

Package ‘saccadr’

July 23, 2025

Type Package

Title Extract Saccades via an Ensemble of Methods Approach

Version 0.1.3

Date 2023-09-06

Description A modular and extendable approach to extract (micro)saccades from gaze samples via an ensemble of methods.
Although there is an agreement about a general definition of a saccade, the more specific details are harder to agree upon.
Therefore, there are numerous algorithms that extract saccades based on various heuristics, which differ in the assumptions about velocity, acceleration, etc. The package uses three methods (Engbert and Kliegl (2003) <[doi:10.1016/S0042-6989\(03\)00084-1](https://doi.org/10.1016/S0042-6989(03)00084-1)>, Otero-Millan et al. (2014) <[doi:10.1167/14.2.18](https://doi.org/10.1167/14.2.18)>, and Nyström and Holmqvist (2010) <[doi:10.3758/BRM.42.1.188](https://doi.org/10.3758/BRM.42.1.188)>) to label individual samples and then applies a majority vote approach to identify saccades. The package includes three methods but can be extended via custom functions. It also uses a modular approach to compute velocity and acceleration from noisy samples. Finally, you can obtain methods votes per gaze sample instead of saccades.

License GPL (>= 3)

URL <https://github.com/alexander-pastukhov/saccadr/>,
<https://alexander-pastukhov.github.io/saccadr/>

BugReports <https://github.com/alexander-pastukhov/saccadr/issues/>

Imports magrittr, Rcpp (>= 1.0.8)

LinkingTo Rcpp

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

VignetteBuilder knitr

Depends R (>= 4.1.0), dplyr, rlang, cluster, signal, tidyr

Suggests rmarkdown, knitr, ggplot2, spelling, testthat (>= 3.0.0)
Language en-US
Config/testthat/edition 3
NeedsCompilation yes
Author Alexander Pastukhov [aut, cre] (ORCID:
 <<https://orcid.org/0000-0002-8738-8591>>)
Maintainer Alexander Pastukhov <pastukhov.alexander@gmail.com>
Repository CRAN
Date/Publication 2023-09-06 15:30:02 UTC

Contents

compute_velocity_ek	2
diff_ek	3
diff_nh	4
extract_saccades	5
method_ek	8
method_nh	10
method_om	11
monocular_ten_trials	12
option_or_default	12
sd_via_median_estimator	13
single_trial	14
single_trial_binocular	14
Index	15

compute_velocity_ek	<i>Compute velocity via Engbert & Kliegl (2003) algorithm.</i>
---------------------	--

Description

Compute velocity via Engbert & Kliegl (2003) algorithm. See formula #1 in the manuscript.

Usage

compute_velocity_ek(x, trial, time_window_in_samples, delta_t)

Arguments

- x Vector of coordinates
- trial Vector with trial labels, so that velocity is computed only within trials.
- time_window_in_samples Width of window for velocity computation in samples.
- delta_t Duration of a single frame (1 / sampling rate).

Value

Velocity vector

diff_ek	<i>Differentiate x and y and compute change amplitude via an algorithm proposed by Engbert & Kliegl (2003)</i>
---------	--

Description

Differentiate x and y and compute change amplitude via an algorithm proposed by Engbert & Kliegl (2003)

Usage

```
diff_ek(x, y, trial, sample_rate, options = NULL)
```

Arguments

x	vector with x coordinates in <i>degrees of visual angle</i>
y	vector with y coordinates in <i>degrees of visual angle</i>
trial	vector with trial index
sample_rate	sample rate in Hz
options	List with method specific options, see Details.

Details

Method options, please refer to Engbert & Kliegl (2003) for details on parameters and the rationale for default values.

- `ek_velocity_time_window` Time window for velocity computation in milliseconds. Defaults to 20 ms.

Value

data.frame with columns x, y, and amp

See Also

`compute_velocity_ek`

Examples

```
diff_ek(rnorm(1000), rnorm(1000), rep(1, 1000), 250, list("ek_velocity_time_window" = 20))
```

diff_nh	<i>Differentiate x and y and compute change amplitude via an algorithm proposed by Nyström and Holmqvist (2010) doi:10.3758/BRM.42.1.188doi:10.3758/BRM.42.1.188</i>
---------	--

Description

Differentiate x and y and compute change amplitude via an algorithm proposed by Nyström and Holmqvist (2010) [doi:10.3758/BRM.42.1.188](https://doi.org/10.3758/BRM.42.1.188). Note that both components and the amplitude are smoothed *independently* via a Savitzky-Golay filter, so the components may not (probably won't) add up to the amplitude. Note that filtering is sensitive to the presence of NA.

Usage

```
diff_nh(x, y, trial, sample_rate, options = NULL)
```

Arguments

x	vector with x coordinates in <i>degrees of visual angle</i>
y	vector with y coordinates in <i>degrees of visual angle</i>
trial	vector with trial index
sample_rate	sample rate in Hz
options	List with method specific options, see Details.

Details

Method options, please refer to Nyström and Holmqvist (2010) for details on parameters and the rationale for default values.

- nh_sg_filter_order Order of Savitzky-Golay filter. Defaults to 2.

Value

data.frame with columns x, y, and amp

See Also

`filter_via_savitzky_golay`

Examples

```
diff_nh(rnorm(1000), rnorm(1000), rep(1, 1000), 250, list("nh_sg_filter_order" = 2))
```

extract_saccades	<i>Extract saccades from samples using votes from selected methods.</i>
------------------	---

Description

Extract saccades from samples using votes from selected methods. Each method votes whether a given sample belongs to a saccade. Next, saccades are identified via a majority vote using the `vote_threshold` parameter, as well as a minimum duration and minimal temporal separation criteria. Please note that units of the gaze samples must be in **degrees of visual angle**. The units are important as some methods use specific (e.g., physiologically plausible) velocity and acceleration thresholds.

By default, ensemble includes methods proposed by Engbert & Kliegl (2003) ("ek"), Otero-Millan et al. ("om"), and Nyström & Holmqvist (2010) ("nh"), see *Implemented Methods* vignette. However, it can be extended via custom methods, see *Using Custom Methods* vignette.

By default, the function returns a table with identified saccades but can return a matrix with methods' votes per sample instead (`return_votes = TRUE`).

Usage

```
extract_saccades(
  x,
  y,
  sample_rate,
  trial = NULL,
  methods = list(method_ek, method_om, method_nh),
  velocity_function = saccadr::diff_ek,
  options = NULL,
  binocular = "merge",
  vote_threshold = ifelse(length(methods) == 1, 1, (length(methods) - 1)),
  minimal_duration_ms = 12,
  minimal_separation_ms = 12,
  return_votes = FALSE
)
```

Arguments

<code>x</code>	Horizontal coordinate, either a vector for monocular data or a two-column matrix for binocular data.
<code>y</code>	Vertical coordinate, either a vector for monocular data or a two-column matrix for binocular data.
<code>sample_rate</code>	Sampling rate in Hz. It is assumed to be common for the entire time series. If the time series contains chunks (trials) that were recorded using different acquisition rate (e.g., SR Research Eyelink allows to set different acquisition rate for each recording / trial), you would need to split the time series and analyze them separately.

trial	Optional vector with trial ID. If omitted, all samples are assumed to belong to a single trial. Velocity, acceleration, and saccades themselves are computed respecting trial borders.
methods	A list with saccade detection methods, can include external functions that implement sample classification (see <i>Using Custom Methods</i> vignette). Package methods include Engbert & Kliegl (2003) (<code>method_ek</code>), Otero-Millan et al. (2014) (<code>method_om</code>), Nyström and Holmqvist (2010) (<code>method_nh</code>). Defaults to the list of all internally implemented methods: <code>list(method_ek, method_om, method_nh)</code> .
velocity_function	A handle to a function to compute velocity and acceleration. Defaults to a method suggested by Engbert & Kliegl (2003) <code>diff_ek</code> . The package also implements the method proposed by Nyström and Holmqvist (2010) <code>diff_nh</code> . See vignette "Velocity computation" for details and information on how to implement a custom method.
options	A named list with options for saccade detection (see <code>method_ek</code> , <code>method_om</code> , <code>method_nh</code>) and velocity (<code>diff_ek</code> , <code>diff_nh</code>) computation. See documentation on specific method for details.
binocular	Specifies how a binocular data is treated. Options are "cyclopean" (binocular data is converted to an average cyclopean image before saccades are extracted), "monocular" (saccades are extracted independently for each eye), "merge" (default, sample votes are obtained from both eyes and for all methods and then averaged. This way only binocular saccades, i.e., eye movements with a sufficient temporal overlap between eyes, are detected.).
vote_threshold	Value between 1 and N (where N is number of used methods) defining a vote threshold for a saccade. By default, all but one method $threshold = N - 1$ must agree for a sample to be considered for a saccade. Threshold of 1 is applied if a single method is used.
minimal_duration_ms	Minimal duration of a saccade in milliseconds. Shorter candidate saccades are discarded,
minimal_separation_ms	Minimal time separation between saccades in milliseconds. Saccades that are separated by a shorter interval of "not a saccade" votes, will be merged including that period.
return_votes	Logical. Whether function should return extracted microsaccades (FALSE, default) or votes per sample (TRUE).

Details

Variables that describe saccade

- Trial Trial index.
- Eye "Monocular" for monocular inputs. "Cyclopean" for binocular data that was averaged *before* applying algorithms. "Binocular" for binocular data with votes averaged *after* applying algorithms. "Left" or "Right" for binocular data when eyes are processed independently.
- OnsetSample Index of the first sample.

- OffsetSample Index of the last sample.
- Onset Onset time relative to the trial start in milliseconds.
- Offset Offset time relative to the trial start in milliseconds.
- Duration Duration in milliseconds.
- DisplacementX Horizontal displacement measured from the *first* to the *last* sample.
- DisplacementY Vertical displacement measured from the *first* to the *last* sample.
- Displacement Displacement magnitude measured from the *first* to the *last* sample.
- DisplacementPhi Displacement direction measured from the *first* to the *last* sample.
- AmplitudeX Horizontal displacement measured from the *leftmost* to the *rightmost* sample.
- AmplitudeY Vertical displacement measured from the *lowest* to the *uppermost* sample.
- Amplitude Displacement magnitude measured from the most extreme samples.
- AmplitudePhi Displacement direction measured from the most extreme samples.
- VelocityPeak Peak velocity.
- VelocityAvg Average velocity.
- AccelerationPeak Peak acceleration.
- AccelerationAvg Average acceleration.
- AccelerationStart Peak acceleration *before* peak velocity was reached.
- AccelerationStop Peak acceleration *after* peak velocity was reached.

Value

A data.frame with saccade properties (see **details**), if return_votes = FALSE. Alternatively, it returns votes per sample (return_votes = TRUE). For a monocular processing (monocular input, cyclopean or merged binocular data) it is a matrix with nrow(x) rows and length(methods) columns with 0/1 votes for each sample and method. For binocular processing, function returns a two element list with the similar matrices but per eye.

See Also

[method_ek](#), [method_om](#), [method_nh](#), [diff_ek](#), [diff_nh](#)

Examples

```
# Single trial
data(single_trial)
saccades <- extract_saccades(single_trial$x, single_trial$y, 500)

# Multiple trials
data(monocular_ten_trials)
saccades <- extract_saccades(monocular_ten_trials$x,
                             monocular_ten_trials$y,
                             500,
                             trial = monocular_ten_trials$trial)

# binocular saccades
```

```

data("single_trial_binocular")
saccades_b <- saccadr::extract_saccades(single_trial_binocular[, c('xL', 'xR')],
                                       single_trial_binocular[, c('yL', 'yR')],
                                       sample_rate = 1000)

# cyclopean saccades from binocular data
saccades_c <- saccadr::extract_saccades(single_trial_binocular[, c('xL', 'xR')],
                                       single_trial_binocular[, c('yL', 'yR')],
                                       sample_rate = 1000,
                                       binocular = "cyclopean")

# monocular saccades from binocular data
saccades_m <- saccadr::extract_saccades(single_trial_binocular[, c('xL', 'xR')],
                                       single_trial_binocular[, c('yL', 'yR')],
                                       sample_rate = 1000,
                                       binocular = "monocular")

# Using a single method
saccades <- extract_saccades(single_trial$x, single_trial$y, 500, methods = method_om)

# Using two methods
saccades <- extract_saccades(single_trial$x,
                             single_trial$y,
                             500,
                             methods = list(method_ek, method_om))

# Alternative velocity computation method
saccades <- extract_saccades(single_trial$x, single_trial$y, 500, velocity_function = diff_nh)

# A strict unanimous decision threshold
saccades <- extract_saccades(single_trial$x, single_trial$y, 500, vote_threshold = 3)

# A slacker criterion that at least one of the three methods must label sample as a saccade
saccades <- extract_saccades(single_trial$x, single_trial$y, 500, vote_threshold = 1)

# Only longish saccades are extracted
saccades <- extract_saccades(single_trial$x, single_trial$y, 500, minimal_duration_ms = 20)

```

method_ek

Extract saccades using an algorithm proposed by Engbert and Kliegl (2003) [Rhrefhttps://doi.org/10.1016/S0042-6989%2803%2900084-1](https://doi.org/10.1016/S0042-6989%2803%2900084-1)doi:10.1016/S00426989(03)000841

Description

Extract saccades using an algorithm proposed by Engbert and Kliegl (2003) [doi:10.1016/S0042-6989\(03\)00084-1](https://doi.org/10.1016/S0042-6989(03)00084-1)

Usage

```
method_ek(x, y, vel, acc, sample_rate, trial, options)
```


Arguments

x	Gaze x coordinate, <code>_arbitrary units_</code> as threshold velocity is computed in units of standard deviation.
y	Gaze y coordinate, <code>_arbitrary units_</code> as threshold velocity is computed in units of standard deviation.
vel	Velocity data.frame with columns x, y, amp.
acc	Acceleration data.frame with columns x, y, amp.
sample_rate	Sample rate in Hz.
trial	Trial id, so that trial borders are respected when computing velocity and saccades.
options	Named list with method options. See <i>details</i> for further information.

Details

Method options, please refer to Engbert and Kliegl (2003) for details on parameters and the rationale for default values.

- `ek_velocity_threshold` Velocity threshold for saccade detection in standard deviations. Defaults to 6.
- `ek_sd_fun` Function used to compute standard deviation for velocities. Defaults to [sd_via_median_estimator](#), as per formula #2 in Engbert and Kliegl (2003). Can be replaced with `mad`, `sd`, etc.
- `ek_minimal_duration_ms` Minimal duration of a saccade in milliseconds. Defaults to 12.
- `ek_minimal_separation_ms` A minimal required time gap between saccades. Defaults to 12.

Value

logical vector marking samples that belong to saccades

See Also

[extract_saccades](#)

Examples

```
# Do not run this function directly, use extract_saccades() instead
```

method_nh	<i>Extract saccades using an algorithm proposed by Nyström and Holmqvist (2010) doi:10.3758/BRM.42.1.188</i>
-----------	---

Description

Extract saccades using an algorithm proposed by Nyström and Holmqvist (2010) [doi:10.3758/BRM.42.1.188](https://doi.org/10.3758/BRM.42.1.188).

Usage

```
method_nh(x, y, vel, acc, sample_rate, trial, options)
```

Arguments

x	Gaze x coordinate, <code>_arbitrary units_</code> as threshold velocity is computed in units of standard deviation.
y	Gaze y coordinate, <code>_arbitrary units_</code> as threshold velocity is computed in units of standard deviation.
vel	Velocity data.frame with columns x, y, amp.
acc	Acceleration data.frame with columns x, y, amp.
sample_rate	Sample rate in Hz.
trial	Trial id, so that trial borders are respected when computing velocity and saccades.
options	Named list with method options. See <i>details</i> for further information.

Details

Method options, please refer to Nyström and Holmqvist (2010) for details on parameters and the rationale for default values.

- `nh_sg_filter_order` Order of Savitzky-Golay filter. Defaults to 2.
- `nh_max_velocity` Maximal physiologically plausible velocity in °/s. Defaults to 1000.
- `nh_max_acceleration` Maximal physiologically plausible acceleration in °/s². Defaults to 100000.
- `nh_initial_velocity_threshold` Initial velocity threshold in °/s. Defaults to 100.

Value

logical vector marking samples that belong to saccades

See Also

[extract_saccades](#)

Examples

```
# Do not run this function directly, use extract_saccades() instead
```

method_om	<i>Extract saccades using an algorithm proposed by Otero-Millan et al. (2014) Rhrefhttps://doi.org/10.1167/14.2.18doi:10.1167/14.2.18</i>
-----------	---

Description

Extract saccades using an algorithm proposed by Otero-Millan et al. (2014) [doi:10.1167/14.2.18](https://doi.org/10.1167/14.2.18)

Usage

```
method_om(x, y, vel, acc, sample_rate, trial, options)
```

Arguments

x	Gaze x coordinate, <code>_arbitrary units_</code> as threshold velocity is computed in units of standard deviation.
y	Gaze y coordinate, <code>_arbitrary units_</code> as threshold velocity is computed in units of standard deviation.
vel	Velocity data.frame with columns x, y, amp.
acc	Acceleration data.frame with columns x, y, amp.
sample_rate	Sample rate in Hz.
trial	Trial id, so that trial borders are respected when computing velocity and saccades.
options	Named list with method options. See <i>details</i> for further information.

Details

Method options, please refer to Otero-Millan et al. (2014) for details on parameters and the rationale for default values.

- `om_minimal_inter_peak_time_ms` Minimal inter-peak interval in milliseconds. Defaults to 30.
- `om_maximal_peaks_per_second` Maximal allowed number of peaks per second. Defaults to 5.
- `om_velocity_threshold_deg_per_sec` Threshold saccade velocity in °/s. Defaults to 3.
- `om_pca_variance_threshold` Minimal variance explained by retained rotated components. Defaults to 0.05.

Value

logical vector marking samples that belong to saccades

See Also

[extract_saccades](#)

Examples

```
# Do not run this function directly, use extract_saccades() instead
```

monocular_ten_trials	<i>A monocular multi-trial recording</i>
----------------------	--

Description

A monocular recording, 10 trials, sampling rate 500 Hz.

Usage

```
monocular_ten_trials
```

Format

A data frame with 14353 rows and 4 variables:

- trial** Trial index.
- x** X coordinate in degrees of visual angle.
- y** Y coordinate in degrees of visual angle.
- time** Sample time in milliseconds.

option_or_default	<i>Extract value for a named list or use default if key is missing</i>
-------------------	--

Description

Extract value for a named list or use default if key is missing

Usage

```
option_or_default(options, key, default)
```

Arguments

- | | |
|---------|--|
| options | Named list |
| key | String key |
| default | Default value to be returned, if key is missing. |

Value

Value from a list or default value

Examples

```
option_or_default(list("A" = 25), "A", 20)
option_or_default(list("A" = 25), "B", 20)
```

`sd_via_median_estimator`

Compute standard deviation via median estimator.

Description

Compute standard deviation via median estimator. Please refer to formula #2 in Engbert & Kliegl (2003). Falls back on mean estimator, if computed standard deviation is smaller than `.Machine$double.eps`. Raises an error if the results using the mean estimator is still smaller than `.Machine$double.eps`.

Usage

```
sd_via_median_estimator(x, na.rm = FALSE)
```

Arguments

<code>x</code>	Numeric values
<code>na.rm</code>	Whether to exclude NA values, defaults to FALSE.

Value

float

Examples

```
sd_via_median_estimator(rnorm(100))
```

single_trial	<i>A single trial monocular samples recorded at 500 Hz.</i>
--------------	---

Description

A single trial monocular samples recorded at 500 Hz.

Usage

single_trial

Format

A data frame with 1006 rows and 2 variables:

x X coordinate in degrees of visual angle.

y Y coordinate in degrees of visual angle.

single_trial_binocular	<i>A single trial binocular recording.</i>
------------------------	--

Description

A single trial binocular recording sampled at 1000 Hz.

Usage

single_trial_binocular

Format

A data frame with 2000 rows and 6 variables:

trial Trial index.

time_rel Sample time in milliseconds relative to the trial start.

xL X coordinate for the left eye in degrees of visual angle.

xR X coordinate for the right eye in degrees of visual angle.

yL Y coordinate for the left eye in degrees of visual angle.

yR Y coordinate for the right eye in degrees of visual angle.

Index

* datasets

- monocular_ten_trials, [12](#)
- single_trial, [14](#)
- single_trial_binocular, [14](#)

compute_velocity_ek, [2](#)

diff_ek, [3](#), [6](#), [7](#)

diff_nh, [4](#), [6](#), [7](#)

extract_saccades, [5](#), [9](#), [10](#), [12](#)

method_ek, [6](#), [7](#), [8](#)

method_nh, [6](#), [7](#), [10](#)

method_om, [6](#), [7](#), [11](#)

monocular_ten_trials, [12](#)

option_or_default, [12](#)

sd_via_median_estimator, [9](#), [13](#)

single_trial, [14](#)

single_trial_binocular, [14](#)