

Package ‘scanpath’

February 26, 2018

Type Package

Title Tools for Analyzing Spatio-Temporal Patterns in Eye Movements

Version 1.06

Date 2018-01-29

Author Titus von der Malsburg

Maintainer Titus von der Malsburg <malsburg@posteo.de>

Description Provides a similarity measure for scanpaths that is sensitive to spatial and temporal patterns. This measure can be used, for example, to perform cluster analyses of sets of scanpaths or to evaluate models of oculo-motor control.

License GPL-2

LazyLoad yes

Imports ggplot2, dplyr

RoxygenNote 6.0.1

NeedsCompilation yes

R topics documented:

avg.group.dist	2
constant.vars	2
eyemovements	3
find.fixation	4
inverse.gnomonic	5
match.scanpath	6
plot_alignment	7
plot_scanpaths	8
rscasim	9
scanpath	11
scasim	11
which.centroid	14

Index	15
--------------	-----------

avg.group.dist	<i>Mean similarities among sets of scanpaths</i>
----------------	--

Description

Calculates the mean similarities among sub-sets of scanpaths given by a grouping variable. For instance, when the groups are the participants of the experiment, we get a matrix of the average similarities between the scanpaths of the subjects.

Usage

```
avg.group.dist(d, groups)
```

Arguments

d	a symmetric matrix of similarities as calculated by the function scasim or a dist object (see dist).
groups	a vector that assigns each column to a group.

Value

a matrix containing the pair-wise average similarities for all groups.

See Also

[scasim](#), [dist](#)

Examples

```
data(eyemovements)
d <- scasim(eyemovements, duration ~ x + y | trial, 512, 384, 60, 1/30)
s <- constant.vars(eyemovements, trial)$subject
avg.group.dist(d, s)
```

constant.vars	<i>Extract variables that are constant within the levels of a grouping variable.</i>
---------------	--

Description

Given a data frame and a grouping variable, this function returns a new data frame that contains only the variables that are constant within the levels of the grouping variable. The results data frame contains only one line for each level of the grouping variable. This function can be used to extract trial-based information from a data frame containing experimental data with multiple measurements per trial.

Usage

```
constant.vars(data, groups)
```

Arguments

data a data frame.
groups a grouping variable. Either a vector with the same length as the data frame or simply a column in data.

Value

A data frame with one entry for each level of the grouping variable.

Examples

```
data(eyemovements)
head(eyemovements)
constant.vars(eyemovements, trial)
```

eyemovements

Fixational eye movements during reading a sentence

Description

This data set consists of nine trials. One of these trials was recorded in an experiment where the participants had to read single sentences on a screen, one at a time. The other eight trials are modifications of the recorded trial. There are three imaginary participants and for each participant there are three trials: in the first trial the participant read the sentence straight from left to right. In the second trial there was a short regression from the last word to the beginning of the sentence. In the third trial the regression was longer and the eyes check several intermediate words before returning to the end of the sentence. In order to simulate different reading speeds, the recorded fixation durations were modified by adding additional time that was sampled from a normal distribution with different means for each subject.

Usage

```
data(eyemovements)
```

Format

In the data frame each row represents one fixation of the eyes. Fixations are ordered chronologically within trial. The data frame has the following columns:

subject the id of the subject
trial the id of the trial
word the word that was targeted by the fixation
x the x coordinate of a fixation in pixels

y the y coordinate of a fixation in pixels
 duration the duration of a fixation in milliseconds

find.fixation	<i>Find specific fixations</i>
---------------	--------------------------------

Description

Finds fixations embedded in a context that is specified using a regular expression.

Usage

```
find.fixation(l, groups, expr, nth = NA, subpattern = 0)
```

Arguments

l	a vector of single letters or single-digit integers each representing a fixation on a region of interest.
groups	a grouping variable indicating to which scanpath each fixation belongs. The search for matching patterns is performed within each group separately and matches crossing group boundaries are not considered.
expr	a regular expression describing the scanpath pattern of interest.
nth	specifies which match should be returned when there are multiple within a group. The default is to return all matches.
subpattern	the subpattern of interest. If zero, the beginnings of the full patterns will be returned. If n>0, the beginnings of the n-th subpatterns will be returned.

Value

A vector giving the indices of the matches in l.

Examples

```
data(eyemovements)
words <- eyemovements$word
trial <- eyemovements$trial

# Find fixations on word 6 (which is represented by letter "f"):
idx <- find.fixation(words, trial, "6")
eyemovements[idx,]

# Same, but use letters as region identifiers, which is necessary
# when there are more than 10 regions. (Code assumes that regions
# are numbered in ascending order starting from 1.)
w <- letters[match(words, 1:26)]
idx <- find.fixation(w, trial, "f")
eyemovements[idx,]
```

```

# Find fixations on word 6 but only the second match within
# a group (i.e., trial):
idx <- find.fixation(words, trial, "6", nth=2)
eyemovements[idx,]

# Find fixations on word 6 that are followed by fixations on word 7:
idx <- find.fixation(words, trial, "67")
eyemovements[idx,]

# Find those fixations on word 6 that are preceded by fixations on
# word 5:
idx <- find.fixation(words, trial, "5(6)", subpattern=1)
eyemovements[idx,]

# Find the beginning of a second sweep over the sentence:
idx <- find.fixation(words, trial, "[123]+[456]+[78]+([123]+[456]+[78]+)", subpattern=1)
eyemovements[idx,]

```

inverse.gnomonic

Project points from a plane to a sphere.

Description

Projects from points given in plane coordinates to lat-lon, i.e., to the surface of a sphere (inverse gnomonic projection). This can be used to project screen coordinates to coordinates of the visual field of somebody sitting in front of the screen.

Usage

```
inverse.gnomonic(x, y, center_x, center_y, distance, unit_size = 1)
```

Arguments

x	the x-coordinates of the points.
y	the y-coordinates of the points.
center_x	is the x-coordinate of the point that is targeted when the eye looks straight ahead (usually the center of the screen).
center_y	is the y-coordinate of the point that is targeted when the eye looks straight ahead (usually the center of the screen).
distance	is the distance of the plane to the center of the sphere.
unit_size	is ratio of one unit of the plane coordinate system and the unit in which the viewing distance was given. So, one unit in the coordinate system is <code>unit_size * unit</code> of distance.

Value

A data frame containing lat and lon coordinates.

References

See <http://mathworld.wolfram.com/GnomonicProjection.html>.

Examples

```
data(eyemovements)
x <- eyemovements$x
y <- eyemovements$y

latlon <- inverse.gnomonic(x, y, 268, 382, 8, 1/30)

# Before projection:
plot(x, y)
# After:
with(latlon, plot(lon, lat))
```

match.scanpath	<i>Identify scanpath patterns</i>
----------------	-----------------------------------

Description

Identifies sequences of fixations that match a pattern specified using a regular expression.

Usage

```
match.scanpath(l, groups, expr, subpattern = 0)
```

Arguments

<code>l</code>	a vector of single letters or single-digit integers each representing a fixation on a region of interest.
<code>groups</code>	a grouping variable indicating to which scanpath each fixation belongs. The search for matching patterns is performed within each group separately and matches crossing group boundaries are not considered.
<code>expr</code>	a regular expression describing the scanpath pattern of interest.
<code>subpattern</code>	the subpattern of interest. If zero, the complete match will be marked. If $n > 0$, only the n -th subpattern will be marked. See examples.

Value

A vector giving the indices of the matching fixations. Only the first match within each scanpath will be marked.

Examples

```

data(eyemovements)
words <- eyemovements$word
trial <- eyemovements$trial

# Scanpaths from the last word until the end of the trial:
idx <- match.scanpath(words, trial, "8.+")
scanpaths <- eyemovements[idx,]
plot_scanpaths(scanpaths, duration~x|trial)

# Same, but use letters as region identifiers, which is necessary
# when there are more than 10 regions. (Code assumes that regions
# are numbered in ascending order starting from 1.)
w <- letters[match(words, 1:26)]
idx <- match.scanpath(w, trial, "h.+")
scanpaths <- eyemovements[idx,]
plot_scanpaths(scanpaths, duration~x|trial)

# Scanpaths from third word until a fixation on the second word:
idx <- match.scanpath(words, trial, "3.+2")
scanpaths <- eyemovements[idx,]
plot_scanpaths(scanpaths, duration~x|trial)

# Find scanpaths from the third word until the end of the trial but
# only if they contain a fixation on the second word:
idx <- match.scanpath(words, trial, "3.+2.*")
scanpaths <- eyemovements[idx,]
plot_scanpaths(scanpaths, duration~x|trial)

# Find scanpaths spanning words 6, 7, and 8 but only those that
# are directly preceded by a fixation on word 4:
idx <- match.scanpath(words, trial, "4([678]+)", subpattern=1)
scanpaths <- eyemovements[idx,]
plot_scanpaths(scanpaths, duration~word|trial)

```

plot_alignment

This function plots the alignment of fixations in two scanpaths

Description

This function plots the alignment of fixations in two scanpaths

Usage

```

plot_alignment(s, t, a, formula, offset_x = 0, offset_y = 0, nudge_x = 0,
  nudge_y = 0)

```

Arguments

<code>s</code>	is a data frame containing the first scanpath. Each line represents one fixation. The fixations have to be listed in chronological order. Required columns are: x- and y- coordinate of fixations, and fixation durations. See the example data set provided with this package.
<code>t</code>	is the second scanpath. Same format as for <code>s</code> .
<code>a</code>	is the alignment data frame as returned by <code>rscasim</code> .
<code>formula</code>	specifies which columns in the given data frame are should be used. The left-hand side specifies the column that contains the fixation duration. The right-hand side consist of terms that specify x- and y-coordinate of the fixations (in that order). See examples below.
<code>offset_x</code>	offset in the x dimension used for scanpath <code>t</code> . Can be used to reduce clutter.
<code>offset_y</code>	like <code>offset_x</code> but in the y dimension.
<code>nudge_x</code>	x offset used for fixation indices.
<code>nudge_y</code>	y offset used for fixation indices.

Value

A ggplot object. Can be modified before display, for example, by adding axis limits, labels, and titles, etc. See examples below.

Examples

```
data(eyemovements)
s <- subset(eyemovements, trial==1)
t <- subset(eyemovements, trial==9)
alignment <- rscasim(s, t, duration ~ x + y | trial,
                     512, 384, 60, 1/30)
plot_alignment(s, t, alignment, duration ~ x + y | trial, 10, 20)
```

<code>plot_scanpaths</code>	<i>This function plots a set of scanpaths, each scanpath in a separate panel.</i>
-----------------------------	---

Description

This function plots a set of scanpaths, each scanpath in a separate panel.

Usage

```
plot_scanpaths(data, formula, groups = NULL)
```


Arguments

data	is the data frame containing the variables used in the formula.
formula	specifies the format of the plots. For example, <code>dur ~ word trial</code> puts words on the x-axis and time on the y-axis. Each trial is plotted in its own panel. This assumes that words are numbered in ascending order. Another example: <code>dur ~ x + y trial</code> plots a circle for each fixation. The location of the circle is determined by the x- and y-coordinates of the fixation and its size (area) by the duration. Other variables of interest can be used instead of duration, for example pupil size.
groups	a variable used to distinguish different groups of scanpaths by color. Can be used to color-code conditions or participants.

Value

A ggplot object. Can be modified before display, for example, by adding axis limits, labels, and titles, etc. See examples below.

Examples

```
data(eyemovements)
plot_scanpaths(eyemovements, duration ~ word | trial)
plot_scanpaths(eyemovements, duration ~ word | trial, subject)
plot_scanpaths(eyemovements, duration ~ x + y | trial)
plot_scanpaths(eyemovements, duration ~ x + y | trial, subject)
p <- plot_scanpaths(eyemovements, duration ~ x + y | trial, subject)
p + ggplot2::xlim(0, 600) + ggplot2::ylim(284, 484)
```

rscasim

*Calculate the similarity of two scanpaths***Description**

Given two scanpaths, this function calculates their dissimilarity. This is a slow pure-R implementation of the scasim measure written provided for educational purposes. For analyses of larger data sets function [scasim](#) should be used.

Usage

```
rscasim(s, t, formula, center_x, center_y, viewing_distance, unit_size,
        modulator = 0.83)
```

Arguments

s	is a data frame containing the first scanpath. Each line represents one fixation. The fixations have to be listed in chronological order. Required columns are: x- and y- coordinate of fixations, and fixation durations. See the example data set provided with this package.
---	---

<code>t</code>	is the second scanpath. Same format as for <code>s</code> .
<code>formula</code>	specifies which columns in the given data frame are should be used. The left-hand side specifies the column that contains the fixation duration. The right-hand side consist of terms that specify x- and y-coordinate of the fixations (in that order). See examples below.
<code>center_x</code>	is the x-coordinate of the point that is targeted when the eye looks straight ahead (usually the center of the screen).
<code>center_y</code>	is the y-coordinate of the point that is targeted when the eye looks straight ahead (usually the center of the screen).
<code>viewing_distance</code>	is the distance of the eyes from the screen in some arbitrary unit.
<code>unit_size</code>	the ratio of one unit of the coordinate system to one of the units in which the viewing distance was given. Example: If the coordinates are pixels on a screen with 60 dpi and the unit of the distance is inches, <code>unit_size</code> has to be set to 1/60.
<code>modulator</code>	specifies how spatial distances between fixations are assessed. When set to 0, any spatial divergence of two compared fixations is penalized independently equally strong. When set to 1, the scanpaths are compared only with respect to their temporal patterns and spatial distances between matching pairs of fixations are ignored. The default value mimics the exponential drop-off in human visual acuity as we move away from the center from the fovea.

Value

A data frame showing how fixations in the two scanpaths were aligned. Columns `s` and `t` contain the indices of the fixations in scanpaths `s` and `t`, and the column `~cost~` the cost of the edit operation. The sum of the costs is the total dissimilarity of `s` and `t`. If either `s` or `t` contain an NA, that means that a fixation in one scanpath didn't have a matching counterpart in the other scanpath.

Author(s)

Titus von der Malsburg <malsburg@posteo.de>

References

- von der Malsburg, T. and Vasishth, S. (2011). What is the scanpath signature of syntactic reanalysis? *Journal of Memory and Language*, 65(2):109-127.
- von der Malsburg, T., Vasishth, S., and Kliegl, R. (2012). Scanpaths in reading are informative about sentence processing. In Michael Carl, P. B. and Choudhary, K. K., editors, *Proceedings of the First Workshop on Eye-tracking and Natural Language Processing*, pages 37-53, Mumbai, India. The COLING 2012 organizing committee.

See Also

[scasim](#), [plot_alignment](#)

Examples

```
data(eyemovements)
s <- subset(eyemovements, trial==1)
t <- subset(eyemovements, trial==9)
rscasim(s, t, duration ~ x + y | trial,
        512, 384, 60, 1/30)
```

scanpath

*Tools for analyzing scanpaths***Description**

This package provides tools for analyzing spatio-temporal patterns in eye movements, a.k.a. scanpaths.

Author(s)

Titus von der Malsburg <malsburg@posteo.de>

References

von der Malsburg, T. and Vasishth, S. (2011). What is the scanpath signature of syntactic reanalysis? *Journal of Memory and Language*, 65(2):109-127.

See Also

[scasim](#), [avg.group.dist](#), [constant.vars](#), [which.centroid](#)

scasim

*Calculate pair-wise similarities among a set of scanpaths***Description**

Given a set of scanpaths, this function calculates the similarities between all pairs of scanpaths in the set.

Usage

```
scasim(data, formula, center_x, center_y, viewing_distance, unit_size,
       modulator = 0.83, data2 = NULL, formula2 = formula, normalize = FALSE)
```

Arguments

<code>data</code>	is a data frame containing the eye movement data. Each line represents one fixation of the eyes. The fixations of a trial have to be listed en bloc and in chronological order. Required columns are: trial ID (unique in the whole data set), x- and y- coordinate of fixations, and the fixation durations. See the example data set provided with this package.
<code>formula</code>	specifies which columns in the given data frame are relevant. The left-hand side specifies the column that contains the fixation duration. The right-hand side consists of terms that specify x- and y-coordinate of the fixations (in that order) conditioned on trial IDs. See examples below.
<code>center_x</code>	is the x-coordinate of the point that is targeted when the eye looks straight ahead (usually the center of the screen).
<code>center_y</code>	is the y-coordinate of the point that is targeted when the eye looks straight ahead (usually the center of the screen).
<code>viewing_distance</code>	is the distance of the eyes to the screen.
<code>unit_size</code>	is ratio of one unit of the coordinate system and the unit in which the viewing distance was given. So, one unit in the coordinate system is <code>unit_size * unit</code> of distance. Example: If the coordinates are pixels on a screen with 60 dpi and the unit of the distance is inches, <code>unit_size</code> has to be set to 1/60.
<code>modulator</code>	specifies how spatial distances between fixations are assessed. When set to 0, any spatial divergence of two compared scanpaths is penalized independently of its degree. When set to 1, the scanpaths are compared only with respect to their temporal patterns. The default value approximates the sensitivity to spatial distance found in the human visual system.
<code>data2</code>	optionally provides a second set of scanpaths. In this case <code>scasim</code> computes the distances from each scanpath in the first set to each scanpath in the second set. The resulting matrix of distances is therefore not necessarily square.
<code>formula2</code>	specifies the relevant columns in <code>data2</code> in case they are named differently in <code>data2</code> than in <code>data</code> . <code>formula2</code> defaults to <code>formula</code> .
<code>normalize</code>	<p>specifies how each calculated similarity value for scanpaths <code>s</code> and <code>t</code> should be normalized. There are three possible modes:</p> <p><code>fixations</code> normalize by dividing by the total number of fixations in <code>s</code> and <code>t</code>, <code>durations</code> divide by the total duration of <code>s</code> and <code>t</code>, <code>FALSE</code> don't normalize at all.</p> <p>The choice of normalization mode can have a strong effect when fitting maps of scanpaths using <code>isoMDS</code> or <code>cmdscale</code>. Not normalizing at all can yield maps that are difficult to interpret when scanpaths differ markedly in their duration or number of fixations.</p>

Details

When using `scasim`, the dissimilarity of two scanpaths is assessed based on the distances of their fixation targets. If the distances are really small, the dissimilarity will mostly be a function of

just the differences of the fixation durations. The evaluation of distance accounts for the particular way the human visual system magnifies the center of the visual field and the drop in acuity at its periphery.

Value

A matrix containing the pair-wise dissimilarities. Columns and rows correspond to trials. Their order is the same as the order of trials in data.

Author(s)

Titus von der Malsburg <malsburg@posteo.de>

References

von der Malsburg, T. and Vasishth, S. (2011). What is the scanpath signature of syntactic reanalysis? *Journal of Memory and Language*, 65(2):109-127.

von der Malsburg, T., Vasishth, S., and Kliegl, R. (2012). Scanpaths in reading are informative about sentence processing. In Michael Carl, P. B. and Choudhary, K. K., editors, *Proceedings of the First Workshop on Eye-tracking and Natural Language Processing*, pages 37-53, Mumbai, India. The COLING 2012 organizing committee.

See Also

[isoMDS](#) can be applied to the output of `scasim` in order to fitting maps of scanpaths.

Examples

```
data(eyemovements)

# Calculating dissimilarities: when looking straight ahead the gaze
# targets the point with the coordinates (512,384), viewing
# distance (eye to screen) is 60 cm, 1 unit in the data (pixel) is
# 1/30 cm.

dissimilarities <- scasim(eyemovements, duration ~ x + y | trial,
                        512, 384, 60, 1/30)

# Using cmdscale for fitting a map:

map <- cmdscale(dissimilarities)
plot(map, cex=4)
text(map, labels=rownames(map))
```

which.centroid	<i>Find item with minimal average distance to other items</i>
----------------	---

Description

Given a matrix of similarities, this function identifies the item that has the smallest average distance to the other items.

Usage

```
which.centroid(d, select = NULL)
```

Arguments

d	a symmetric matrix of similarities or a dist object.
select	a vector of names of items that should be included in the analysis. These names should correspond to the row and column names of d. Items not listed are ignored completely. The default is to take all items into account.

Details

Although the function is called `which.centroid`, the selected item is not necessarily the centroid in the strict sense. However, it is the item which is closer to the centroid than any other item in the set.

Value

the name of the item that has the minimal average distance to the other items.

Examples

```
data(eyemovements)
d <- scasim(eyemovements, duration ~ x + y | trial, 512, 384, 60, 1/30)
which.centroid(d)
which.centroid(d, c("1", "2", "3"))
```

Index

*Topic **cluster**

rscasim, [9](#)
scanpath, [11](#)
scasim, [11](#)

*Topic **datasets**

eyemovements, [3](#)

*Topic **eye**

rscasim, [9](#)
scanpath, [11](#)
scasim, [11](#)

*Topic **movements**

rscasim, [9](#)
scanpath, [11](#)
scasim, [11](#)

*Topic **package**

scanpath, [11](#)

*Topic **scanpaths**

rscasim, [9](#)
scanpath, [11](#)
scasim, [11](#)

avg.group.dist, [2](#), [11](#)

cmdscale, [12](#)

constant.vars, [2](#), [11](#)

dist, [2](#), [14](#)

eyemovements, [3](#)

find.fixation, [4](#)

inverse.gnomonic, [5](#)

isoMDS, [12](#), [13](#)

match.scanpath, [6](#)

plot_alignment, [7](#), [10](#)

plot_scanpaths, [8](#)

rscasim, [8](#), [9](#)

scanpath, [11](#)

scanpath-package (scanpath), [11](#)

scasim, [2](#), [9–11](#), [11](#)

which.centroid, [11](#), [14](#)