

# Package ‘gglogo’

February 17, 2017

**Title** Geom for Logo Sequence Plots

**Version** 0.1.3

**License** GPL-3

**URL** <https://github.com/heike/gglogo>

**BugReports** <https://github.com/heike/gglogo/issues>

**Depends** R (>= 3.1)

**Imports** ggplot2 (>= 2.0.0), grid, jpeg, plyr, dplyr, magrittr, purrr,  
reshape2, knitr

**Suggests** RColorBrewer

**Description** Visualize sequences in (modified) logo plots. The design choices used by these logo plots allow sequencing data to be more easily analyzed. Because it is integrated into the 'ggplot2' geom framework, these logo plots support native features such as faceting.

**Collate** 'logos.r' 'text.r' 'thin.r' 'data.r' 'alphabet.r' 'position.r'

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**LazyData** true

**NeedsCompilation** no

**Author** Heike Hofmann [aut],  
Eric Hare [aut, cre],  
GGobi Foundation [aut]

**Maintainer** Eric Hare <erichare@iastate.edu>

**Repository** CRAN

**Date/Publication** 2017-02-17 23:36:57

## R topics documented:

aacids . . . . .	2
alphabet . . . . .	3

calcInformation . . . . .	3
createPolygons . . . . .	4
determineOrder . . . . .	5
fortify . . . . .	5
GeomLogo . . . . .	6
getOutline . . . . .	7
ggfortify . . . . .	8
identifyParts . . . . .	9
letterObject . . . . .	10
letterToPolygon . . . . .	10
logo . . . . .	11
mainPlusIslands . . . . .	12
point_line_dist . . . . .	12
position_logo . . . . .	13
sequences . . . . .	14
simplifyPolygon . . . . .	14
simplify_rec . . . . .	15
splitSequence . . . . .	15
StatLogo . . . . .	16

<b>Index</b>	<b>18</b>
--------------	-----------

---

aacids

*properties of amino acids*

---

### Description

properties of amino acids: polarity and water affiliation

### Usage

data(aacids)

### Details

Amino acid properties

---

alphabet	<i>alphabet polygon data</i>
----------	------------------------------

---

**Description**

polygons of upper and lower case letters and digits 0-9

**Usage**

```
data(alphabet)
```

```
data(alphabet_comic)
```

```
data(alphabet_xkcd)
```

```
data(alphabet_braille)
```

**Details**

alphabet polygon data

The alphabet polygon data is based on the font Helvetica. Upper and lower case letters are included as well as digits. Other alphabet data are 'alphabet\_comic' (based on Comic Sans MS), 'alphabet\_xkcd' (based on the xkcd font), and 'alphabet\_braille' (based on Swell Braille)

---

calcInformation	<i>Compute shannon information based on position and treatment</i>
-----------------	--

---

**Description**

Compute shannon information based on position and treatment

**Usage**

```
calcInformation(dframe, trt = NULL, pos, elems, k = 4, weight = NULL,
  method = "shannon")
```

**Arguments**

dframe	data frame of peptide (or any other) sequences and some treatment factors
trt	(vector of) character string(s) of treatment information
pos	character string of position
elems	character string of elements
k	alphabet size: 4 for DNA/RNA sequences, 21 for standard amino acids

weight	number of times each sequence is observed, defaults to 1 in case no weight is given
method	either "shannon" or "frequency" for Shannon information or relative frequency of element by position.

**Value**

extended data frame with additional information of shannon info in bits and each elements contribution to the total information

**Examples**

```
data(sequences)
dm2 <- splitSequence(sequences, "peptide")
dm3 <- calcInformation(dm2, pos="position", trt="class", elems="element", k=21)
# precursor to a logo plot:
library(ggplot2)
# library(biovizBase)
```

---

createPolygons	<i>Create a data set of polygons for a set of letters from a specified font-family</i>
----------------	--

---

**Description**

Create a data set of polygons for a set of letters from a specified fontfamily

**Usage**

```
createPolygons(letters, font, fontsize = 400, dim = c(720, 720),
  scale = FALSE)
```

**Arguments**

letters	set of characters for which polygons are to be created
font	character describing the name of a font - use fonts() from package extrafont to check on available fonts
fontsize	size of letter to be created - larger means higher resolution, but also bigger result sets
dim	dimensions of the box in which the created letter is supposed to fit
scale	scale the values along the y axis to use result in geom_logo

**Value**

data set compatible to work with geom\_logo

**Examples**

```
# check that all letters and digits are nicely shaped:
new_alphabet <- createPolygons(c("f", "g", "W", "*"), font="Garamond")

library(ggplot2)
qplot(x,y, geom="polygon", data=new_alphabet, facets=~group)
```

---

determineOrder	<i>Determine order in which to pass through a set of points</i>
----------------	---

---

**Description**

Greedy algorithm to connect points, with the idea that a point is connected by a line with the two points closest to each, that haven't yet been connected into the shape. Results depend on the starting point.

**Usage**

```
determineOrder(x, y)
```

**Arguments**

x	x coordinates
y	y coordinates

**Value**

vector of indices for ordered traversal along border

---

fortify	<i>Convert image matrix to a data frame</i>
---------	---

---

**Description**

S3 method to create a data frame from a 3d array.

**Usage**

```
fortify(model, data, ...)
```

```
## Default S3 method:
fortify(model, data, ...)
```

**Arguments**

model	array as e.g. returned from readJPEG
data	not used by this method
...	not used by this method

**Value**

data frame

- x x coordinate in pixels
- y y coordinate in pixels (usually negative)
- red number vector in (0,1) describing the amount of red of the pixel in an RGB model
- green number vector in (0,1) describing the amount of green of the pixel in an RGB model
- blue number vector in (0,1) describing the amount of blue of the pixel in an RGB model

---

GeomLogo

*Sequence logo plots.*

---

**Description**

Sequence logo plots.

**Usage**

GeomLogo

```
geom_logo(mapping = NULL, data = NULL, stat = "logo", position = "logo",
  show.legend = NA, inherit.aes = TRUE, width = 0.9, alpha = 0.6,
  na.rm = TRUE, alphabet = NULL, ...)
```

**Arguments**

mapping	The aesthetic mapping, usually constructed with aes or aes_string. Only needs to be set at the layer level if you are overriding the plot defaults.
data	A layer specific dataset - only needed if you want to override the plot defaults,
stat	The statistical transformation to use on the data for this layer,
position	The position adjustment to use for overlapping points on this layer,
show.legend	Whether to show the legend or not
inherit.aes	Whether to inherit the aes or not
width	maximum width of the letters, defaults to 0.9,
alpha	amount of alpha blending used for putting letters on top of rectangle, defaults to 0.25,
na.rm	Whether to remove NAs or not

alphabet	Specifies which alphabet is used in rendering the logo. alphabet can be a dataframe (output from createPolygons), a character specifying a font or NULL. If NULL, the default alphabet set is used (based on Helvetica). Use output from 'createPolygons' to generate alphabet polygons for a different font.
...	other arguments passed on to layer. This can include aesthetics whose values you want to set, not map. See layer for more details.

### Format

An object of class GeomLogo (inherits from Geom, ggproto) of length 6.

### Examples

```
library(ggplot2)
data(sequences)
ggplot(data = ggfortify(sequences, "peptide")) +
  geom_logo(aes(x=position, y=bits, group=element,
               label=element, fill=interaction(Polarity, Water)),
            alpha = 0.6) +
  scale_fill_brewer(palette="Paired") +
  theme(legend.position = "bottom")

ggplot(data = ggfortify(sequences, "peptide", treatment = "class")) +
  geom_logo(aes(x=class, y=bits, group=element,
               label=element, fill=element)) +
  facet_wrap(~position, ncol=18) +
  theme(legend.position = "bottom")

ggplot(data = ggfortify(sequences, "peptide", treatment = "class")) +
  geom_logo(aes(x=position, y=bits, group=element, label=element, fill=element)) +
  facet_wrap(~class, ncol=1) + theme_bw()

ggplot(data = ggfortify(sequences, "peptide", treatment = "class")) +
  geom_logo(aes(x=class, y=bits, group=element,
               label=element, fill=interaction(Polarity, Water))) +
  scale_fill_brewer("Amino-acids properties", palette="Paired") +
  facet_wrap(~position, ncol=18) +
  theme(legend.position="bottom") +
  xlab("") + ylab("Shannon information in bits")
```

---

getOutline

*Determine boundary between foreground and background in an image*

---

### Description

Determine boundary between foreground and background in an image

**Usage**

```
getOutline(imdf, var = "red", threshold = 0.5)
```

**Arguments**

imdf	dataframe describing a pixellated image in x and y. Has to have columns x, y, and var
var	dimension along which foreground and background of a shape in the image are well separated. Usually one of 'red', 'green', or 'blue', but could be extended to any other numerical variable.
threshold	value specifying the cutoff along variable var. Values of var higher than the threshold are considered to belong to the foreground.

**Value**

subset of data frame imdf consisting of just boundary points:

- x x coordinate in pixels
- y y coordinate in pixels (usually negative)
- red number vector in (0,1) describing the amount of red of the pixel in an RGB model
- green number vector in (0,1) describing the amount of green of the pixel in an RGB model
- blue number vector in (0,1) describing the amount of blue of the pixel in an RGB model

---

 ggfortify

---

*Convert sequence data to a format suitable for logo plots*


---

**Description**

Convert sequence data to a format suitable for logo plots

**Usage**

```
ggfortify(data, sequences, treatment = NULL, weight = NULL,
  method = "shannon")
```

**Arguments**

data	data frame with the sequences
sequences	variable containing the sequences
treatment	co-variate(s) used in collecting sequence data
weight	numeric variable of weights
method	either "shannon" or "frequency" for Shannon information or relative frequency of element by position.



**Value**

data frame with position, element and information value

**Examples**

```
library(ggplot2)
data(sequences)

ggplot(data = ggfortify(sequences, "peptide", treatment = "class")) +
  geom_logo(aes(x = class, y = bits, fill = Water, label = element)) +
  facet_wrap(~position)

ggplot(data = ggfortify(sequences, "peptide", treatment = "class")) +
  geom_logo(aes(x = class, y = bits, fill = Polarity, label = element)) +
  facet_wrap(~position, ncol = 18) +
  theme(legend.position = "bottom")
```

---

identifyParts

*Identify different parts of a polygon*

---

**Description**

Identify different parts of a polygon

**Usage**

```
identifyParts(data, tol = NULL)
```

**Arguments**

data	is a data frame with coordinates x, y, and order.
tol	numerical tolerance for minimal distance between groups. If this value is not specified, a tolerance is derived from the marginal frequency break down of observed (squared) distances between consecutive points.

**Value**

data frame group variable is added to the input data

---

letterObject	<i>Convert a text element into an R object</i>
--------------	--

---

**Description**

Convert a text element into an R object

**Usage**

```
letterObject(ch, fontfamily = "Helvetica", fontsize = 576, dim = c(480,
480))
```

**Arguments**

ch	text to be converted, usually just a single letter
fontfamily	R has a few default fonts that are always available, such as e.g. Helvetica, Arial, Courier New, and Garamond. Other fonts might be available depending on the platform used.
fontsize	by default 576. If the resulting string exceeds the boundary of the matrix returned, reduced font size
dim	vector of length two specifying width and height (in pixels) of the temporary jpg file created for the letter. Defaults to 480 x 480 pixels.

**Value**

three dimensional matrix of dimension 480 x 480 x 3 of the pixel values, black background and white letter

**Examples**

```
plot(letterObject("g", fontfamily="Garamond", fontsize=400))
plot(letterObject("q", fontsize=400))
plot(letterObject("B"))
```

---

letterToPolygon	<i>Convert an image file to a polygon</i>
-----------------	---

---

**Description**

Convert an image file to a polygon

**Usage**

```
letterToPolygon(ch, fontfamily = "Helvetica", fontsize = 576, tol = 1,
  dim = c(480, 480), threshold = 0.5, var = "red")
```

**Arguments**

ch	letter
fontfamily	R has a few default fonts that are always available, such as e.g. Helvetica, Arial, Courier New, and Garamond. Other fonts might be available depending on the platform used.
fontsize	by default 576. If the resulting string exceeds the boundary of the matrix returned, reduced font size
tol	tolerance
dim	vector of length two specifying width and height (in pixels) of the temporary jpg file created for the letter. Defaults to 480 x 480 pixels.
threshold	numerical cutoff between 0 and 1
var	one of "red", "green", "blue".

**Examples**

```
library(ggplot2)
letter <- letterToPolygon("R", fontfamily="Helvetica")
qplot(x, y, geom="polygon", data = letter, fill=I("black"), alpha=I(0.8))+
  coord_equal()
```

---

 logo

*Logo plot*


---

**Description**

Simple logo plot of sequences. For more complicated sequence logos, such as with treatment comparisons or subsets see `geom_logo`.

**Usage**

```
logo(sequences)
```

**Arguments**

sequences	vector of text sequences, for which consensus logo is to be shown
-----------	---

**Value**

ggplot2 object for simple sequence

## Examples

```
data(sequences)
library(ggplot2)
library(RColorBrewer)
cols <- rep(brewer.pal(12, name="Paired"),22)
logo(sequences$peptide) + aes(fill=element) + scale_fill_manual(values=cols)
```

---

mainPlusIslands	<i>Set the orientation of a polygon</i>
-----------------	---

---

## Description

Set the orientation of a polygon

## Usage

```
mainPlusIslands(imdf)
```

## Arguments

imdf	dataframe describing a pixellated image in x and y. Has to have columns x, y, and group
------	---

## Value

reordered version of data frame imdf consistent with an assumption of group 1 being the main outline and any other groups being cutouts

---

point_line_dist	<i>Distance between point and line</i>
-----------------	--

---

## Description

Compute distance between point given as (px, py) and line spanned by points (lx1, ly1) and (lx2, ly2). From <http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html>

## Usage

```
point_line_dist(px, py, lx_1, ly_1, lx_2, ly_2)
```

**Arguments**

px	x coordinate of point outside the
py	y coordinate of point
lx_1,	x coordinate of 1st point spanning a line
ly_1,	y coordinate of 1st point spanning a line
lx_2,	x coordinate of 2nd point spanning a line
ly_2,	y coordinate of 2nd point spanning a line

---

position\_logo

*Logo positioning for overlapping objects on top of one another.*


---

**Description**

position\_classic is stacking objects in an ordered fashion from largest to smallest element, position\_logo reverses the classic order and additionally shifts stacks downward to align the largest objects along their vertical minimum, position\_fill additionally standardises each stack to have unit height.

**Usage**

```
position_logo()
```

```
position_classic()
```

**See Also**

See [geom\\_logo](#) for more examples.

**Examples**

```
library(ggplot2)
data(sequences)

# to make the most of comparisons, largest letters ar aligned along their minimum to
# work out the main sequence.
ggplot(data = ggfortify(sequences, "peptide", treatment = "class")) +
  geom_logo(aes(x = class, y = bits, fill = Water, label = element), position="logo") +
  facet_wrap(~position)

# in the classic logo plots letters are stacked in an ordered fasahion on top of each other
ggplot(data = ggfortify(sequences, "peptide", treatment = "class")) +
  geom_logo(aes(x = class, y = bits, fill = Water, label = element), position="classic") +
  facet_wrap(~position)
```

---

sequences	<i>peptide sequence data</i>
-----------	------------------------------

---

**Description**

sequences showing the amino acid usage in the adenylate kinase lid available through biovis redesign contest 2013, see <http://biovis.net/year/2013/info/redesign-contest> published in Wong, B. Nat Methods 7, 889 (2011)

**Usage**

```
data(sequences)
```

**Details**

Sequence data

---

simplifyPolygon	<i>Douglas-Peucker algorithm adjusted fo polyons</i>
-----------------	--

---

**Description**

Implementation of the Douglas-Peucker algorithm for line thinning [http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker\\_algorithm](http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm)

**Usage**

```
simplifyPolygon(points, tol = 1)
```

**Arguments**

points	matrix of x and y points
tol	tolerance

---

simplify_rec	<i>Douglas Pecker algorithm for line thinning</i>
--------------	---

---

**Description**

Implementation of the Douglas-Peucker algorithm for line thinning [http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker\\_algorithm](http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm)

**Usage**

```
simplify_rec(points, tol = 0.01)
```

**Arguments**

points	matrix of x and y points
tol	tolerance

---

splitSequence	<i>Reshape data set according to elements in sequences</i>
---------------	--

---

**Description**

prepare data set for plotting in a logo

**Usage**

```
splitSequence(dframe, sequences)
```

**Arguments**

dframe	data frame of peptide (or any other) sequences and some treatment factors
sequences	character string or index for the character vector of (peptide) sequence

**Examples**

```
data(sequences)  
dm2 <- splitSequence(sequences, "peptide")
```

---

StatLogo

*Calculation of all pieces necessary to plot a logo sequence plot*

---

### Description

Calculation of all pieces necessary to plot a logo sequence plot

### Usage

StatLogo

```
stat_logo(mapping = NULL, data = NULL, geom = "logo", position = "logo",  
  show.legend = NA, inherit.aes = TRUE, width = 0.9, na.rm = TRUE, ...)
```

### Arguments

mapping	The aesthetic mapping, usually constructed with <code>aes</code> or <code>aes_string</code> . Only needs to be set at the layer level if you are overriding the plot defaults.
data	A layer specific dataset - only needed if you want to override the plot defaults,
geom	The geometric object to use display the data,
position	The position adjustment to use for overlapping points on this layer,
show.legend	Whether to show the legend or not
inherit.aes	Whether to inherit the aes or not
width	maximum width of the letters, defaults to 0.9,
na.rm	Whether to remove NAs or not
...	other arguments passed on to layer. This can include aesthetics whose values you want to set, not map. See <code>layer</code> for more details.

### Format

An object of class `StatLogo` (inherits from `Stat`, `ggproto`) of length 4.

### Value

proto object

A proto object

### Examples

```
# See geom_logo for examples  
# Generate data  
data(sequences)  
library(ggplot2)  
  
ggplot(data = ggfortify(sequences, "peptide")) +
```



```
geom_logo(aes(x=position, y=bits, label=element,  
             group=interaction(position, element)),  
          alpha=0.5)
```

# Index

## \*Topic **datasets**

- GeomLogo, 6
  - position\_logo, 13
  - StatLogo, 16
- splitSequence, 15
- stat\_logo (StatLogo), 16
- StatLogo, 16
- aacids, 2
- alphabet, 3
- alphabet\_braille (alphabet), 3
- alphabet\_comic (alphabet), 3
- alphabet\_xkcd (alphabet), 3
- calcInformation, 3
- createPolygons, 4
- determineOrder, 5
- fortify, 5
- geom\_logo, 13
- geom\_logo (GeomLogo), 6
- GeomLogo, 6
- getOutline, 7
- ggfortify, 8
- identifyParts, 9
- letterObject, 10
- letterToPolygon, 10
- logo, 11
- mainPlusIslands, 12
- point\_line\_dist, 12
- position\_classic (position\_logo), 13
- position\_logo, 13
- PositionClassic (position\_logo), 13
- PositionLogo (position\_logo), 13
- sequences, 14
- simplify\_rec, 15
- simplifyPolygon, 14