Package: SIPETool (via r-universe)

September 13, 2024

Type Package

Title SIFT-MS and CPET Data Processor

Version 0.1.0

Author Federico Vivaldi; Tommaso Lomonaco

Maintainer Federico Vivaldi <federico-vivaldi@virgilio.it>

Description Processor for selected ion flow tube mass spectrometer (SIFT-MS) output file from breath analysis. It allows the filtering of the SIFT output file (i.e., variation over time of the target analyte concentration) and the following analysis for the determination of: maximum, average, and standard deviation value of target concentration measured at each exhalation, and the respiratory rate over the measurement. Additionally, it is possible to align the SIFT-MS data with other on-line techniques such as cardio pulmonary exercise test (CPET) for a comprehensive characterization of breath samples.

License GPL-3

Encoding UTF-8

LazyData true

Imports Convolutioner

RoxygenNote 7.2.3

NeedsCompilation no

Depends R (>= 3.5.0)

Date/Publication 2023-02-22 14:50:07 UTC

Repository https://federicoviv.r-universe.dev

RemoteUrl https://github.com/cran/SIPETool

RemoteRef HEAD

RemoteSha ed511fe41784b8299fe67af7b9e6b31efd1465c5

Contents

`CPET_time` *SIPETool example files*

Description

Raw and filtered output from SIFT-MS, and CPET system. Four different files are available: raw_sift -> raw data from SIFT-MS CPET_time -> data from CPET system for time allignment SIFT_time -> filtered data from SIFT-MS for time allignment SIFT_filtered -> raw data from SIFT-MS filtered using SIFT_output_filter

---

`data_indexer` *data indexer*

Description

This function takes as input a vector and return the data index according to the selected time frame

Usage

```
data_indexer(dat, time_frame_index = NA)
```

Arguments

`dat` input vector

`time_frame_index`
custom data range from the time column

Value

a vector indexed according to the specified time frame

Examples

```
data_indexer(c(1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1))
```

normalizer *Data normalizer*

Description

This function takes as input a vector and returns it normalized between a specified range

Usage

```
normalizer(dat, norm_range = c(0, 1))
```

Arguments

`dat`	the vector to normalize
`norm_range`	the range used for normalization

Value

vector normalized between norm_range

Examples

```
normalizer(c(1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1))
```

raw_SIFT *SIPETool example files*

Description

Raw and filtered output from SIFT-MS, and CPET system. Four different files are available: raw_SIFT -> raw data from SIFT-MS CPET_time -> data from CPET system for time allignment SIFT_time -> filtered data from SIFT-MS for time allignment SIFT_filtered -> raw data from SIFT-MS filtered using SIFT_output_filter

`SIFT_filtered` *SIPETool example files*

Description

Raw and filtered output from SIFT-MS, and CPET system. Four different files are available: raw_sift -> raw data from SIFT-MS CPET_time -> data from CPET system for time allignment SIFT_time -> filtered data from SIFT-MS for time allignment SIFT_filtered -> raw data from SIFT-MS filtered using SIFT_output_filter

`SIFT_output_filter` *SIFT output filter*

Description

This function takes as input the output file generated by the SIFT-MS and returns a .csv containing the TIME and the concentrations data selected by the user

Usage

```
SIFT_output_filter(
  setdir = getwd(),
  input_name = file.choose(),
  output_name,
  n_parameters = 2,
  param_names = c("Isoprene", "Acetone"),
  out_file = TRUE
)
```

Arguments

`setdir`	allow the selection of the working directory
`input_name`	allow the selection of the input file
`output_name`	name of the .csv output file
`n_parameters`	number of analytes
`param_names`	vector with name of the analytes
`out_file`	flag for the export of a csv file

Value

Filtered data and optional csv from SIFT input

Examples

```
data(raw_SIFT)
SIFT_output_filter(input_name = raw_SIFT, output_name = "testfile", out_file = FALSE)
```

`SIFT_time` *SIPETool example files*

Description

Raw and filtered output from SIFT-MS, and CPET system. Four different files are available: raw_sift -> raw data from SIFT-MS CPET_time -> data from CPET system for time allignment SIFT_time -> filtered data from SIFT-MS for time allignment SIFT_filtered -> raw data from SIFT-MS filtered using SIFT_output_filter

`sign_detect` *Sign detection*

Description

This function takes as input a vector and returns the sign of each element

Usage

```
sign_detect(dat)
```

Arguments

`dat`	the vector to be used

Value

vector with the signs of each element of the original matrix

Examples

```
sign_detect(c(1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1))
```

`tidal_analyzer` *Tidal analyzer*

Description

This function takes as input a csv file containing a time column and data columns and returns the position of the end tidals for each data column maximazing the syncronization between data. This function was originally devised for the analysis of the end tidals coming from exhaled breath analyzed through SIFT-MS technology

Usage

```
tidal_analyzer(
  setdir = getwd(),
  input_name = file.choose(),
  output_name,
  starting_threshold = 0.03,
  time_frame = NA,
  out_file = TRUE
)
```

Arguments

`setdir`	working directory
`input_name`	csv file
`output_name`	name of the output file
`starting_threshold`	initial value for the dynamic threshold
`time_frame`	custom data range from the time column
`out_file`	flag for the export of a csv file

Value

csv containing the end tidals, their maximum, average, frequency, and timing

Examples

```
data(SIFT_filtered)
tidal_analyzer(input_name = head(SIFT_filtered, n = 100), output_name = "out", out_file = FALSE)
```

`tidal_finder` *Tidal finder*

Description

This function takes as input a matrix and returns for each column the end tidals depending of the threshold set. It is possible to set a custom time frame for the search of the tidals. Note: a minimum amount of 45 points are necessary.

Usage

```
tidal_finder(
  dat,
  height_threshold = 0.2,
  refine = FALSE,
  time_frame_index = NA
)
```

Arguments

`dat`	the input matrix
`height_threshold`	the minimum height of the tidal
`refine`	refine the dataset
`time_frame_index`	custom time frame

Value

matrix with the tidals for each column

Examples

```
tidal_finder(c(1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1))
```

`time_filter` *Time_alignment*

Description

This function takes as input two data set containing a time vector and a data vector and return the two data sets aligned. This is done by reducing the dimensions of the data set with higher points. The first data set is the one coming from the CPET-ESE and the second one from the SIFT-MS

Usage

```
time_filter(Cy = file.choose(), sift = file.choose())
```

Arguments

`Cy`	CPET-ESE output file
`sift`	SIFT-MS refined file

Value

A plot and the SIFT-MS data file resized for the alignment with the CPET-ESE file

Examples

```
data(SIFT_time)
data(CPET_time)
time_filter(CPET_time, SIFT_time)
```

`trend` *Trend finder*

Description

This function takes as input a vector and returns the trend of each column expressed as the difference between two consecutive elements

Usage

```
trend(dat)
```

Arguments

`dat`	the vector to analyze

Value

vector containing the trend of the each column

Examples

```
trend(c(1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1,1:10,10:1))
```

Index