Package 'phonR'

May 9, 2026

Type Package

Title Tools for Phoneticians and Phonologists

Version 1.0-7

Date 2016-08-22

Author Daniel R. McCloy

Maintainer Daniel R. McCloy <drmccloy@uw.edu>

Depends R (>= 2.10)

Imports splancs, deldir, plotrix, stats, grDevices, graphics

Description Tools for phoneticians and phonologists, including functions for normalization and plotting of vowels.

License GPL-3

URL http://drammock.github.io/phonR/

BugReports https://github.com/drammock/phonR/issues

LazyLoad yes

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2016-08-25 22:37:29

Contents

indoVowels *Formant frequency measurements for Indonesian vowels*

Description

This data set gives F1 and F2 values for five vowels of Standard Indonesian, as spoken by eight speakers (4 male, 4 female), measured from wordlist recordings.

Usage

```
indo
```

Format

A data frame of 1725 rows, with columns `subj`, `gender`, `vowel`, `f1`, and `f2`.

Source

McCloy, D. R. 2014 "Phonetic effects of morphological structure in Indonesian vowel reduction". *Proceedings of Meetings on Acoustics* 12, 060009. http://dx.doi.org/10.1121/1.4870068

Normalize vowel formant frequencies *Normalize formant frequency values using a variety of algorithms*

Description

Functions for transforming vowel formant frequency data measured in Hertz, using one of several normalization schemes commonly used in phonetic and sociolinguistic research. `normVowels` is a convenience function wrapping to the individual `norm[Method]` functions.

Usage

```
normBark(f)
normErb(f)
normLog(f)
normMel(f)
normLobanov(f, group=NULL)
normLogmean(f, group=NULL, exp=FALSE, ...)
normNearey1(f, group=NULL, exp=FALSE, ...)
normNearey2(f, group=NULL, exp=FALSE, ...)
normSharedLogmean(f, group=NULL, exp=FALSE, ...)
normWattFabricius(f, vowel, group=NULL)
normVowels(method, f0=NULL, f1=NULL, f2=NULL, f3=NULL,
           vowel=NULL, group=NULL, ...)
```

Arguments

`f`	Vector or matrix of formant frequencies. For `normNearey`, `f` must be an N-by-4 matrix of frequencies, with column order "f0", "F1", "F2", "F3". For `normWattFabricius`, `f` must be an N-by-2 matrix or data frame of F1 and F2 values. If passing a matrix to `normLogmean`, formants must be grouped within columns, not rows.
`vowel`	Vector or factor of vowel symbols, with `length(vowel)==nrow(f)`. Used only in `normVowels(method="wattfabricius", ...)` or `normWattFabricius(...)`.
`group`	Vector or factor indicating rows of `f` that should be normalized together. This is useful for, e.g., calculating talker-intrinsic normalizations when `group` encodes talker identity.
`exp`	Logical; should the result of the logmeans calculation be passed through the `exp` function before being returned?
`f0`, `f1`, `f2`, `f3`	Separate vectors of formant or fundamental frequency values used in the convenience method `plotVowels`. `f1` and `f2` are required when `method` is `"wattfabricius"`, `"logmean"`, `"shared"`, `"nearey1"`, or `"nearey2"`.
`method`	Specification of the normalization method to use when calling the convenience method `normVowels`. Possible values are "bark", "erb", "lobanov", "log", "logmean", "mel", "shared", and "wattfabricius". "zscore" is an accepted synonym for "lobanov"; "nearey1" is an accepted synonym for "logmean"; "nearey2" is an accepted synonym for "shared"; and "scentroid", is an accepted synonym for "wattfabricius".
`...`	Additional arguments passed to `colMeans` by functions `normLogmean` and `normSharedLogmean` (useful for specifying the value of `na.rm`).

Details

`normLogmean` is a synonym for `normNearey1`, which is also sometimes confusingly called "single logmean". `normSharedLogmean` is a synonym for `normNearey2`. The argument `exp=TRUE` for these functions will yield values that are consistent with the `norm.nearey` implementation, which takes the result of Nearey's original formulae and uses it as the exponent of the base of the natural logarithm (presumably so that the function always yields positive values).

Note that `normErb` returns the "ERB-rate scale" value (i.e., the number of ERBs below the given frequency), not the ERB of the auditory filter centered at the given frequency.

The implementation of the Watt-Fabricius method varies slightly from the formula in Watt & Fabricius (2002), since `normWattFabricius` simply calculates which vowel has the highest mean F1 value and designates it as the low corner of the triangle, rather than asking the user to expressly specify the "TRAP" or "START" vowel. Similarly, `normWattFabricius` simply calculates which vowel has the highest mean F2 value and uses that to calculate the upper left corner, rather than expressly looking for the mean of the "point-vowel" /i/. The upper right corner is, as in the original method, derived from the other two. If the vowels with the highest mean F1 and highest mean F2 are not the same pair of vowels for all members of `group`, `normWattFabricius` returns an error.

Value

Most of the functions return a vector or matrix of the same dimensions as were passed in. The exceptions are `normVowels`, which returns an n-by-m matrix of n data points by m formants with

formants in ascending order with fundamental frequency first (if present), and `normWattFabricius` (or `normVowels` with `method=wattfabricius`), which only and always returns F1 and F2, regardless of whether f0 and F3 were supplied.

Author(s)

McCloy, Daniel <drmccloy@uw.edu>

References

Glasberg, B. R., & Moore, B. C. J. 1990 "Derivation of auditory filter shapes from notched-noise data." *Hearing Research*, 47(1-2), 103-138. http://dx.doi.org/10.1016/0378-5955(90)90170-T

Lobanov, B. M. 1971 "Classification of Russian vowels spoken by different speakers." *The Journal of the Acoustical Society of America*, 49(2), 606-608. http://dx.doi.org/10.1121/1.1912396

McCloy, D. R. 2012 "Normalizing and plotting vowels with the phonR package." *Technical Reports of the UW Linguistic Phonetics Laboratory*. http://dan.mccloy.info/pubs/McCloy2012_phonR.pdf

Nearey, T. M. 1978 "Phonetic feature systems for vowels" (Doctoral dissertation, University of Alberta). Reprinted by the Indiana University Linguistics Club. http://www.ualberta.ca/~tnearey/Nearey1978_compressed.pdf

Stevens, S. S., & Volkmann, J. 1940 "The relation of pitch to frequency: A revised scale." *The American Journal of Psychology*, 53(3), pp. 329-353.

Traunmuller, H. 1990 "Analytical expressions for the tonotopic sensory scale." *The Journal of the Acoustical Society of America*, 88(1), 97-100. http://dx.doi.org/10.1121/1.399849

Watt, D., & Fabricius, A. H. 2002 "Evaluation of a technique for improving the mapping of multiple speakers' vowel spaces in the F1 ~ F2 plane." *Leeds Working Papers in Linguistics and Phonetics*, 9, 159-173.

Examples

```
data(indoVowels)
bark <- with(indo, normBark(cbind(f1, f2)))
erbn <- with(indo, normErb(cbind(f1, f2)))
mel <- with(indo, normMel(cbind(f1, f2)))
lobanov <- with(indo, normLobanov(cbind(f1, f2), group=subj))
lognormed <- with(indo, normLog(cbind(f1, f2)))
# using the convenience wrapper:
bark <- with(indo, normVowels('bark', f1=f1, f2=f2))
wattfab <- with(indo, normVowels('wattfabricius', f1=f1, f2=f2,
                vowel=vowel, group=subj))
```

`Plot vowel formant frequencies`

Plot vowel formant data and a variety of derivative measures.

Description

Generates high-quality plots of provided formant values using either the default onscreen device (X11, Quartz, or Win32) or direct-to-file using built-in R file output methods (PDF, SVG, JPG, PNG, TIFF, or BMP).

Usage

```
plotVowels(f1, f2, vowel=NULL, group=NULL,
           plot.tokens=TRUE, pch.tokens=NULL,
           cex.tokens=NULL, alpha.tokens=NULL,
           plot.means=FALSE, pch.means=NULL,
           cex.means=NULL, alpha.means=NULL,
           hull.line=FALSE, hull.fill=FALSE,
           hull.args=NULL, poly.line=FALSE,
           poly.fill=FALSE, poly.args=NULL,
           poly.order=NA, ellipse.line=FALSE,
           ellipse.fill=FALSE, ellipse.conf=0.6827,
           ellipse.args=NULL, diph.arrows=FALSE,
           diph.args.tokens=NULL, diph.args.means=NULL,
           diph.label.first.only=TRUE,
           diph.mean.timept=1, diph.smooth=FALSE,
           heatmap=FALSE, heatmap.args=NULL,
           heatmap.legend=FALSE, heatmap.legend.args=NULL,
           var.col.by=NULL, var.sty.by=NULL,
           fill.opacity=0.3, label.las=NULL,
           legend.kwd=NULL, legend.args=NULL,
           pretty=FALSE, output='screen', ...)
```

Arguments

`f1`, `f2`	Vector or matrix of formant frequency values. To plot multiple timepoints for each vowel, `f1` and `f2` should be matrices with vowel tokens varying along the rows and timepoints varying across the columns.
`vowel`	Vector of vowel symbols/labels.
`group`	Vector or factor that determines the groups used in calculating vowel means, e.g., a factor indicating "gender", "speaker", "sociolinguistic register", etc.
`plot.tokens`	Logical; should individual vowel tokens be plotted?
`pch.tokens`	Vector of strings or integers; the symbol(s) to use when plotting vowel tokens. Integers are interpreted as standard R `pch` values (see `points`).
`cex.tokens`	Numeric; size of individual vowel points relative to `par("cex")`.

`alpha.tokens` Numeric in range [0,1], indicating opacity of plotted vowel tokens.

`plot.means` Logical; should individual vowel tokens be plotted?

`pch.means` Vector of strings or integers; the symbol(s) to use when plotting vowel means. Integers are interpreted as standard R `pch` values (see `points`).

`cex.means` Size of vowel means relative to `par("cex")`.

`alpha.means` Numeric in range [0,1], indicating opacity of plotted vowel means.

`hull.line` Logical; should a line be drawn tracing the convex hull encompassing all tokens (separately for each level of `group`)?

`hull.fill` Logical; should the convex hull(s) have a color fill?

`hull.args`, `poly.args`, `ellipse.args`
Named list of arguments to be passed to `polygon`. Useful for controlling line width, etc. See "Details" for notes about color handling.

`poly.line` Logical; should a line be drawn tracing the polygon connecting the mean values for each vowel (separately for each level of `group`)?

`poly.fill` Logical; should the polygon(s) connecting mean values for each vowel have a color fill?

`poly.order` Vector or factor indicating the order in which the polygon vertices should be connected. Should match the levels of `factor(vowel)`. If there are values of `vowel` not included in `poly.order`, they will not be connected to the polygon line.

`ellipse.line` Logical; should vowel density ellipses be drawn with an outer line?

`ellipse.fill` Logical; should vowel density ellipses be filled?

`ellipse.conf` Numeric in range (0,1]; the size of the ellipse expressed as a confidence level of the estimate of the true mean (i.e., 0.95 gives a 95% confidence ellipse). The default value (0.6827) corresponds to plus-or-minus one sample standard deviation along the major and minor axes of the bivariate normal density contour.

`diph.arrows` Logical; should the last timepoint of each vowel be marked with an arrowhead?

`diph.args.tokens`, `diph.args.means`
List of named arguments to be passed to `points` and/or `arrows` when plotting tokens or means (ignored if `f1` and `f2` are 1-dimensional). Default is to plot bare lines (`type="l"`) when `pch.tokens` is `NULL`, to draw points and lines overplotted (`type="o"`) when `diph.arrows=FALSE`, and to plot bare lines with the first timepoint of each diphthong plotted when `diph.arrows=TRUE`. When `pretty=TRUE` and `diph.arrows=TRUE`, additional default settings are `length=0.1` and `angle=20`. For `diph.args.means`, `lwd` defaults to `2 * par("lwd")` when `pretty=TRUE`. All of these defaults are overridden by values passed to `diph.args.tokens` or `diph.args.means` in the function call.

`diph.label.first.only`
Logical; if plotting diphthongs, should a symbol or label be drawn only at the first timepoint? Note that if plotting means, the label or symbol may not correspond to the *first* timepoint; it depends on the value of `diph.mean.timept`.

`diph.mean.timept`
A strictly positive integer indicating which timepoint of the diphthongs should be used to calculate means, ellipses, and polygons. For example, if the `f1` and

	f2 arguments are N-by-5 matrices where the 3rd column of each represents the formant measurement at the vowel midpoint (with the other columns providing 2 onglide and 2 offglide measurements), you can plot the means, ellipses, and polygons based on the midpoint measures by specifying diph.mean.timept=3.
diph.smooth	Logical; should a smoothing spline be drawn instead of segments connecting individual timepoints? This feature is under development, unstable, unsupported, and to the extent that it is implemented, it is only implemented for tokens, not means.
heatmap	Logical; should a repulsive force heatmap be drawn?
heatmap.args	Named list of additional arguments passed to repulsiveForceHeatmap. The arguments x, y, and type are passed automatically by plotVowels; heatmap.args is probably most useful for controlling resolution, colormap, and the xform function.
heatmap.legend	Logical; should a legend be drawn showing the color scale used in the repulsive force heatmap?
heatmap.legend.args	Named list of additional arguments passed to repulsiveForceHeatmapLegend. Parameters likely to be user-specified here are x and y for specifying the endpoints of the colorbar, labels for label text at the two ends of the colorbar, and smoothness for the number of color steps to display (limited by the number of levels in the color scale used). See repulsiveForceHeatmapLegend for a more complete description of the available arguments.
var.col.by	Vector or factor indicating the dimension along which to vary color.
var.sty.by	Vector or factor indicating the dimension along which to vary linetype and plotting symbol.
fill.opacity	Number in the range [0, 1] indicating the opacity of color fills for ellipses, hulls, and polygons (if drawn). Does not affect force.heatmap colors, which are specified via force.colmap.
legend.kwd	Keyword indicating legend placement (see legend). If NULL (the default), no legend will be printed.
label.las	Controls the orientation of axis labels relative to the axis line (independently from the axis tick numbers). Possible values are integers 0-4 (see par), or NULL (to use the default value of par("las"), or a value passed to plotVowels as las if available).
legend.args	Named list of additional arguments to be passed to legend, for controlling things like inset, ncol, seg.len, etc.
pretty	Logical; a switch that sets various graphical parameters: mar, las, mgp, xpd, fg, and tcl, as well as arrow parameters like length and angle and plotVowels-specific parameters such as color.palette. It is permissible to set pretty=TRUE and also pass any of these parameters to plotVowels to override the pretty defaults.
output	Graphical device to plot to. Supported values are "screen", "pdf", "svg", "jpg", "tif", "png", "bmp".
...	Other graphical parameters passed to methods; e.g., width, height, units, asp, res, xlim, xlab, main, etc. Two arguments of plot: ann and type, are always silently overridden by plotVowels.

Details

Notes on color handling. If no `col` or `border` arguments are passed to `hull.args`, `poly.args`, or `ellipse.args`, then color is handled as follows: if `pretty=FALSE`, colors default to the values in `palette()`, with opacity for fills set by `fill.opacity`. If `pretty=TRUE`, equally-spaced hues of HCL colors are used instead of `palette()`. If the values passed to `var.col.by` and `vowel` are identical, hull and polygon lines are drawn in black, and fills are drawn black with appropriate `fill.opacity`.

Author(s)

McCloy, Daniel <drmccloy@uw.edu>

See Also

normVowels

Examples

```
data(indoVowels)
with(indo, plotVowels(f1, f2, vowel, group=gender, plot.means=TRUE,
                      pch.means=vowel, ellipse.line=TRUE, poly.line=TRUE,
                      poly.order=c('i','e','a','o','u'), var.col.by=vowel,
                      var.sty.by=gender, pretty=TRUE, alpha.tokens=0.3,
                      cex.means=2))
# simulate some diphthongs
f1delta <- sample(c(-10:-5, 5:15), nrow(indo), replace=TRUE)
f2delta <- sample(c(-15:-10, 20:30), nrow(indo), replace=TRUE)
f1coefs <- matrix(sample(c(2:5), nrow(indo) * 2, replace=TRUE),
                  nrow=nrow(indo))
f2coefs <- matrix(sample(c(3:6), nrow(indo) * 2, replace=TRUE),
                  nrow=nrow(indo))
indo <- within(indo, {
    f1a <- f1  + f1delta * f1coefs[,1]
    f2a <- f2  + f2delta * f2coefs[,1]
    f1b <- f1a + f1delta * f1coefs[,2]
    f2b <- f2a + f2delta * f2coefs[,2]
    })
with(indo, plotVowels(cbind(f1, f1a, f1b), cbind(f2, f2a, f2b), vowel,
                      group=gender, plot.tokens=TRUE, pch.tokens=NA,
                      alpha.tokens=0.3, plot.means=TRUE, pch.means=vowel,
                      var.col.by=vowel, var.sty.by=gender, pretty=TRUE,
                      diph.arrows=TRUE, diph.args.tokens=list(lwd=0.8),
                      diph.args.means=list(lwd=2)))
```

`Repulsive force` *Repulsive force calculations and plots for vowel systems.*

Description

For each point, calculates the sum of inverse squared distances to all points that are not of the same type.

Usage

```
repulsiveForce(x, y, type, xform=log, exclude.inf=TRUE)
repulsiveForceHeatmap(x, y, type=NULL, xform=log,
                      exclude.inf=TRUE, resolution=10,
                      colormap=NULL, fast=FALSE, ...)
repulsiveForceHeatmapLegend(x, y, labels=c("low", "high"),
                            pos=c(1, 3), colormap=NULL,
                            smoothness=50, lend=2, lwd=12, ...)
```

Arguments

`x`, `y`	Numeric vector of x and y values (e.g., F2 and F1 frequencies), or in the case of `repulsiveForceHeatmapLegend`, length-2 vectors specifying the endpoints of the legend colorbar.
`type`	Attribute of the `(x, y)` points used to interpolate values for intermediate points. In typical linguistic usage of this function, `type` would be the vowel identities of the F2 and F1 values passed to `x` and `y` (respectively).
`xform`	A function to apply to the calculated force values before applying the colormap. Default is to use the `log` function.
`exclude.inf`	Logical; should infinite force values be excluded? If true, force for points with identical `x` and `y` values but different values of `type` will be calculated as if the distance between those points was half as long as the smallest non-zero distance in the data (instead of 0).
`resolution`	Number of points to interpolate between each axis unit. Higher resolution yields smoother heatmaps at the cost of increased computational time. NOTE: in typical linguistic usage, an appropriate `resolution` value will depend on the type of units used to plot the formant data (e.g., you will need higher resolution for vowels plotted on the Bark scale to get an equivalently smooth heatmap to one plotted with lower resolution on a Hertz scale.)
`colormap`	Colormap to use when drawing the heatmap and legend (see `color.scale`. Note that although the heatmap may use semi-transparent colors, this transparency does not usually translate well to the `force.legend` colorbar, due to the way that `color.scale.lines` works. In short: the legend colorbar is made of lots of little segments that don't quite touch when plotted, leaving thin gaps in the colorbar where the background shows through. In phonR this has been avoided by adding a square cap to the line ends of each segment in the colorbar, causing neighboring segments to overlap. This overlapping is not noticeable when the `force.colormap` uses fully opaque colors, but usually yields a colorbar that looks opaque even when the `force.colormap` colors are semi-transparent. For better results, generate the `force.colmap` with pale, opaque colors instead of intense colors that are semi-transparent.

fast	Logical; should an interpolation algorithm be used instead of the normal repulsive force function when assigning force values to grid points? If FALSE, the function assigns each grid point a vowel identity based on its nearest neighbor and assigns a color value based on the repulsive force that would occur for a vowel at that location (if it had existed in the dataset). If TRUE, the function performs a Delaunay triangulation of the vowel space using the vowel points in the dataset, and then uses Juan Pineda's triangle filling algorithm to interpolate force values for grid points inside the triangles. This method is fast even at high resolutions, but may appear discontinuous at the edges of adjacent triangles.
labels	Vector of strings (length 2); the labels to write at each end of the force legend color bar. Ignored if heatmap.legend is FALSE.
pos	Vector of integers (length 2); position codes for the colorbar labels. See the pos argument of text for explanation.
smoothness	Number of color steps to use when drawing the legend colorbar. Limited by the number of colors specified in colormap; values larger than length(colormap) will be ignored and use length(colormap) instead.
lend	End-cap style for the individual segments of the colorbar. See par.
lwd	Width of the colorbar. See par.
...	Additional arguments passed to image by repulsiveForceHeatmap, or passed to color.scale.lines by repulsiveForceHeatmapLegend.

Details

Given endpoints x and y, forceHeatmapLegend draws a colorbar legend with smoothness number of steps using the provided colormap (or defaults to grayscale if colormap is NULL.

Value

repulsiveForce returns the sum of the repulsive forces calculated at each point (x,y).

Author(s)

McCloy, Daniel <drmccloy@uw.edu>

References

Liljencrants, J., & Lindblom, B. 1972 "Numerical simulation of vowel quality systems: The role of perceptual contrast". *Language*, 48(4), 839-862. http://www.jstor.org/stable/411991

McCloy, D. R., Wright, R. A., & Souza, P. E. 2014 "Talker versus dialect effects on speech intelligibility: A symmetrical study". *Language and Speech*. http://dx.doi.org/10.1177/0023830914559234

Pineda, J. 1988 "A parallel algorithm for polygon rasterization". *ACM SIGGRAPH Computer Graphics*, 22(4), 17-20. http://dx.doi.org/10.1145/378456.378457

See Also

plotVowels

Examples

```
require(plotrix)
data(indoVowels)
force <- with(indo[indo$subj==indo$subj[1],],
              repulsiveForce(f2, f1, vowel))
colmap <- color.scale(x=0:100, cs1=c(0, 180), cs2=100,
                      cs3=c(25, 100), color.spec='hcl')
with(indo[indo$subj==indo$subj[1],],
    repulsiveForceHeatmap(f2, f1, type=vowel, resolution=10,
                          colormap=colmap, add=FALSE))
xl <- rep(max(range(indo$f2)), 2)
yl <- range(indo$f1) + c(abs(diff(range(indo$f1)) / 2), 0)
repulsiveForceHeatmapLegend(xl, yl, colormap=colmap, useRaster=TRUE)
```

Vowel space area functions

Calculate the area of a vowel space

Description

Calculate the area of an F2 x F1 vowel space, either as the area of a polygon connecting vowel formant means, or the area of a convex hull encompassing all tokens.

Usage

```
convexHullArea(f1, f2, group=NULL)
vowelMeansPolygonArea(f1, f2, vowel, poly.order, group=NULL)
```

Arguments

f1	Numeric vector of first formant frequencies.
f2	Numeric vector or second formant frequencies.
vowel	Vector or factor of vowel identifiers (typically a character vector, though numeric will work).
poly.order	Order in which the polygon vertices should be connected. Should contain each value in vowel once.
group	Vector or factor indicating groupings of points to fit separate convex hulls to. If NULL (the default), a single hull will be generated for all points.

Author(s)

McCloy, Daniel <drmccloy@uw.edu>

See Also

chull, areapl

Examples

```
data(indoVowels)
hull.area <- with(indo, convexHullArea(f1, f2, group=subj))
poly.area <- with(indo, vowelMeansPolygonArea(f1, f2, vowel,
    poly.order=c("i", "e", "a", "o", "u"), group=subj))
```

Index