= nrow(F1[,-1]))
titlename <- "Neuro"
spikes.plot(DNeur, xrange, yrange, titlename)
```

par contre
ça serait
bien de
mettre la
fonction
de génération
avec et
sans injection

+ il faut qu'on
puisse changer les
couleurs.
même si on garde
vert et gris par
défaut...
ptêt vert plus
sombre <- vidéo
pro...

pour voir ce que ça donne . → plutôt DataNeur



UE.plot	Function for plotting unitary events (coincident spikes and detected windows)
---------	---

Description

Function for plotting coincident spikes and detected windows

Usage

```
UE.plot(A1t, UE, xrange, yrange)
```

Arguments

A1t	Time window matrix. It should contain detected time windows after Benjamini-Hochberg procedure.
UE	List of Unitary Events. Contains fields N1 (first neuron) and N2 (second neuron). UE\$N1 is a 3-by-ntrials matrix.
xrange	Double; Numeric vector of length two to define the x-coordinate range in the plot.
yrange	Double; Numeric vector of length two to define the y-coordinate range in the plot.

See Also

draw_windows

Examples

```
# time window
#TW$a is a time window matrix
TW <- compute_time_windows(0.001, 0.901, 0.1, 0.1)

# creation of spikes
lambda <- 50 #firing rate
a <- 0; b <- 2 # times
p = lambda*(b-a)
trials <- 10; N <- 20
ntops <- rpois(N,p)
M <- matrix(data=0, nrow=N, ncol=max(ntops)+1)
M[,1] <- ntops
for (i in 1:N) # for each trial
{
  r <- ntops[i] # depending on ntops
  if (r!=0)
  { x <- runif(r,a,b) # uniform law
    M[i,2:(r+1)] <- sort(x) # sorting
  }
}
```

on met
pas les
signes
(-1, 1)?

+ rajouter chgt calcul
possible

appel
automatique
à la demande
data base
(on
autorise
les 2)

```

}
F1 <- M[1:trials,]
F2 <- M[(trials+1):(2*trials),]
UE <-list(N1=F1, N2=F2)

xrange <- c(0,max(F1[,-1],F2[,-1])) #maximum spike value
yrange <- c(-50,2*nrow(F1[,-1])+2) # number of trials

t <- UE.plot(TW$A, UE, xrange, yrange)

```

UnitEvents	<i>Multiple Tests based on a Gaussian Approximation of the Unitary Events method with delayed coincidence count (MTGAUE) or on the permutation method.</i>
------------	--

Description

Multiple Tests based on a Gaussian Approximation of the Unitary Events method with delayed coincidence count (MTGAUE) or on the permutation method. Currently works for two neurons.

Usage

```

UnitEvents(neurostatpath="", DataNeur, delay=0.005, level=0.05,
rasterPlot=TRUE, export=FALSE, Rtest="all", TW,
iperms=FALSE, B=10000, num_threads=1, statistical_value="T")

```

Arguments

<p>à ne pas mettre en premier</p>	<p>neurostatpath</p>	string; It allows to define the path of neuro-stat code if necessary. It could be a relative or an absolute path. Default value is empty which means that default path is used (for example '~/R/x86_64-pc-linux-gnu-library/3.2' on Linux)
	DataNeur	A DataNeur e.g. the output of DNeur function. Contains the matrices of spike times.
	delay	double; delay is the value used in coincidence detection
	level	double; level is the value used in p-value computation, level is the maximum value for the false positive.
	rasterPlot	bool; To plot spikes in the window. Default is TRUE.
	export	bool; To export the raster to png format. Default is FALSE.
	Rtest	string; Rtest allows to define the type of the test. Only possible values are "all", "symmetric", "upper", "lower". Default values is "all" which means that upper and lower tests are performed and discernible.
	TW	To define time windows on which statistical tests are performed; Contains fields A, L and len.
	iperms	Logical to select permutation method or not. Default value is FALSE - it means no permutation and MTGAUE method is chosen

B integer; Number of bootstraps used in permutation method. Default is 10000.
num_threads integer; Number of threads for neuro-stat computation. Default is 1.
statistical_value string; Only possible values are "T" or "H". Default is "H". Corresponds to a centered computation or not.

Value

The output is a two-field list containing

- A: list of all detected time windows as a three-row matrix (the last row contains the detection sign)
- UE: list of unitary events as a two-row matrix with spike time and trial number for each unitary event

References

Multiple Tests based on a Gaussian Approximation of the Unitary Events method with delayed coincidence count. Tuleau-Malot C., Rouis A., Grammont F. and Reynaud-Bouret P., *Neural Computation*, **26**(7), pp1408–1454, 2014.

Surrogate Data Methods Based on a Shuffling of the Trials for Synchrony Detection: the Centering Issue. Albert M., Bouret Y., Fromont M., and Reynaud-Bouret P., *Neural Computation*, **28**(11), pp2352–2392, 2016.

See Also

DNeur

Examples

```
DataNeur = DNeur(neur1file = "UnitEvents/data/Neur1_c13.txt",
neur2file = "UnitEvents/data/Neur2_c13.txt")
res = UnitEvents(DataNeur = DataNeur,
TW = compute_time_windows(a=1e-3, b=2.1, 1e-1, 0.05))
```


Index

BH, 2
BHbN, 3, 5
BHtest, 4, 4

check_input, 5, 7
compute_pvalues, 5
compute_time_windows, 6

data_treatment, 7, 8
DNeur, 3, 4, 8, 10, 13, 15, 16
draw_windows, 9, 14

findUE, 10

get_nspath, 11

isequalUE, 11

load_neurostatpath, 12

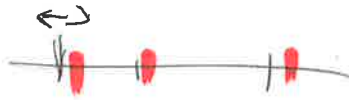
maxtime, 12

spikes.plot, 13

UE.plot, 14
UnitEvents, 15

wink_coincmat, 6

essai 1



essai 2

