Package 'pavo'

June 11, 2015

Description A cohesive framework for parsing, analyzing and organizing color from spectral data.

Title Perceptual Analysis, Visualization and Organization of Spectral Color Data in R

Version 0.5-2

License GPL (>= 2)

Date 2015-06-06

URL www.rafaelmaia.net/r-packages/pavo

Depends rgl

Imports geometry, mapproj, rcdd

Collate 'aggplot.R' 'aggspec.R' 'as.rspec.r' 'coldist.R' 'data-bgandilum.R' 'data-sicalis.R' 'data-teal.R' 'data-ttvertex.R' 'data-vissyst.R' 'explorespec.R' 'getspec.R' 'internal.R' 'pavo-package.R' 'peakshape.R' 'plot.rspec.r' 'plotsmooth.R' 'procspec.R' 'projplot.R' 'segclass.R' 'sensmodel.R' 'spec2rgb.R' 'summary.rspec.R' 'summary.tcs.R' 'tcs.R' 'tcsplot.R' 'tcspoints.R' 'tcsvol.R' 'vismodel.R' 'voloverlap.R' 'merge.rspec.R' 'summary.vismodel.R' 'subset.rspec.R' 'irrad2flux.R'

NeedsCompilation no

Author Rafael Maia [aut, cre], Chad Eliason [aut], Pierre-Paul Bitton [aut]

Maintainer Rafael Maia <rm72@zips.uakron.edu>

Repository CRAN

Date/Publication 2015-06-11 01:27:01

R topics documented:

`pavo-package` *pavo: analyzing color data in R*

Description

An R package for the perceptual analysis, visualization and organization of color data

Details

Package:	pavo
Type:	Package
Version:	0.0-1
Date:	2012-10-13
License:	BSD
LazyLoad:	yes

Author(s)

Rafael Maia <rm72@zips.uakron.edu>, Chad Eliason <cme16@zips.uakron.edu>, Pierre-Paul Bitton <bittonp@uwindsor.ca>

Examples

```
#see functions.
```

`aggplot` *Plot aggregated reflectance spectra*

Description

Combines and plots spectra (by taking the average and the standard deviation, for example) according to an index or a vector of identities.

Usage

```
aggplot(rspecdata, by = NULL, FUN.center = mean, FUN.error = sd,
  lcol = NULL, shadecol = NULL, alpha = 0.2, ...)
```

Arguments

`rspecdata`	(required) data frame containing the spectra to be manipulated and plotted.
`by`	(required) either a single value specifying the range of spectra within the data frame to be combined (for example, by = 3 indicates the function will be applied to groups of 3 consecutive columns in the spectra data frame) or a vector containing identifications for the columns in the spectra data frame (in which case the function will be applied to each group of spectra sharing the same identification).
`FUN.center`	the function to be applied to the groups of spectra, calculating a measure of central tendency (defaults to `mean`).
`FUN.error`	the function to be applied to the groups of spectra, calculating a measure of variation (defaults to `sd`).
`lcol`	color of plotted lines indicating central tendency.
`shadecol`	color of shaded areas indicating variance measure.
`alpha`	transparency of the shaded areas.
`...`	additional graphical parameters to be passed to plot.

Value

Plot containing the lines and shaded areas of the groups of spectra.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>, Chad Eliason <cme16@zips.uakron.edu>

References

Montgomerie R (2006) Analyzing colors. In: Hill G, McGraw K (eds) Bird coloration. Harvard University Press, Cambridge, pp 90-147.

Examples

```
## Not run:
data(sicalis)
bysic <- gsub("^ind[0-9].",'',names(sicalis)[-1])
aggplot(sicalis,bysic)
aggplot(sicalis,bysic, shade=spec2rgb(sicalis),lcol=1)
aggplot(sicalis,bysic,lcol=1, FUN.error=function(x)sd(x)/sqrt(length(x)))
## End(Not run)
```

---

aggspec *Aggregate reflectance spectra*

Description

Combines spectra (by taking the average, for example) according to an index or a vector of identities.

Usage

```
aggspec(rspecdata, by = NULL, FUN = mean, trim = TRUE)
```

Arguments

rspecdata	(required) data frame, possibly of class rspec containing the spectra to be manipulated. If it contains a wavelength column containing "wl", that column will be ignored.
by	(required) either a single value specifying the range of spectra within the data frame to be combined (for example, by = 3 indicates the function will be applied to groups of 3 consecutive columns in the spectra data frame); a vector containing identifications for the columns in the spectra data frame (in which case the function will be applied to each group of spectra sharing the same identification); or a list of vectors, e.g., by = list(sex, species).
FUN	the function to be applied to the groups of spectra. (defaults to mean)
trim	logical. if TRUE (default), the function will try to identify and remove numbers at the end of the names of the columns in the new rspec object.

Value

A data frame of class `rspec` containing the spectra after applying the aggregating function.

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

References

Montgomerie R (2006) Analyzing colors. In: Hill G, McGraw K (eds) Bird coloration. Harvard University Press, Cambridge, pp 90-147.

Examples

```
## Not run:
data(teal)
# Average every two spectra
teal.sset1 <- aggspec(teal, by = 2)
plot(teal.sset1)
# Create factor and average spectra by levels 'a' and 'b'
ind <- rep(c('a','b'), times=6)
teal.sset2 <- aggspec(teal, by=ind)
plot(teal.sset2)
## End(Not run)
```

---

as.rspec *Convert data to an rspec object*

Description

Converts data frames or matrices containing spectral data to `rspec` object

Usage

```
as.rspec(object, whichwl = NULL, interp = TRUE, lim = NULL)

is.rspec(object)
```

Arguments

`object`	(required) a data frame or matrix containing spectra to process
`whichwl`	specifies which column contains wavelengths. If NULL (default), function searches for column containing equally spaced numbers and sets it as wavelengths "wl". If no wavelengths are found or `whichwl` is not given, returns arbitrary index values
`interp`	whether to interpolate wavelengths in 1-nm bins (defaults to TRUE)
`lim`	vector specifying wavelength range to interpolate over (e.g., `c(300, 700)`)

Value

an object of class `rspec` for use in further `pavo` functions

a logical value indicating whether the object is of class `rspec`

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

Examples

```
## Not run:

# Generate some fake reflectance data
fakedat <- data.frame(refl1 = rnorm(401), refl2 = rnorm(401), wavelength = c(300:700))
head(fakedat)

# Determine if is rspec object
is.rspec(fakedat)

# Convert to rspec object
fakedat2 <- as.rspec(fakedat)
is.rspec(fakedat2)
head(fakedat2)
## End(Not run)
```

---

bgandilum *Default backgound and illuminant data*

Description

Default backgound and illuminant data

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Endler, J. (1993). The Color of Light in Forests and Its Implications. Ecological Monographs, 63, 1-27.

coldist *Color distances*

Description

Applies the visual models of Vorobyev et al. (1998) to calculate color distances with receptor noise based on relative photoreceptor densities.

Usage

```
coldist(vismodeldata, qcatch = c("Qi", "fi"), vis = c("tetra", "tri", "di"),
  noise = c("neural", "quantum"), subset = NULL, achro = TRUE, n1 = 1,
  n2 = 2, n3 = 2, n4 = 4, v = 0.1)
```

Arguments

vismodeldata	(required) quantum catch color data. Can be either the result from vismodel or independently calculated data (in the form of a data frame with four columns, representing the avian cones).
qcatch	if the object is of class vismodel, such as one generated using pavo, this argument is ignored. If the object is a data frame of quantal catches from another source, this argument is used to specify what type of quantum catch is being used, so that the noise can be calculated accordingly: • Qi: Quantum catch for each photoreceptor (default) • fi: Quantum catch according to Fechner law (the signal of the receptor channel is proportional to the logarithm of the quantum catch)
vis	visual system phenotype to use in the model: • tetra: Tetrachromatic color vision (default) • tri: Trichromatic color vision • di: Dichromatic color vision # • mono: Monochromatic vision
noise	how the noise will be calculated: • neural: noise is proportional to the Weber fraction and is independent of the intensity of the signal received. • quantum: noise is the sum of the neural noise and receptor noise, and is thus proportional to the Weber fraction and inversely proportional to the intensity of the signal received (the quantum catches). Note that the quantum option will only work with objects of class vismodel.
subset	If only some of the comparisons should be returned, a character vector of length 1 or 2 can be provided, indicating which samples are desired. The subset vector must match the labels of the imput samples, but partial matching (and regular expressions) are supported.
achro	logical. If TRUE, last column of the data frame is used to calculate the achromatic contrast, with noise based on the Weber fraction calculated using n4

n1,n2,n3,n4	tetrachromatic photoreceptor densities for u, s, m & l (default to blue tit *Cyanistes caeruleus* densities: 1:2:2:4). If vis does not equal 'tetra', only n1 and n2 (vis='di') or n1, n2 and n3 (vis='tri') are used for chromatic contrast (NOTE: n4 is still the value used for the achromatic contrast.)
v	Noise-to-signal ratio of a single cone (defaults to 0.1, so that under the default densities, the Weber fraction for the large cone will be 0.05, as estimated from behavioral experiment with the Perkin robin, *Leiothrix lutea*)

Value

A data frame containing 4 columns. The first two (patch1, patch2) refer to the two colors being contrasted; dS is the chromatic contrast (delta S, in JNDs) and dL is the achromatic contrast (delta L, in JNDs)

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Vorobyev, M., Osorio, D., Bennett, A., Marshall, N., & Cuthill, I. (1998). Tetrachromacy, oil droplets and bird plumage colours. Journal Of Comparative Physiology A-Neuroethology Sensory Neural And Behavioral Physiology, 183(5), 621-633.

Hart, N. S. (2001). The visual ecology of avian photoreceptors. Progress In Retinal And Eye Research, 20(5), 675-703.

Endler, J. A., & Mielke, P. (2005). Comparing entire colour patterns as birds see them. Biological Journal Of The Linnean Society, 86(4), 405-431.

Examples

```
## Not run:
data(sicalis)
vis.sicalis <- vismodel(sicalis, visual='avg.uv', relative=FALSE)
coldist.sicalis <- coldist(vis.sicalis, vis='tetra')
## End(Not run)
```

---

explorespec *Plot spectral curves*

Description

Plots one or multiple spectral curves in the same graph to rapidly compare groups of spectra.

Usage

```
explorespec(rspecdata, by = 1, scale = c("equal", "free"),
  legpos = "topright", ...)
```

Arguments

rspecdata	(required) a data frame, possibly an object of class rspec that has wavelength range in the first column, named 'wl', and spectral measurements in the remaining columns.
by	number of spectra to include in each graph (defaults to 1)
scale	defines how the y-axis should be scaled. 'free': panels can vary in the range of the y-axis; 'equal': all panels have the y-axis with the same range.
legpos	legend position control. Either a vector containing x and y coordinates or a single keyword from the list: "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center".
...	additional parameters to be passed to plot

Value

Spectral curve plots

Note

Number of plots presented per page depends on the number of graphs produced.

Author(s)

Pierre-Paul Bitton <bittonp@uwindsor.ca>

Examples

```
## Not run:
data(sicalis)
explorespec(sicalis, 3)
explorespec(sicalis, 3, ylim=c(0,100), legpos=c(500,80))
## End(Not run)
```

---

getspec *Import spectra files*

Description

Finds and imports spectra files from a folder. Currently works for reflectance files generated in Ocean Optics SpectraSuite (USB2000, USB4000 and Jaz spectrometers), CRAIC software (after exporting) and Avantes (after exporting).

Usage

```
getspec(where = getwd(), ext = "txt", lim = c(300, 700), decimal = ".",
  subdir = FALSE, subdir.names = FALSE)
```

Arguments

where	(required) folder in which files are located.
ext	file extension to be searched for, without the "." (defaults to "txt").
lim	a vector with two numbers determining the wavelength limits to be considered (defaults to 300 and 700).
decimal	character to be used to identify decimal plates (defaults to ".")
subdir	should subdirectories within the where folder be included in the search? (defaults to FALSE)
subdir.names	should subdirectory path be included in the name of the spectra? (defaults to FALSE)

Value

A data frame, of class rspec, containing individual imported spectral files as columns. Reflectance values are interpolated to the nearest wavelength integer.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Montgomerie R (2006) Analyzing colors. In: Hill G, McGraw K (eds) Bird coloration. Harvard University Press, Cambridge, pp 90-147.

Examples

```
## Not run:
getspec('examplespec/', lim=c(400,900))
getspec('examplespec/', ext='ttt')
## End(Not run)
```

---

irrad2flux *Converts bewtween irradiance and photon (quantum) flux*

Description

Some spectrometers will give illuminant values in units of irradiance (uWatt*cm^-2), but physiological models require illuminants in units of photon (quantum) flux (umol*s^-1*m^-2). The functions irrad2flux and flux2irrad allows for easy conversion of rspec objects between these units.

Usage

```
irrad2flux(rspecdata)

flux2irrad(rspecdata)
```

Arguments

rspecdata	(required) a rspec object containing illuminant values

Value

a converted rspec object

a converted rspec object

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

---

merge.rspec *Merge two rspec objects*

Description

Merges two rspec or data.frame objects into a single rspec object

Usage

```
## S3 method for class 'rspec'
merge(x, y, by = "wl", ...)
```

Arguments

x,y	(required) two data frames (or rspec objects) to merge
by	wavelength column name (defaults to "wl")
...	additional class arguments

Value

an object of class rspec for use with pavo functions

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

See Also

as.rspec, aggspec

Examples

```
## Not run:

# Load and split dataset into 2 sections
data(teal)
teal1 <- teal[, c(1, 3:5)]
teal2 <- teal[, c(1, 2, 6:12)]
teal.mer <- merge(teal1, teal2, by='wl')
head(teal.mer)
par(mfrow=c(1, 2))
plot(teal.mer)
plot(teal)

## End(Not run)
```

---

peakshape *Peak shape descriptors*

Description

Calculates height, location and width of peak at the reflectance midpoint (FWHM). Note: bounds should be set wide enough to incorporate all minima in spectra. Smoothing spectra using procspec is also recommended.

Usage

```
peakshape(rspecdata, select = NULL, lim = c(300, 700), plot = TRUE,
  ask = FALSE, ...)
```

Arguments

rspecdata	(required) a data frame, possibly an object of class rspec, with a column with wavelength data, named 'wl', and the remaining column containing spectra to process.
select	specification of which spectra to plot. Can be a numeric vector or factor (e.g., sex=='male')
lim	a vector specifying the wavelength range to analyze
plot	logical. Should plots indicating calculated parameters be returned? (Defaults to TRUE)
ask	logical, specifies whether user input needed to plot multiple plots when number of spectra to analyze is greater than 1 (defaults to FALSE)
...	additional arguments to be passed to plot

Value

a data frame containing column names (id); peak height (max value, B3), location (hue, H1) and full width at half maximum (FWHM), as well as half widths on left (HWHM.l) and right side of peak (HWHM.r). Incl.min column indicates whether user-defined bounds incorporate the actual minima of the spectra. Function will return a warning if not.

Author(s)

Chad Eliason <cme16@zips.uakron.edu>, Rafael Maia <rm72@zips.uakron.edu>

See Also

procspec

Examples

```
## Not run:
data(teal)
peakshape(teal, select = 3)
peakshape(teal, select = 10)
# Use wavelength bounds to narrow in on peak of interest
peakshape(teal, select = 10, lim=c(400, 550))
## End(Not run)
```

---

plot.rspec *Plot spectra*

Description

Plots reflectance spectra in different arrangements.

Usage

```
## S3 method for class 'rspec'
plot(x, select = NULL, type = c("overlay", "stack",
  "heatmap"), varying = NULL, n = 100, ...)
```

Arguments

x	(required) a data frame, possibly an object of class rspec, with a column with wavelength data, named 'wl', and the remaining column containing spectra to plot.
select	specification of which spectra to plot. Can be a numeric vector or factor (e.g., sex=='male')
type	what type of plot should be drawn. Possibilities are:

- overlay (default) for plotting multiple spectra in a single panel with a common y-axis

- stack for plotting multiple spectra in a vertical arrangement
- heatmap for plotting reflectance values by wavelength and a third variable (varying)

varying	a numeric vector giving values for y-axis in heatplot
n	number of bins with which to interpolate colors and varying for the heatplot.
...	additional arguments passed to plot (or image for 'heatmap').

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

See Also

spec2rgb, image, plot

Examples

```
## Not run:
data(teal)
plot(teal, type = 'overlay')
plot(teal, type = 'stack')
plot(teal, type = 'heatmap')
## End(Not run)
```

---

plotsmooth *Plot loess smoothed curves*

Description

Plots curves with various levels of loess smoothing to help determine what loess parameters are best for the data.

Usage

```
plotsmooth(rspecdata, minsmooth = 0.05, maxsmooth = 0.2, curves = 5,
  specnum = 0, ask = TRUE)
```

Arguments

rspecdata	(required) a data frame, possibly of class rspec, which contains a column containing a wavelength range , named 'wl', and spectra data in remaining columns.
minsmooth	the minimum f value of the loess function to visualize (defaults to 0.05)
maxsmooth	the maximum f value of the loess function to visualize (defaults to 0.20)
curves	the number of curves to display on the same plot (defaults to 5)
specnum	the number of spectral curves, from the data frame, to visualize (defaults to ALL)
ask	logical. if TRUE, asks for user input before changing plot pages

Value

Series of plot with curves processed with varying level of loess smoothing

Author(s)

Pierre-Paul Bitton <bittonp@uwindsor.ca>

Examples

```
## Not run:
data(sicalis)
plotsmooth(sicalis,0.05,0.1,7,6)
## End(Not run)
```

`procspec` *Process spectra*

Description

Applies normalization and/or smoothing to spectra for further analysis or plotting

Usage

```
procspec(rspecdata, opt = c("none", "smooth", "maximum", "minimum", "bin",
  "sum", "center"), fixneg = c("none", "addmin", "zero"), span = 0.25,
  bins = 20)
```

Arguments

`rspecdata`	(required) a data frame, possibly an object of class `rspec`, with a column with wavelength data, named 'wl', and the remaining column containing spectra to process.
`opt`	what type of processing options to apply. User can select multiple options by providing a vector. Possibilites are: • `"none"` does not perform any processing (default). • `"smooth"` applies LOESS smoothing to each spectrum using `loess.smooth`. Optimal smoothing parameter can be assessed by using `plotsmooth`. • `"minimum"` subtracts the minimum from each individual spectra. • `"maxmimum"` divides each spectrum by its maximum value • `"sum"` divides each spectrum by summed values. • `"bin"` bins each spectrum into specified wavelength ranges. User should specify. • `"center"` centers individual spectra by subtracting mean reflectance from all values.
`fixneg`	how to handle negative values. Possibilities are:

- "none" does not perform negative value correction (default).
- "zero" sets all negative values to zero.
- "addmin" adds the absolute value of the maximally negative values of each spectra to the reflectance at all other wavelengths (setting the minimum value to zero, but scaling other values accordingly).

span	sets the smoothing parameter used by loess.smooth
bins	sets the number of equally sized wavelength bins for opt="bin"

Value

A data frame of class rspec with the processed data.

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

References

Cuthill, I., Bennett, A. T. D., Partridge, J. & Maier, E. 1999. Plumage reflectance and the objective assessment of avian sexual dichromatism. The American Naturalist, 153, 183-200.

Montgomerie R. 2006. Analyzing colors. In Hill, G.E, and McGraw, K.J., eds. Bird Coloration. Volume 1 Mechanisms and measuremements. Harvard University Press, Cambridge, Massachusetts.

See Also

loess.smooth

Examples

```
## Not run:
data(teal)
plot(teal, select = 10)
# Smooth data to remove noise
teal.sm <- procspec(teal, opt = 'smooth', span = 0.25)
plot(teal.sm, select = 10)
# Normalize to max of unity
teal.max <- procspec(teal, opt = c('max'), span = 0.25)
plot(teal.max, select = 10)
## End(Not run)
```

projplot *Hue projection plot*

Description

Produces a 2D projection plot of points in a color space

Usage

```
projplot(tcsdata, ...)
```

Arguments

tcsdata	(required) color space coordinates, possibly a result from the tcs function, containing values for the 'h.theta' and 'h.phi' coordinates as columns (labeled as such).
...	additonal parameters to be passed to the plotting of data points.

Value

projplot creates a 2D plot of color points projected from the tetrahedron to its encapsulating sphere, and is ideal to visualize differences in hue.

Note

projplot uses the Mollweide projection, and not the Robinson projection, which has been used in the past. Among other advantages, the Mollweide projection preserves area relationships within latitudes without distortion.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

Endler, J. A., Westcott, D., Madden, J., & Robson, T. (2005). Animal visual systems and the evolution of color patterns: Sensory processing illuminates signal evolution. Evolution, 59(8), 1795-1818.

Examples

```
## Not run:
data(sicalis)
vis.sicalis <- vismodel(sicalis, visual='avg.uv')
tcs.sicalis <- tcs(vis.sicalis)
projplot(tcs.sicalis, pch=16, col=setNames(rep(1:3, 7), rep(c('C','T','B'),7)))
## End(Not run)
```

segclass *Segment classification*

Description

Calculates segment classification measures as defined in Endler (1990)

Usage

```
segclass(rspecdata, range = c(300, 700))
```

Arguments

rspecdata	(required) a data frame, such as objects of class rspec, with a column containing wavelength range, named 'wl' and spectra data in remaining columns.
range	vector of length 2 indicating the lower and upper wavelength bounds used to calculate segments (defaults to 300nm to 700nm).

Value

A data frame with LM and MS segment classification scores.

Author(s)

Pierre-Paul Bitton <bittonp@uwindsor.ca>

References

Endler, J. A. (1990) On the measurement and classification of color in studies of animal color patterns. Biological Journal of the Linnean Society, 41, 315-352.

Examples

```
## Not run:
data(sicalis)
segclass(sicalis)
## End(Not run)
```

sensmodel *Modelling spectral sensitivity*

Description

Models spectral sensitivity (with oil droplets; optional) based on peak cone sensitivity according to the models of Govardovskii et al. (2000) and Hart & Vorobyev (2005).

Usage

```
sensmodel(peaksense, range = c(300, 700), lambdacut = NULL, Bmid = NULL,
  oiltype = NULL, beta = TRUE, om = NULL, integrate = TRUE)
```

Arguments

peaksense	(required) a vector with peak sensitivities for the cones to model
range	a vector of length 2 for the range over which to calculate the spectral sensitivities (defaults to 300nm to 700nm)
lambdacut	a vector of same length as peaksense that lists the cut-off wavelength value for oil droplets. Needs either Bmid or oiltype to also be entered. See Hart and Vorobyev (2005)
Bmid	a vector of same length as peaksense that lists the gradient of line tangent to the absorbance spectrum of the oil droplets. See Hart and Vorobyev (2005)
oiltype	a list of same length as peaksense that lists the oil droplet types (currently accepts only "T", C", "Y", "R", "P") when Bmid is not known. Calculates Bmid based on the regression equations found in Hart ad Vorobyev (2005).
beta	logical. If TRUE the sensitivities will include the beta peak See Govardovskii et al.(2000) (defaults to TRUE).
om	a vector of same length as range1-range2 that contains ocular media transmission data. If included, cone sensitivity will be corrected for ocular media transmission. Currently accepts "bird" using values from Hart et al. (2005), or user-defined values.
integrate	logical. If TRUE, each curve is transformed to have a total area under the curve of 1 (best for visual models; defaults to TRUE).

Value

A data frame of class rspec containing each cone model as a column.

Author(s)

Pierre-Paul Bitton <bittonp@uwindsor.ca>, Chad Eliason <cme16@zips.uakron.edu>

References

Govardovskii VI, Fyhrquist N, Reuter T, Kuzmin DG and Donner K. 2000. In search of the visual pigment template. Visual Neuroscience 17:509-528

Hart NS, and Vorobyev M. 2005. Modelling oil droplet absorption spectra and spectral sensitivities of bird cone photoreceptors. Journal of Comparative Physiology A. 191: 381-392

Hart NS, Partridge JC, Cuthill IC, Bennett AT (2000) Visual pigments, oil droplets, ocular media and cone photoreceptor distribution in two species of passerine bird: the blue tit (Parus caeruleus L) and the blackbird (Turdus merula L). J Comp Physiol A 186:375-387

Examples

```
## Not run:
# Blue tit visual system based on Hart et al (2000)
bluesens <- sensmodel(c(371,448,502,563),beta=F,oiltype = c("T", "C", "Y","R"),om= TRUE)
# Danio aequipinnatus based on Govardovskii et al. (2000)
daniosens <- sensmodel(c(357,411,477,569))
## End(Not run)
```

---

sicalis *Spectral curves from three body regions of Stripe-Tailed Yellow Finch (*Sicalis citrina*) males*

Description

dataset containing reflectance measurements from 3 body parts ("C": crown, "B": breast, "T": throat) from seven male stripe-tailed yellow finches

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

---

spec2rgb *Spectrum to rgb color conversion*

Description

Calculates rgb values from spectra based on human color matching functions

Usage

```
spec2rgb(rspecdata)
```

Arguments

`rspecdata`	(required) a data frame, possibly an object of class `rspec`, with a column with wavelength data, named 'wl', and the remaining column containing spectra to process.

Value

A character vector of class `spec2rgb` consisting of hexadecimal color values for passing to further plotting functions.

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

References

CIE(1932). Commission Internationale de l'Eclairage Proceedings, 1931. Cambridge: Cambridge University Press.

Color matching functions obtained from Colour and Vision Research Laboratory online data repository at http://www.cvrl.org/.

http://www.cs.rit.edu/~ncs/color/t_spectr.html.

Examples

```
## Not run:
data(teal)
spec2rgb(teal)
# Plot data using estimated perceived color
plot(teal, col = spec2rgb(teal), type = 'o')
## End(Not run)
```

`subset.rspec` *Subset rspec, tcs and vismodel objects*

Description

Subsets various object types based on a given vector or grep partial matching of data names

Usage

```
## S3 method for class 'rspec'
subset(x, subset, ...)

## S3 method for class 'tcs'
subset(x, subset, ...)

## S3 method for class 'vismodel'
subset(x, subset, ...)
```

Arguments

x	(required) an object of class rspec, tcs or vismodel containing spectra, visual model output or colourspace data to subset
subset	a string used for partial matching of observations
...	class consistency (ignored).

Value

a subsetted object of the same class as the input object

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

Examples

```
## Not run:

# Load the 'sicalis' dataset
data(sicalis)
# Generate a visual model
vm1 <- vismodel(sicalis)
# Make a tetracolorspace
tcs1 <- tcs(vm1)
# Subset all 'crown' patches (C in file names)
head(subset(sicalis, "B"))
subset(vm1, "B")
subset(tcs1, "B")[, 1:5]
## End(Not run)
```

---

summary.rspec *Colorimetric variables*

Description

Calculates all 23 colorimetric variables reviewed in Montgomerie (2006).

Usage

```
## S3 method for class 'rspec'
summary(object, subset = FALSE, wlmin = NULL,
  wlmax = NULL, ...)
```

Arguments

object	(required) a data frame, possibly an object of class rspec, with a column with wavelength data, named 'wl', and the remaining column containing spectra to process.
subset	Either FALSE (the default), TRUE, or a character vector. If FALSE, all variables calculated are returned. If TRUE, only a subset of the complete ouput (composed of B2, S8 and H1; the variables described in Andersson and Prager 2006) are returned. Finally, a user-specified string of variable names can be used in order to filter and show only those variables.
wlmin,wlmax	minimum and maximum used to define the range of wavelengths used in calculations (default is to use entire range in the rspec object)
...	class consistency (ignored)

Value

A data frame containing either 23 or 5 (subset = TRUE) variables described in Montgomerie (2006) with spectra name as row names. The colorimetric variables calculated by this function are described in Montgomerie (2006) with corrections included in the README CLR file from the May 2008 distribution of the CLR sofware. Authors should reference both this package,Montgomerie (2006), and the original reference(s). Description and notes on the measures:

B1 (Total brightness): Sum of the relative reflectance over the entire spectral range (area under the curve). Frequently used but should be discouraged because values are difficult to compare across studies (B2 is preferred). REF 1-4, 6, 8, 10, 13

B2 (Mean brightness): Mean relative reflectance over the entire spectral range. This is prefered to B1 since values are easier to compare across studies. REF 5, 11

B3 (Intensity): Maximum relative reflectance (Reflectance at wavelength of maximum reflectance). Note that may be sensitive to noise near the peak. REF 1, 7, 9

S1 (Chroma): Relative contribution of a spectral range to the total brightness (B1) S1 is arbitrarily devided in 6 measures of chroma based on the wavelength ranges normally associated with specific hues. The values are calculated using the following ranges: S1U (UV, if applicable): lambda min-400nm; S1v (Violet) lambda min-415nm; S1B (Blue) 400nm-510nm; S1G (Green) 510nm-605nm; S1Y (Yellow) 550nm-625nm; S1R (Red) 605nm-lambda max. REF 3, 4, 6, 11-13

S2 (Spectral saturation): Rmax/Rmin This measure is sensitive to spectral noise. Proper interpretation of this value may be difficult for spectra with multiple peaks in the range of interest. REF 1

S3 (Chroma): Reflectance over the Rmax +- 50nm range divided by B1. Values for peaks within 50nm of either the minimum or maximum range of the data will not be comparable since the area under the curve for the area of interest will not always be based on the same wavelength range. Therefore, S3 should be interpreted with caution for peaks in the UV or Red range. REF 13

S4 (Spectral purity): |bmaxneg| , calculated by approximating the derivative of the spectral curve. As such, it is very sensitive to noise and should only be considered when data is adequately smoothed. NAs are returned for curves which do not, at any range of wavelength, decrease in intensity. Therefore, reflectance curves for brown and red surfaces, for example, should not generate a values. REF 1

S5 (Chroma): Similar in design to segment classification measures (see Montgomerie 2006) for details. REF 8

S6 (Contrast): Rmax - Rmin. Because it uses both Rmin and Rmax, this measure may be sensitive to spectral noise. REF 7, 9

S7 (Spectral saturation): Relative reflectance between the area around the peak with reflectance equal to or larger to half of that of the peak (an approximation to the full-width at half maxima. See Montgomerie (2006) for details). Somewhat sensitive to noise and can be misleading when more than one maxima and/or minima are present. REF 2, 10

S8 (Chroma): (Rmax - Rmin)/B2. Because it uses both Rmin and Rmax, this measure may be sensitive to spectral noise. REF 2, 6

S9 (Carotenoid chroma): (R450 - R700)/R700. Should only be used when the color of the surface is clearly due to carotenoid pigmentation and R450 is lower than R700. Could be sensitive to noise. REF 12

S10 (Peaky chroma): (Rmax - Rmin)/B2 x |bmaxneg|. Should be used with properly smoothed curves. REF 3

H1 (Peak wavelength, hue): Wavelength of maximum reflectance. May be sensitive to noise and may be variable if there is more than one maxima. REF 1, 3-7, 11, 13

H2 (Hue): Wavelength at bmaxneg. Should be calculated using smoothed data. REF 4, 6

H3 (Hue): Wavelength at Rmid. Sensitive to noisy spectra and may be variable if there are more than one maxima and minima. REF 2, 6, 10

H4 (Hue): Similar in design to segment classification measures see Montgomerie (2006) for details. REF 8

H5 (Hue): Wavelength at bmax. Sensitive to noise and may be variable if there is more than one maxima and minima. REF 9

Note

If minimum wavelength is over 400, UV chroma is not computed.

Variables which compute bmax and bmaxneg should be used with caution, for they rely on smoothed curves to remove noise, which would otherwise result in spurious results. Make sure chosen smoothing parameters are adequate.

Smoothing affects only B3, S2, S4, S6, S10, H2, and H5 calculation. All other variables can be reliably extracted using non-smoothed data.

Author(s)

Pierre-Paul Bitton <bittonp@windsor.ca>, Rafael Maia <rm72@zips.uakron.edu>

References

Montgomerie R. 2006. Analyzing colors. In Hill, G.E, and McGraw, K.J., eds. Bird Coloration. Volume 1 Mechanisms and measuremements. Harvard University Press, Cambridge, Massachusetts.

References describing variables:

1- Andersson, S. 1999. Morphology of uv reflectance in a whistling-thrush: Implications for the study of structural colour signalling in birds. Journal of Avian Biology 30:193-204.

2- Andersson, S., J. Ornborg, and M. Andersson. 1998. Ultraviolet sexual dimorphism and assortative mating in blue tits. Proceedings of the Royal Society B 265:445-450.

3- Andersson, S., S. Pryke, J. Ornborg, M. Lawes, and M. Andersson. 2002. Multiple receivers, multiple ornaments, and a trade-off between agonistic and epigamic signaling in a widowbird. American Naturalist 160:683-691.

4- Delhey, K., A. Johnsen, A. Peters, S. Andersson, and B. Kempenaers. 2003. Paternity analysis reveals opposing selection pressures on crown coloration in the blue tit (parus caeruleus). Proceedings of the Royal Society B 270:2057-2063.

5- Keyser, A. and G. Hill. 1999. Condition-dependent variation in the blue-ultraviolet coloration of a structurally based plumage ornament. Proceedings of the Royal Society B 266:771-777.

6- Keyser, A.J. and G. Hill. 2000. Structurally based plumage coloration is an honest signal of quality in male blue grosbeaks. Behavioural Ecology 11:202-209.

7- Ornborg, J., S. Andersson, S. Griffith, and B. Sheldon. 2002. Seasonal changes in a ultraviolet structural colour signal in blue tits, parus caeruleus. Biological Journal of the Linnean Society 76:237-245.

8- Peters, A., A. Denk, K. Delhey, and B. Kempenaers. 2004. Carotenoid-based bill colour as an indicator of immunocompetence and sperm performance in male mallards. Journal of Evolutionary Biology 17:1111-1120.

9- Pryke, S., M. Lawes, and S. Andersson. 2001. Agonistic carotenoid signalling in male red-collared widowbirds: Aggression related to the colour signal of both the territory owner and model intruder. Animal Behaviour 62:695-704.

10- Saks, L., K. Mcgraw, and P. Horak. 2003. How feather colour reflects its carotenoid content. Functional Ecology 17:555-561.

11- Shawkey, M., A. Estes, L. Siefferemán, and G. Hill. 2003. Nanostructure predicts intraspecific variation in ultraviolet-blue plumage colour. Proceedings of the Royal Society B 270:1455-1460.

12- Siefferman, L. and G. Hill. 2005. Uv-blue structural coloration and competition for nestboxes in male eastern bluebirds. Animal Behaviour 69:67-72.

13- Smiseth, P., J. Ornborg, S. Andersson, and T. Amundsen. 2001. Is male plumage reflectance correlated with paternal care in bluethroats? Behavioural Ecology 12:164-170.

Examples

```
## Not run:
data(sicalis)
summary(sicalis)
summary(sicalis, subset = TRUE)
summary(sicalis, subset = c('B1', 'H4'))
## End(Not run)
```

summary.tcs *Tetracolorspace avian visual model summary*

Description

Calculates characteristics of a cloud of points (spectra) in avian tetrahedral color space

Usage

```
## S3 method for class 'tcs'
summary(object, by = NULL, ...)
```

Arguments

object	(required) Results of tcs or Can be either the result from vismodel or independently calculated data (in the form of a data frame with four columns, representing the avian cones).
by	either a single value specifying the range of color points for which summary colorspace variables should be calculated (for example, by = 3 indicates summary will be calculated for groups of 3 consecutive color points (rows) in the quantum catch color data frame) or a vector containing identifications for the rows in the quantum catch color data frame (in which case summaries will be calculated for each group of points sharing the same identification). If by is left blank, the summary statistics are calculated accross all color points in the data.
...	class consistency (ignored)

Value

a data.frame containing summary statistics with groups of points in rows and the following variables in columns:

centroid.u, .s, .m, .l the centroids of usml coordinates of points

c.vol the total volume occupied by the points

colspan.m the mean hue span

colspan.v the variance in hue span

mean.ra mean saturation

max.ra maximum saturation achieved by the group of points

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

Endler, J. A., & Mielke, P. (2005). Comparing entire colour patterns as birds see them. Biological Journal Of The Linnean Society, 86(4), 405-431.

Examples

```
## Not run:
data(sicalis)
vis.sicalis <- vismodel(sicalis, visual='avg.uv')
tcs.sicalis <- tcs(vis.sicalis)
summary(tcs.sicalis, by=rep(c('C','T','B'),7))
## End(Not run)
```

---

summary.vismodel *Visual Model Summary*

Description

Returns the attributes used when calculating a visual model using `vismodel`

Usage

```
## S3 method for class 'vismodel'
summary(object, ...)
```

Arguments

`object`	(required) Results of `vismodel`
`...`	class consistency (ignored)

Value

Returns all attributes chosen when calculating the visual model, as well as the default `data.frame` summary

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Vorobyev, M., Osorio, D., Bennett, A., Marshall, N., & Cuthill, I. (1998). Tetrachromacy, oil droplets and bird plumage colours. Journal Of Comparative Physiology A-Neuroethology Sensory Neural And Behavioral Physiology, 183(5), 621-633.

Hart, N. S. (2001). The visual ecology of avian photoreceptors. Progress In Retinal And Eye Research, 20(5), 675-703.

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

Endler, J. A., & Mielke, P. (2005). Comparing entire colour patterns as birds see them. Biological Journal Of The Linnean Society, 86(4), 405-431.

Examples

```
## Not run:
data(sicalis)
vis.sicalis <- vismodel(sicalis, visual='avg.uv')
summary(vis.sicalis)

## End(Not run)
```

---

tcs *Tetracolorspace avian visual model*

Description

Calculates coordinates and colorimetric variables that represent reflectance spectra in the avian tetrahedral color space.

Usage

```
tcs(vismodeldata)
```

Arguments

`vismodeldata`	(required) quantum catch color data. Can be either the result from `vismodel` or independently calculated data (in the form of a data frame with four columns, representing the avian cones).

Value

A data frame of class `tcs` consisting of the following rows:

`u`, `s`, `m`, `l`: the quantum catch data used to calculate the remaining variables. NOTE: even if visual sistem is of type V-VIS, the output column will be labeled `u`.

`u.r`, `s.r`, `m.r`, `l.r`: relative cone stimulation, for a given hue, as a function of saturation. See Stoddard & Prum (2008) for details.

`x`, `y`, `z`: cartesian coordinates for the points in the tetrahedral color space.

h.theta, h.phi: angles theta and phi, in radians, determining the hue of the color.

r.vec: the r vector (saturation, distance from the achromatic center).

r.max: the maximum r vector achievable for the color's hue.

r.achieved: the relative r distance from the achromatic center, in relation to the maximum distance achievable (r.vec/r.max)

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

Endler, J. A., & Mielke, P. (2005). Comparing entire colour patterns as birds see them. Biological Journal Of The Linnean Society, 86(4), 405-431.

Examples

```
## Not run:
data(sicalis)
vis.sicalis <- vismodel(sicalis, visual='avg.uv')
tcs.sicalis <- tcs(vis.sicalis)
## End(Not run)
```

---

tcsplot *Plot a Tetrahedral Color Space*

Description

tcsplot produces a 3D plot of a tetrahedral color space using OpenGL capabilities

tcspoints plots points in a tetrahedral color space

tcsvol produces a 3D convex hull in tetrahedral color space

Usage

```
tcsplot(tcsdata, size = 0.02, alpha = 1, col = "black",
  vertexsize = 0.02, achrosize = 0.01, achrocol = "grey", lwd = 1,
  lcol = "lightgrey", new = FALSE, hspin = FALSE, vspin = FALSE,
  floor = TRUE, grid = TRUE, fill = TRUE)

tcspoints(tcsdata, size = 0.02, col = "black", alpha = 1)

tcsvol(tcsdata, col = "black", alpha = 0.2, grid.alpha = 1, grid = T,
  fill = T)
```

Arguments

tcsdata	(required) a data frame, possibly a result from the tcs function, containing values for the 'x', 'y' and 'z' coordinates as columns (labeled as such)
size	size of the points in the plot (defaults to 0.02)
alpha	transparency of points (or volume fill in tcsvol)
col	color of the points in the plot (defaults to black)
vertexsize	size of the points at the vertices
achrosize	size of the point in the achromatic center
achrocol	color of the point in the achromatic center
lwd	line width for the edges of the tetrahedron
lcol	line color for the edges of the tetrahedron
new	should a new 3D plot be called (defaults to FALSE)?
hspin	if TRUE, the graphic will spin horizontally (around the 'z' axis)(defaults to FALSE)
vspin	if TRUE, the graphic will spin vertically (around the 'x' axis)(defaults to FALSE)
floor	if TRUE, a reference xy plane is plotted under the tetrahedron (defaults to TRUE)
grid	if TRUE, connects the polygon outlining the volume occupied by points (defaults to TRUE)
fill	if TRUE, fills the volume occupied by points (WARNING: transparency is not saved properly if exported using rgl.postscript)(defaults to TRUE).
grid.alpha	transparecny of the volume polygon grid lines

Value

tcsplot creates a 3D plot using functions of the package rgl, based on openGL capabilities. Plot is interactive and can be manipulated with the mouse (left button: rotate along 'z' axis; right button: rotate along 'x' axis; third button: zoom). tcsvol creates polygon based on points, determining the volume occupied by them in the colorspace. tcspoints adds points to the plot. Points are currently plotted only as spheres to maintain export capabilities.

tcspoints creates 3D points in a tetrahedral color space plot produced by tcsplot using functions of the package rgl, based on openGL capabilities.

tcsvol creates a 3D convex hull within a tcsplot object

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

Endler, J. A., & Mielke, P. (2005). Comparing entire colour patterns as birds see them. Biological Journal Of The Linnean Society, 86(4), 405-431.

See Also

spheres3d,rgl.postscript, rgl.snapshot,rgl.material

Examples

```
## Not run:
# For plotting
data(sicalis)
vis.sicalis <- vismodel(sicalis, visual='avg.uv')
tcs.sicalis <- tcs(vis.sicalis)
tcsplot(tcs.sicalis, size=0.005)
rgl.postscript('testplot.pdf',fmt='pdf')
rgl.snapshot('testplot.png')

# For adding points
patch <- rep(c('C','T','B'),7)
tcs.crown <- subset(tcs.sicalis, 'C')
tcs.breast <- subset(tcs.sicalis, 'B')
tcsplot(tcs.crown, col='blue')
tcspoints(tcs.breast, col='red')

# For plotting convex hull
tcsplot(tcs.sicalis, col='blue', size=.005)
tcsvol(tcs.sicalis)
## End(Not run)
```

---

teal *Angle-resolved reflectance data for the iridescent wing patch of a male green-winged teal (Anas carolinensis)*

Description

dataset containing reflectance measurements from the wing patch of a single male at different incident angles (15-75 degrees in 5-degree increments).

Author(s)

Chad Eliason <cme16@zips.uakron.edu>

ttvertex *vertex for the tetrahedral color space*

Description

internal data for plotting devices.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

vismodel *Visual Models*

Description

Applies the visual models of Vorobyev et al. (1998) to calculate quantum catches at each photoreceptor. Relative values may also be obtained, in which case the model reduces to the color space as described in Endler & Mielke (2005) and Stoddard & Prum (2008).

Usage

```
vismodel(rspecdata, qcatch = c("Qi", "fi"), visual = c("avg.uv", "avg.v",
  "bluetit", "star", "pfowl"), achromatic = c("bt.dc", "ch.dc", "st.dc", "ml",
  "none"), illum = c("ideal", "bluesky", "D65", "forestshade"),
  vonkries = F, scale = 1, bkg = "ideal", relative = TRUE)
```

Arguments

rspecdata	(required) a data frame, possibly an object of class rspec that has wavelength range in the first column, named 'wl', and spectral measurements in the remaining columns.
qcatch	Which quantal catch metric to return. Options are: • Qi: Quantum catch for each photoreceptor • fi: Quantum catch according to Fechner law (the signal of the receptor channel is proportional to the logarithm of the quantum catch)
visual	the visual system to be used. Options are:

- a data frame such as one produced containing by `sensmodel`, containing sensitivity for the user-defined visual system. The data frame must contain a `'wl'` column with the range of wavelengths included, and the sensitivity for each other cone as a column
- `avg.uv`: average avian UV system
- `avg.v`: average avian V system
- `bluetit`: Blue tit *Cyanistes caeruleus* visual system
- `star`: Starling *Sturnus vulgaris* visual system
- `pfowl`: Peafowl *Pavo cristatus* visual system

`achromatic` the sensitivity data to be used to calculate luminance (achromatic) cone stimulation. Either a vector containing the sensitivity for a single receptor, or one of the options:

- `bt.dc`: Blue tit *Cyanistes caeruleus* double cone
- `ch.dc`: Chicken *Gallus gallus* double cone
- `st.dc`: Starling *Sturnus vulgaris* double cone
- `ml`: sum of the two longest-wavelength cones
- `none`

`illum` either a vector containing the illuminant, or one of the options:

- `ideal`: homogeneous illuminance of 1 accross wavelengths (default)
- `'bluesky'`
- `'D65'`: standard daylight
- `'forestshade'`

`vonkries` logical. Should the von Kries color correction transformation be applied? (defaults to `FALSE`)

`scale` a value by which the illuminant will be multiplied. Useful for when the illuminant is a relative value (i.e. transformed to a maximum of 1 or to a percentage), and does not correspond to quantum flux units ($umol*s^-1*m^-2$). Useful values are, for example, 500 (for dim light) and 10000 (for bright illumination). Note that if `vonkries=TRUE` this transformation has no effect.

`bkg` either a vector containing the background spectra, or an ideal (white) background is used (Default assumes an idealized homogeneous background).

`relative` should relative quantum catches be returned (i.e. is it a color space model? Defaults to `TRUE`).

Value

An object of class `vismodel` containing the photon catches for each of the photoreceptors considered. Information on the parameters used in the calculation are also stored and can be called using the `summary.vismodel` function.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Vorobyev, M., Osorio, D., Bennett, A., Marshall, N., & Cuthill, I. (1998). Tetrachromacy, oil droplets and bird plumage colours. Journal Of Comparative Physiology A-Neuroethology Sensory Neural And Behavioral Physiology, 183(5), 621-633.

Hart, N. S. (2001). The visual ecology of avian photoreceptors. Progress In Retinal And Eye Research, 20(5), 675-703.

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

Endler, J. A., & Mielke, P. (2005). Comparing entire colour patterns as birds see them. Biological Journal Of The Linnean Society, 86(4), 405-431.

Examples

```
## Not run:
data(sicalis)
vis.sicalis <- vismodel(sicalis, visual='avg.uv')
tcs.sicalis <- tcs(vis.sicalis)
## End(Not run)
```

---

vissyst *Animal visual systems data*

Description

Internal data for visual model calculations.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

Endler, J. A., & Mielke, P. (2005). Comparing entire colour patterns as birds see them. Biological Journal Of The Linnean Society, 86(4), 405-431.

voloverlap *Color volume overlap*

Description

Calculates the overlap between the volumes defined by two sets of points in cartesian space

Usage

```
voloverlap(tcsres1, tcsres2, plot = FALSE, col = c("blue", "red",
  "darkgrey"), new = TRUE, montecarlo = FALSE, nsamp = 1000,
  psize = 0.001)
```

Arguments

tcsres1,tcsres2	(required) data frame, possibly a result from the tcs function, containing values for the 'x', 'y' and 'z' coordinates as columns (labeled as such)
plot	logical. Should the volumes and points be plotted? (defaults to FALSE)
col	a vector of length 3 with the colors for (in order) the first volume, the second volume, and the overlap.
new	logical. Should a new plot window be called? If FALSE, volumes and their overlap are plotted over the current plot (defaults to TRUE).
montecarlo	logical. If TRUE, Monte Carlo simulation is used instead of exact solution (not recommended; defaults to FALSE)
nsamp	if montecarlo=TRUE, determines the number of points to be sampled.
psize	if montecarlo=TRUE and plot=TRUE, sets the size to plot the points used in the Monte Carlo simulation.

Value

Calculates the overlap between the volumes defined by two set of points in colorspace. The volume from the overlap is then given relative to:

- vsmallest the volume of the overlap divided by the smallest of that defined by the the two input sets of color points. Thus, if one of the volumes is entirely contained within the other, this overlap will be vsmallest = 1.
- vboth the volume of the overlap divided by the combined volume of both input sets of color points.

The Monte Carlo solution is available mostly for legacy and benchmarking, and is not recommended (see notes). If used, the output will be different:

- s_in1, s_in2 the number of sampled points that fall within each of the volumes individually
- s_inboth the number of sampled points that fall within both volumes

- s_ineither the number of points that fall within either of the volumes
- psmallest the proportion of points that fall within both volumes divided by the number of points that fall within the smallest volume
- pboth the proportion of points that fall within both volumes divided by the total number of points that fall within both volumes

If the Monte Carlo solution is used, a number of points much greater than the default should be considered (Stoddard & Stevens(2011) use around 750,000 points, but more or fewer might be required depending on the degree of overlap.)

Note

Stoddard & Stevens (2011) originally obtained the volume overlap through Monte Carlo simulations of points within the range of the volumes, and obtaining the frequency of simulated values that fall inside the volumes defined by both sets of color points.

Here we present an exact solution based on finding common vertices to both volumes and calculating its volume. However, we also the Monte Carlo solution is available through the montecarlo=TRUE option.

Stoddard & Stevens (2011) also return the value of the overlap relative to one of the volumes (in that case, the host species). However, for other applications this value may not be what one expects to obtain if (1) the two volumes differ considerably in size, or (2) one of the volumes is entirely contained within the other. For this reason, we also report the volume relative to the union of the two input volumes, which may be more adequate in most cases.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>, with code from Sebastien Villeger

References

Stoddard, M. C., & Prum, R. O. (2008). Evolution of avian plumage color in a tetrahedral color space: A phylogenetic analysis of new world buntings. The American Naturalist, 171(6), 755-776.

Stoddard, M. C., & Stevens, M. (2011). Avian vision and the evolution of egg color mimicry in the common cuckoo. Evolution, 65(7), 2004-2013.

Villeger, S., Novack-Gottshall, P. M., & Mouillot, D. (2011). The multidimensionality of the niche reveals functional diversity changes in benthic marine biotas across geological time. Ecology Letters, 14(6), 561-568.

Examples

```
## Not run:
data(sicalis)
tcs.sicalis.C <- subset(tcs(vismodel(sicalis)), 'C')
tcs.sicalis.T <- subset(tcs(vismodel(sicalis)), 'T')
tcs.sicalis.B <- subset(tcs(vismodel(sicalis)), 'B')
voloverlap(tcs.sicalis.T,tcs.sicalis.B)
voloverlap(tcs.sicalis.T,tcs.sicalis.C, plot=T)
voloverlap(tcs.sicalis.T,tcs.sicalis.C, plot=T, col=1:3)
## End(Not run)
```

Index