

Package ‘SwimR’

March 30, 2021

Type Package

Title SwimR: A Suite of Analytical Tools for Quantification of *C. elegans* Swimming Behavior

Version 1.28.0

Date 2013-08-20

Author Jing Wang <jing.wang.2@vanderbilt.edu>, Andrew Hardaway
<hardawayja@gmail.com> and Bing Zhang
<bing.zhang@vanderbilt.edu>

Maintainer Randy Blakely <Randy.Blakely@vanderbilt.edu>

Description SwimR is an R-based suite that calculates, analyses, and plots the frequency of *C. elegans* swimming behavior over time. It places a particular emphasis on identifying paralysis and quantifying the kinetic elements of paralysis during swimming. Data is input to SwimR from a custom built program that fits a 5 point morphometric spine to videos of single worms swimming in a buffer called Worm Tracker.

License LGPL-2

LazyLoad yes

Depends R (>= 3.0.0), methods, gplots (>= 2.10.1), heatmap.plus (>= 1.3), signal (>= 0.7), R2HTML (>= 2.2.1)

Imports methods

Collate createFrequencyMatrix.R SwimR.R

biocViews Visualization

git_url <https://git.bioconductor.org/packages/SwimR>

git_branch RELEASE_3_12

git_last_commit 63b051a

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

R topics documented:

| | |
|---------------------------------|---|
| SwimR-package | 2 |
| createFrequencyMatrix | 2 |
| SwimR | 4 |

Index**8**

| | |
|---------------|--|
| SwimR-package | <i>SwimR: A Suite of Analytical Tools for Quantification of C. elegans Swimming Behavior</i> |
|---------------|--|

Description

SwimR is an R-based suite that calculates, analyses, and plots the frequency of *C. elegans* swimming behavior over time. It places a particular emphasis on identifying paralysis and quantifying the kinetic elements of paralysis during swimming. Data is input to SwimR from a custom built program that fits a 5 point morphometric spine to videos of single worms swimming in a buffer called Worm Tracker.

Details

| | |
|-----------|-------------|
| Package: | SwimR |
| Type: | Package |
| Version: | 0.99.1 |
| Date: | 2013-08-17 |
| License: | LGPL (>= 2) |
| LazyLoad: | yes |

Author(s)

Jing Wang, Andrew Hardaway, Bing Zhang Maintainer: Randy Blakely <Randy.Blakely@Vanderbilt.edu>

References

An Open-Source Analytical Platform for Analysis of *C. elegans* Swimming Induced Paralysis (under review).

See Also

[SwimR](#)

| | |
|-----------------------|--|
| createFrequencyMatrix | <i>Analyzing the frequency of worm thrashing over time and creating the frequency matrix and annotation file</i> |
|-----------------------|--|

Description

This function analyzes the frequency of worm thrashing over time for the files returned by the Tracker program and then combined the frequency of all files into a matrix and extracted genotype information from tracker file names to create annotation file.

Arguments

| | |
|-----------------|--|
| inputPath | inputPath is a directory which contains the files returned by the Tracker program. Because annotation file is generated by extracting the genotype information from tracker file names, the user should use dashes to denote genotypes and separate the date in the file name like Genotype_Drug(if used)_Dose(if used)_Date_#. The following are some examples of tracker file names: N4_AMPH_100uM_2-11-14_1 (genotype is N4), dat-1ok157_IMI_10uM_12-12-12_7 (genotype is dat-1ok157), cat-2e1112dat-1ok157_2-20-09_2 (genotype is cat-2e1112dat-1ok157) and dat-1ok157dop-3vs106_5-17-06_4 (genotype is dat-1ok157dop-3vs106). |
| outputPath | outputPath is a directory which saves the plots and files returned by the function. |
| method | The function provides four different counting methods: "FFT" (Fast Fourier Transform), "Extrema", "PeakDet" (peak delta) and "RT+GP" (Get Peaks plus Racetrack Filter) and the users can select one of them to output the corresponding frequency matrix. The default outputted method is "Extrema". |
| Threshold | Threshold is the amount of degrees (in radians) required to count at as thrash and the default is 0.6. |
| DeltaPeakDt | DeltaPeakDt is the threshold for "peak delta" algorithm, similar to Threshold and the default is 1.6. |
| MinFrameBtwnMax | MinFrameBtwnMax is the minimum number of frames between maxima and the default is 4. |
| MinDelta | MinDelta is similar to DeltaPeakDet and the default is 2.5. |
| longPeriod | longPeriod is the longest period cycle which is not zero and the default is 5. |
| AvWindowSize | AvWindowSize is the length of the average window in seconds and the default is 10 seconds. |
| fps | fps is the frame per second and the default is 15. |
| ZP_Length | ZP_Length is the Zero-padding length and the default is 100. |
| WindowSize | WindowSize is the size of window for computing the Fast Fourier Transform and the default is 30. |
| MaxCompWin | MaxCompWin is the window size on deciding if the current angle is a maxima and the default is 2. |
| minTime | minTime is the minimum threshold of time points for the following analysis and the default is 0 second. |
| maxTime | maxTime is the maximum threshold of time points for the following analysis and the default is 600 seconds. |

Details

The createFrequencyMatrix function outputs six files:

1. outputDescription_createFrequencyMatrix.html contains a summary of all output files.
2. XFig.jpg is the image of scatter plot of one animal plotted as "Frequency vs Time(min)" with all four counting methods overlaid. "X" of "XFig.jpg" represents the input file names.
3. XFigSub.jpg is the same as XFig.jpg except counting methods are broken up into four different plots. This is very helpful in checking through a video to make sure that Tracker tracked the worm properly. Bad contrast can be a problem with Tracker missing the worm and these files help to identify troublesome videos for retracking or discarding.

4. XFreq.csv is the CSV file of raw data organized by column, where column one represents frequency as counted by FFT, column two represents frequency calculated by Extrema, column three represents frequency calculated by PeakDt, column four represents frequency as counted by RT+GP and column five represents time in seconds.
5. frequencyMatrix.txt is a TXT file which contains the information of frequencyMatrix.
6. annotationfile.txt is a TXT file which contains the information of annotation.

Value

The createFrequencyMatrix function returns a list object which contains the following information:

| | |
|-----------------|--|
| frequencyMatrix | frequencyMatrix combines the analysis results of the frequency of worm thrashing over time for all Tracker files in the inputPath. |
| annotation | annotation contains all genotype information extracted from file names of all Tracker files in the inputPath. |

Note

The worm frequency analysis script in this function is converted from matlab script coded by Katherine Fleming and Paul Fleming.

Author(s)

Jing Wang and Andrew Hardaway

See Also

[SwimR](#)

Examples

```
inputPath <- system.file("extdata", "trackerFiles", package="SwimR")
outputPath <- getwd()
freMat <- createFrequencyMatrix(inputPath, outputPath, method = "Extrema",
Threshold = 0.6, DeltaPeakDt = 1.6, MinFrameBtwnMax = 4, MinDelta = 2.5, longPeriod = 5,
AvWindowSize = 10, fps = 15, ZP_Length = 100, WindowSize = 30, MaxCompWin = 2, minTime = 0,
maxTime = 600)
```

Description

This function analyzes and visualizes worm swimming data returned by `link{createFrequencyMatrix}`. It places a particular emphasis on identifying paralysis and quantifying the kinetic elements of paralysis during swimming.

Arguments

| | |
|---------------------------------------|---|
| <code>expfile</code> | <code>expfile</code> is the path of the frequency matrix returned by the <code>link{createFrequencyMatrix}</code> function. |
| <code>annfile</code> | <code>annfile</code> is the path of annotation file returned by the <code>link{createFrequencyMatrix}</code> function. |
| <code>projectname</code> | <code>projectname</code> is the name of the project. |
| <code>outputPath</code> | <code>outputPath</code> is a directory which saves the plots and files returned by the function. |
| <code>color</code> | The function provides four colors to plot the heat map plot: "red/green", "red/blue", "yellow/blue" and "white/black". The default color is "red/green". |
| <code>data.collection.interval</code> | <code>data.collection.interval</code> is the time interval between two points and the default is 0.067. |
| <code>window.size</code> | <code>window.size</code> is the size of the window for the running average that is calculated to smooth the data. The default is 150. |
| <code>mads</code> | <code>mads</code> is the number of median absolute deviations that a given animal must deviate from the median sum of frequencies to be called an outlier. The default is 4.4478. |
| <code>quantile</code> | <code>quantile</code> is the proportion of data points that are used in calculating the color scheme for the heat map and the default is 0.95. |
| <code>interval</code> | <code>interval</code> is the minimum time that a given animal must lie below a threshold to be called a paralyzed worm for the first calculation and the default is 20. |
| <code>degree</code> | <code>degree</code> is the paralytic degree for the first calculation and the default is 0.2. |
| <code>paralysis.interval</code> | <code>paralysis.interval</code> is the same as <code>interval</code> but for the second calculation and the default is 20. |
| <code>paralysis.degree</code> | <code>paralysis.degree</code> is the paralytic degree for the second calculation and the default is 0.2. |
| <code>rev.degree</code> | <code>rev.degree</code> is the threshold that an animal must cross to be called a revertant and the default is 0.5. |

Details

The SwimR function outputs 13 files: 1. `output_SwimR.html` contains a summary of all output files.

2. `P_sample_t_half.txt` is a TXT file which contains the information of `sample_t_half`. "P" of "`P_sample_t_half.txt`" is the `projectname` inputted by users.

3. `P_group_data.txt` is a TXT file which contains the information of `group_data`.

4. `P_heatmap_withingroup_ordered_globalcentering.jpg` is a JPEG file of the heat map of all of the samples included in the data matrix after outlier exclusion, smoothing, ordering based on the latency to paralyse, and centering the color based on the quantile percent that can be set by the user in the parameters section of SwimR.

5. `P_heatmap_withingroup_ordered.txt` is a TXT file of the raw data used to plot the heat map which is the same with `group_ordered.data`.

6. After exclusion and smoothing, `P_histogram.nooutliers.smoothed.jpg` is a JPEG file of all frequency data points broken up into increasing 0.1 Hz bins and then plotted as the fraction of the total as a histogram.

7. `P_histogram.nooutliers.smoothed.data.G.txt` is a TXT file of the raw data used to plot the histogram, which is the same with `nooutliers.smoothed.data`. "G" in the "`P_histogram.nooutliers.smoothed.data.G.txt`" is the genotype in the annotation file.
8. `P_individual_data.txt` is a TXT file which contains the information of `individual_data`. If there is no paralyzed animal, this file will not be outputted.
9. `P_individual_data1.txt` is a TXT file which contains the information of `individual_data1`. If there is no paralyzed animal, this file will not be outputted.
10. `P_intermediate.results.txt` describes some key features of your samples after running SwimR, and is a great way to get a quick look at the incidence of paralysis amongst your samples. At the top of the file, it lists the parameters used in the subsequent calculations. Below that, it lists the summed frequency values for each of the animals included in the sample. And then the p value of the bimodal test for each genotype was listed. Below that, it lists each of the animals included and excluded after outlier detection. After that, it lists which animals were considered paralyzed and which not. For paralyzed animals, it then lists which of them were called revertants.
11. `P_scatter.jpg` is a JPEG image of the average frequency plotted against time after outlier exclusion, but w/o smoothing.
12. `P_nooutliers_smoothed_scatter.jpg` is a JPEG image of the average frequency plotted against time after outlier exclusion and smoothing.
13. `P_nooutliers_smoothed_scatter_data.txt` is a TXT file of the raw data used to plot the smoothed scatter.pdf, which is the same with `group_means`.

Value

The SwimR function returns a list object which contains the following information:

| | |
|----------------------------|--|
| <code>sample_t_half</code> | <code>sample_t_half</code> contains each animal and their corresponding latency to paralyze. For non-paralyzers, N/A will be listed. |
| <code>group_data</code> | The columns of <code>group_data</code> is defined as follow. "freq_max_mean": Mean maximal swimming frequency; "freq_max_sd": Standard deviation of Mean maximal swimming frequency; "freq_min_mean": Mean minimum swimming frequency; "freq_min_sd": Standard deviation of Mean minimum swimming frequency; "freq_range_mean": Mean range between maximum and minimum; "freq_range_sd": Standard deviation of Mean range between maximum and minimum; "paralytic_count": The number of paralyzed animals amongst the samples; "non-paralytic_count": The number of non-paralyzed animals amongst the samples; "t_half_mean": Mean latency to cross the paralytic threshold set by the users (default is 20 interval (default is 20 seconds); "t_half_sd": Standard deviation of <code>t_half_mean</code> ; "t_p_start_mean": The mean time point (in seconds) at which each animal crosses a frequency that is min+paralytic threshold and stays below that threshold for the paralytic interval; "t_p_start_sd": Standard deviation of <code>t_p_start_mean</code> ; "t_p2end_mean": The average range of time after paralysis; "t_p2end_sd": Standard deviation of <code>t_p2end_mean</code> ; "rev_count": The number of revertants amongst the samples as defined by the threshold set by the user (default is animals have to recross 50 frequency range for any length of time; "rev_percent": The number of revertants; "rev_frequency_mean": The number of reversion events; "t_p2r_mean": Mean time between 1st reversion and <code>t_p_start_mean</code> ; "t_p2r_sd": Standard deviation of <code>t_p2r_mean</code> ; "t_r_total_mean": Mean of total time spent in reversion for all revertants; "t_r_total_sd": Standard deviation of <code>t_r_total_mean</code> ; "t_r_average_mean": Mean length of an individual reversion event; "t_r_average_sd": Standard deviation of <code>t_r_average_mean</code> ; "r_amp_mean": Mean of total amplitude of reversion for all revertants, where |

amplitude is defined by the area beyond the reversion threshold set by user (default is 50 discrete values for each measurement (same unit as frequency));
 "r_amp_sd": Standard deviation of r_amp_mean.

`group.ordered.data`

`group.ordered.data` contains the data after outlier exclusion, smoothing and ordering based on the latency to paralyze.

`individual.data`

`individual.data` contains reversion information for individual animals. The definitions are identical to the `group_data`, but "R_count" is the number of reversion events for that animal. If there is no paralyzed animal, it will not be returned.

`individual.data1`

For animals that paralyzed: The `R_instances` row tells the user exactly when the animal reverted. For animals that did not revert, N/A will be listed. If there is no paralyzed animal, it will not be returned.

`group_means`

`group_means` contains average frequency and standard deviation for each group. The row names are the time.

`nooutliers.smoothed.data`

`nooutliers.smoothed.data` is a list object which contains all frequency data points broken up into increasing 0.1 Hz bins after exclusion and smoothing for each of genotypes.

Author(s)

Jing Wang, Andrew Hardaway and Bing Zhang

See Also

[createFrequencyMatrix](#)

Examples

```
expfile <- system.file("extdata", "SwimExample", "SwimR_Matrix.txt", package="SwimR")
annfile <- system.file("extdata", "SwimExample", "SwimR_anno.txt", package="SwimR")
projectname <- "SwimR"
outputPath <- getwd()
result <- SwimR(expfile, annfile, projectname, outputPath, color = "red/green",
data.collection.interval = 0.067, window.size = 150, mads = 4.4478, quantile = 0.95,
interval = 20, degree = 0.2, paralysis.interval = 20, paralysis.degree = 0.2,
rev.degree = 0.5)
```

Index

* **methods**

createFrequencyMatrix, [2](#)
SwimR, [4](#)

* **package**

SwimR-package, [2](#)

createFrequencyMatrix, [2](#), [7](#)

SwimR, [2](#), [4](#), [4](#)

SwimR-package, [2](#)