# Package ‘colourvision’

October 12, 2022

**Type** Package  
**Title** Colour Vision Models  
**Version** 2.0.4  
**Date** 2021-07-31  
**Author** Felipe M. Gawryszewski  
**Maintainer** Felipe Malheiros Gawrysewski <f.gawry@gmail.com>  

**Description**  

**License** GPL-2  
**Imports** graphics, stats, utils, Matrix  
**Suggests** testthat, knitr, rmarkdown, corrplot, rgl  
**VignetteBuilder** knitr  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2021-08-01 23:10:11 UTC

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>colourvision-package</td>
<td>2</td>
</tr>
<tr>
<td>bee</td>
<td>4</td>
</tr>
<tr>
<td>colour_space</td>
<td>5</td>
</tr>
<tr>
<td>CTTKhexagon</td>
<td>6</td>
</tr>
<tr>
<td>CTTKhexagon3D</td>
<td>8</td>
</tr>
<tr>
<td>CTTKmodel</td>
<td>9</td>
</tr>
<tr>
<td>D65</td>
<td>12</td>
</tr>
<tr>
<td>deltaS</td>
<td>13</td>
</tr>
<tr>
<td>EMline</td>
<td>14</td>
</tr>
</tbody>
</table>
Colour Vision Models

Description


Details

The DESCRIPTION file:

Package: colourvision
Type: Package
Title: Colour Vision Models
Version: 2.0.4
Date: 2021-07-31
Author: Felipe M. Gawryszewski
Maintainer: Felipe Malheiros Gawrysewski <f.gawry@gmail.com>
Description: Colour vision models, colour spaces and colour thresholds. Provides flexibility to build user-defined colour vision models for n number of photoreceptor types.
License: GPL-2
Imports: graphics, stats, utils, Matrix
Suggests: testthat, knitr, rmarkdown, corrplot, rgl
VignetteBuilder: knitr

Index of help topics:

CTTKhexagon Chittka (1992) colour hexagon
CTTKhexagon3D Chittka (1992) colour space for tetrachromatic animals.
CTTKmodel Chittka (1992) colour vision model
D65 CIE Standard Illuminant D65 in quantum flux (umol/m2/s)
EMline Endler and Mielke (2005) 1-D colour space
EMmodel Endler and Mielke (2005) colour vision model
EMtetrahedron Endler and Mielke (2005) tetrahedron colour space
EMtriangle Endler and Mielke (2005) triangle colour space
GENmodel N-dimensional generic colour vision model
GENplot Generic model colour space 2D and 1D plot
GENplot3d Generic model colour space 3D plot
Q Total photon capture
Qr Photoreceptor relative quantum catch
RNLmodel Receptor Noise Limited Models (Vorobyev & Osorio 1998)
RNLplot Receptor noise limited model 2D and 1D plot
RNLplot3d Receptor noise limited model 3D plot
Rb Brazilian savannah background reflectance spectrum.
bee Honeybee photoreceptors
colour_space N-dimensional colour spaces
colourvision-package Colour Vision Models
deltaS Chromaticity distances
denstoytoflux Irradiance from energy to quantum units.
logistic Logistic curve
noise_e Receptor noise
photoreceptor Photoreceptor sensitivity spectra.
plot.colourvision Plot colour vision models into chromaticity diagrams
plot3d.colourvision Plot colour vision models into 3D chromaticity diagrams.
radarplot Radar plot
spec.denoise Smooth function for reflectance spectra.

Author(s)

Felipe M. Gawryszewski
Maintainer: Felipe Malheiros Gawrysewski <f.gawry@gmail.com>

References


Examples

### Honeybee photoreceptor sensitivity curves

data("bee")

### Grey background:
## with 10 percent. reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(10, length(300:700)))

### Read CIE D65 standard illuminant already converted to quantum flux:
data("D65")

### Reflectance data
## with a sigmoid spectrum and midpoint at 500nm and 550 nm
R1<-logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R2<-logistic(x=seq(300,700,1), x0=550, L=50, k=0.04)
R<-cbind(R1, R2[,2])

### Run colour vision model:
model<-CTTKmodel(photo="tri", R=R, I=D65, Rb=Rb, C=bee)

### plot data in the colour space
plot(model)

---

**bee**

*Honeybee photoreceptors*

---

**Description**

Honeybee (Apis mellifera) photoreceptor sensitivity curves.

**Usage**

data("bee")

**Format**

A data frame with 401 observations on the following 4 variables.

- **Wavelength** a numeric vector
- **UV** a numeric vector
- **Blue** a numeric vector
- **Green** a numeric vector
**Details**

Original data were interpolated to 1nm intervals from 300 to 700nm.

**Source**


**Examples**

```r
data("bee")
plot(bee[,2]~bee[,1], col = "violet", type="l", xlab="Wavelength(nm)", ylab= "Absorbance")
lines(bee[,3]~bee[,1], col = "blue", type="l")
lines(bee[,4]~bee[,1], col = "green", type='l')
```

---

**colour_space**

*N-dimensional colour spaces*

**Description**

Generates a colour space based on any number of photoreceptor types and finds a colour locus for a given photoreceptor output.

**Usage**

```r
colour_space(n, type="length", length=NA, edge=NA,
             q=rep(1,n), recep.noise=FALSE, e=NA)
```

**Arguments**

- **n**: Number of photoreceptor types. Function accepts any number of photoreceptor types >=2. For instance, trichromatic: n=3; tetrachromatic: n=4, etc.
- **type**: Whether the colour space should be built with a fixed vector length (type="length"), or a fixed distance between vertices (type="edge"; e.g. Pike 2012 and Renoult et al. 2015).
- **length**: Vector length used to construct the colour space. Used when type="length". Typically length=1.
- **edge**: Edge length used to construct the colour space. Used when type="edge".
- **q**: Photoreceptor output values
- **recep.noise**: Whether receptor noise should be used to calculate colour locus coordinates.
- **e**: Vector representing photoreceptor noises. Used when recep.noise=TRUE

**Details**

This function is used internally in colour vision models.
Value

A list with the following dimensions:

- coordinates: Colour locus of photoreceptor output values \( q \) in the colour space
- vector_matrix: Matrix of column vectors, each representing one photoreceptor type

Author(s)

Felipe M. Gawryszewski &lt;f.gawry@gmail.com&gt;

References


See Also

- Q, Qr
- CTTKmodel, EMmodel, RNLmodel, GENmodel

Examples

```r
# A trichromatic colour space based on Endler and Mielke (2005)
tri <- colour_space(n=3, length=0.75, q=c(0.5, 0.2, 0.3))

# showing:
#(1) Limits of the colour space (triangle)
plot(0, ylim=c(-1,1), xlim=c(-1,1), asp=1, ylab="X2", xlab="X1", type="n")
polygon(x=tri$vector_matrix[,1], y=tri$vector_matrix[,2], lty=2)

#(2) Vectors (length=0.75) used to build the colour space (arrows)
arrows(x0=0, y0=0, x1=tri$vector_matrix[1,1], y1=tri$vector_matrix[2,1], col="red")
arrows(x0=0, y0=0, x1=tri$vector_matrix[1,2], y1=tri$vector_matrix[2,2], col="red")
arrows(x0=0, y0=0, x1=tri$vector_matrix[1,3], y1=tri$vector_matrix[2,3], col="red")

#(3) Colour loci of given photoreceptor outputs
points(x=tri$coordinates[[1]], y=tri$coordinates[[2]], pch=21, col="blue", bg="blue")
```

CTTKhexagon

**Chittka (1992) colour hexagon**

Description

Plots Chittka (1992) colour hexagon for trichromatic animals and a line plot for dichromatic animals.
CTTKhexagon

Usage

CTTKhexagon(x, y, photo=3, 
    vnames=c(expression(E[1]), expression(E[2]), expression(E[3])), 
    pch=16, bty="n", yaxt="n", xaxt="n", col="black", 
    xlim="auto", ylim="auto", asp=1, ann=FALSE, 
    axes=FALSE, vectors=FALSE, ...)

Arguments

x x-coordinate
y y-coordinate. y=0 when photo=2
photo Number of photoreceptor types. photo=3 for a trichromatic animal, and photo=2
    for a dichromatic animal.
vnames Vector names.
pch see par function.
bty see par function.
yaxt see par function.
xaxt see par function.
col see par function.
xlim see plot function. Default calculates xlim automatically.
ylim see plot function. Default calculates ylim automatically.
aspect see plot function.
am see par function.
axes see plot.default function.
vectors Whether vectors representing direction of photoreceptor outputs should be plotted.
... Other arguments passed to plot function.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References

Chittka, L. 1992. The colour hexagon: a chromaticity diagram based on photoreceptor excitations

See Also

CTTKmodel, CTTKhexagon3D
Examples

```r
## Honeybee photoreceptor sensitivity curves
data("bee")

## Grey background:
## with 7 percent reflectance from 300 to 700 nm:
Rb <- data.frame(300:700, rep(7, length(300:700)))

## Read CIE D65 standard illuminant already converted to quantum flux:
data("D65")

## Reflectance data
## with a sigmoid spectrum and midpoint at 500 nm and 550 nm
R1 <- logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R2 <- logistic(x=seq(300,700,1), x0=550, L=50, k=0.04)
R <- cbind(R1, R2[,2])

## Run colour vision model:
model <- CTTKmodel(photo="tri", R=R, I=D65, Rb=Rb, C=bee)

## Plot data in the colour space
CTTKhexagon(x=model[,"X1"], y=model[,"X2")
```

---

**CTTKhexagon3D**

Chittka (1992) colour space for tetrachromatic animals.

**Description**

Plots a hexagonal trapezohedron representing Chittka (1992) colour space for tetrachromatic animals (Thery and Casas, 2002).

**Usage**

```r
CTTKhexagon3D(x, y, z, s.col = "red", f.col = "black", vnames = c("E1","E2","E3","E4"), type = "p", radius = 0.01, add = F, xlab = "", ylab = "", zlab = "", box = F, axes = F, ylim = c(-1, 1), xlim = c(-1, 1), zlim = c(-1,1), aspect = T, vectors=F, ...)
```

**Arguments**

- `x`: x coordinate of points to be plotted
- `y`: y coordinate of points to be plotted
- `z`: z coordinate of points to be plotted
- `s.col`: Colour to be used for plotted items. See plot3d in the rgl package.
- `f.col`: Colour of tetrahedron lines. See plot3d in rgl package.
CTTKmodel

vnames Vector names.
type see plot3d function in the rgl package.
radius see plot3d function in the rgl package.
add see plot3d function in the rgl package.
xlab see plot3d function in the rgl package.
ylab see plot3d function in the rgl package.
zlab see plot3d function in the rgl package.
box see plot3d function in the rgl package.
axes see plot3d function in the rgl package.
ylim see plot3d function in the rgl package.
xlim see plot3d function in the rgl package.
zlim see plot3d function in the rgl package.
aspect see plot3d function in the rgl package.
vectors Whether vectors representing direction of photoreceptor outputs should be plotted.
... Other arguments passed to function plot3d in the rgl package.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References


See Also

CTTKmodel, CTTKhexagon

CTTKmodel | Chittka (1992) colour vision model

Description

Chittka (1992) colour hexagon extended to animals with any number of photoreceptors types.

Usage

CTTKmodel(photo=ncol(C)-1, R, I, Rb, C, interpolate=TRUE, nm=seq(300,700,1))
Arguments

- **photo**: Number of photoreceptor types. Model accepts any number of photoreceptor types (>=2). For instance, dichromatic: photo=2; trichromatic: photo=3; tetra-chromatic: photo=4, etc. Default gets number of photoreceptor types from C argument.

- **R**: Reflectance of observed objects. A data frame with first column corresponding to wavelength values and following columns with reflectance values. R must be in the same scale as Rb (percentage or proportion).

- **I**: Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.

- **Rb**: Background reflectance. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values. Rb must be in the same scale as R (percentage or proportion).

- **C**: Photoreceptor sensitivity curves, from lowest to longest lambda-max. A data frame: first column corresponding to wavelength values and following columns with photoreceptor sensitivity values (see function `photor`).

- **interpolate**: Whether data files should be interpolated before further calculations. See `approx`.

- **nm**: A sequence of numeric values specifying where interpolation is to take place. See `approx`.

Details

The original model is available for trichromatic animals only. Thery and Casas (2002) derived a version for tetrachromatic animals which is implemented here. In colourvision, this model was extended to any number of photoreceptors types (Gawryszewski 2018; see also Pike 2012). The colour hexagon in Chittka (1992) has a vector of length = 1.0 The chromaticity coordinates in colourvision preserve the same vector length.

Photoreceptor outputs \(E_i\) are calculated by:

\[
E_i = \frac{q_i}{q_i + 1}
\]

where \(q_i\) is given by \(Qr\).

Then, for trichromatic vision, coordinates in the colour space are found by (Chittka 1992):

\[
X_1 = \frac{\sqrt{3}}{2}(E_3 - E_1)
\]

\[
X_2 = E_2 - \frac{1}{2}(E_1 + E_3)
\]

For tetrachromatic vision (Thery and Casas 2002):

\[
X_1 = \frac{\sqrt{3}\sqrt{2}}{3}(E_3 - E_4)
\]

\[
X_2 = E_1 - \frac{1}{3}(E_2 + E_3 + E_4)
\]
For a pentachromatic animal following the same vector length:

\[
X_1 = \frac{5}{2\sqrt{2}\sqrt{5}}(E_2 - E_1)
\]

\[
X_2 = \frac{5\sqrt{2}}{2\sqrt{3}\sqrt{5}}(E_3 - \frac{E_1 + E_2}{2})
\]

\[
X_3 = \frac{5\sqrt{3}}{4\sqrt{5}}(E_4 - \frac{E_1 + E_2 + E_3}{3})
\]

\[
X_4 = E_5 - \frac{E_1 + E_2 + E_3 + E_4}{4}
\]

**Value**

**Qri** Photoreceptor photon catch values after the von Kries transformation (see function `Qr`).

**Ei** Photoreceptor output values. Values can vary from 0 to 1.

**Xi** Coordinates in the colour space.

**deltaS** Euclidean distance to the origin of the colour space. It represents the conspicuousness of the stimulus (R) in relation to the background (Rb).

**Author(s)**

Felipe M. Gawryszewski <f.gawry@gmail.com>

**References**


**See Also**

`CTTKhexagon`, `CTTKhexagon3D`, `photor`, `RNLmodel`, `EMmodel`, `deltaS`
Examples

```r
## Photoreceptor sensitivity curves
## with lambda max at 350nm, 450nm and 550nm:
C <- photor(lambda.max = c(350, 450, 550))

## Grey background
## with 10 percent reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(10, length(300:700)))

## Read CIE D65 standard illuminant
data("D65")

## Reflectance data
## with a sigmoid spectrum and midpoint at 500nm
R <- logistic(x = seq(300, 700, 1), x0 = 500, L = 50, k = 0.04)

## Run model
model <- CTTKmodel(photo = 3, R = R, I = D65, 
                   Rb = Rb, C = C)

# plot
plot(model)
```

---

### D65

**CIE Standard Illuminant D65 in quantum flux (umol/m2/s)**

**Description**

CIE Standard Illuminant D65. Datum has already been converted to quantum flux (umol/m2/s) and therefore can be used in colour vision models directly.

**Usage**

```r
data("D65")
```

**Format**

A data frame with 107 observations on the following 2 variables.

- `l.nm` a numeric vector
- `Standard.Illuminant.D65` a numeric vector

**Source**

[http://www.cie.co.at/](http://www.cie.co.at/)

**Examples**

```r
data("D65")
plot(D65, type = "1")
```
**deltaS**

---

**Chromaticity distances**

### Description

Calculates a matrix with all possible pairwise comparison between stimulus reflectance spectra based on a given colour vision model output.

### Usage

```r
deltaS(model)
```

### Arguments

- **model**
  
  Output of a colour vision model.

### Value

A matrix with pairwise chromaticity distances.

### Author(s)

Felipe M. Gawryszewski `<f.gawry@gmail.com>`

### See Also

- `CTTKmodel`
- `RNLmodel`
- `EMmodel`
- `GENmodel`

### Examples

```r
## Photoreceptor sensitivity curves
## with lambda max at 350nm, 450nm and 550nm:
C <- photor(lambda.max = c(350, 450, 550))

## Grey background
## with 10 percent reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(10, length(300:700)))

## Read CIE D65 standard illuminant
data("D65")

## Reflectance data
## with a sigmoid spectrum and midpoint at 500nm
R1 <- logistic(x = seq(300, 700, 1), x0 = 450, L = 50, k = 0.04)
R2 <- logistic(x = seq(300, 700, 1), x0 = 500, L = 50, k = 0.04)
R3 <- logistic(x = seq(300, 700, 1), x0 = 550, L = 50, k = 0.04)
R <- cbind(R1, R2[,2], R3[,2])

## Run model
```
model<-CTTKmodel(photo=3, R=R, I=D65, Rb=Rb, C=C)

#Chromaticity distances between R1, R2 and R3
deltaS(model)

**EMline**

*Endler and Mielke (2005) 1-D colour space*

**Description**

Plots a colour space for dichromatic Endler and Mielke (2005) colour vision model.

**Usage**

```r
EMline(x,y=rep(0, length(x)), type="length", vnames=c("E1","E2"), ylim="auto", xlim="auto", ann=FALSE, axes = FALSE, ...)
```

**Arguments**

- `x` x-coordinate
- `y` y-coordinate
- `type` Whether the colour space should be built with a fixed vector length, or a fixed edge length (distance between vertices).
- `vnames` Vector names
- `ylim` see `plot` function.
- `xlim` see `plot` function.
- `ann` see `par` function.
- `axes` see `plot.default` function.
- `...` Other arguments passed to `plot`.

**Details**

The original model is available for tetrachromatic animals only. Colour space is built either with a vector length = 0.75 or a edge length = sqrt(3/2), to match the tetrahedron proposed by Endler and Mielke (2005).

**Author(s)**

Felipe M. Gawryszewski <f.gawry@gmail.com>

**References**

**Description**

Endler and Mielke (2005) colour vision model extended to animals with any number of photoreceptor types.

**Usage**

```r
EMmodel(photo = ncol(C)-1, type="length", R, I, Rb, C,
        interpolate=TRUE, nm=seq(300,700,1))
```

**Arguments**

- `photo` Number of photoreceptor types. Model accepts any number of photoreceptor types (\(\geq 2\)). For instance, dichromatic: `photo=2`; trichromatic: `photo=3`; tetrachromatic: `photo=4`, etc. Default gets number of photoreceptor types from `C` argument.
- `type` Whether the colour space should be built with a fixed vector length (type="length"), or a fixed edge length (distance between vertices; type="edge").
- `R` Reflectance of observed objects. A data frame with first column corresponding to wavelength values and following columns with reflectance values. `R` must be in the same scale as `Rb` (percentage or proportion).
- `I` Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.
- `Rb` Background reflectance. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values. `Rb` must be in the same scale as `R` (percentage or proportion).
- `C` Photoreceptor sensitivity curves, from lowest to longest lambda-max. A data frame: first column corresponding to wavelength values and following columns with photoreceptor sensitivity values (see function `photor`).
- `interpolate` Whether data files should be interpolated before further calculations. See `approx`.
- `nm` A sequence of numeric values specifying where interpolation is to take place. See `approx`.

**Examples**

```r
EMline(x=0.1, pch=16, col="red")
```
The original model is available for tetrachromatic animals only. In colour vision, the model was extended to any number of photoreceptors types (see also Pike 2012 formula).

First, relative quantum catches are log-transformed:

\[ f_i = \ln q_i \]

where \( q_i \) is the relative quantum catch of photoreceptor type \( i \), given by \( Q_r \). The model uses only relative output values, so that photoreceptor outputs are given by:

\[ E_i = \frac{f_i}{\sum_{i=1}^{n} f_i} \]

For tetrachromatic vision (Endler and Mielke 2005):

\[ X_1 = \sqrt{\frac{3}{2}} \left( \frac{1 - 2E_2 - E_3 - E_1}{2} \right) \]
\[ X_2 = \frac{-1 + 3E_3 + E_1}{2\sqrt{2}} \]
\[ X_3 = E_1 - \frac{1}{4} \]

Tetrachromatic chromaticity diagram (tetrahedron) in Endler and Mielke (2005) has a vector of length = 0.75 and edge length = sqrt(3/2). The chromaticity coordinates for other colour spaces may preserve either the same vector length or edge length.

For instance, for dichromatic vision coordinate (\( X_1 \)) in the colour space preserving the same vector length is found by:

\[ X_1 = \frac{3}{4} (E_2 - E_1) \]

Whereas for trichromatic vision coordinates (\( X_1 \) and \( X_2 \)) are found by:

\[ X_1 = \frac{3\sqrt{3}}{8} (E_2 - E_1) \]
\[ X_2 = \frac{3}{4} \left( E_3 - \frac{E_2 + E_1}{2} \right) \]

**Value**

| \( Q_{ri} \) | Photoreceptor photon catch values after the von Kries transformation (see function \( Q_r \)). |
| \( E_i \) | Photoreceptor outputs after conversion to relative values. |
| \( X_i \) | Coordinates in the colour space. |
| \( \delta S \) | Euclidean distance to the origin of the colour space. It represents the conspicuousness of the stimulus (\( R \)) in relation to the background (\( R_b \)). |

**Author(s)**

Felipe M. Gawryszewski <f.gawry@gmail.com>
References


See Also

EMline, EMtriangle, EMtetrahedron, photor, CTTKmodel, RNLmodel, GENmodel

Examples

##Photoreceptor sensitivity curves
##with lambda max at 350nm, 450nm and 550nm:
C<-photor(lambda.max=c(350,450,550))

##Gray background
##with 7 percent reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(7, length(300:700)))

## Read CIE D65 standard illuminant
data("D65")

##Reflectance data
## with a sigmoid spectrum and midpoint at 500nm and 550 nm
R1<-logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R2<-logistic(x=seq(300,700,1), x0=550, L=50, k=0.04)
R<-cbind(R1, R2[,2])
R[,2]<-R[,2]+10
R[,3]<-R[,3]+10

## Run model
model<-EMmodel(photo=3, type="edge",
R=R, I=D65, Rb=Rb, C=C)

plot(model)

EMtetrahedron

**Endler and Mielke (2005) tetrahedron colour space**

Description

Plots Endler and Mielke (2005) tetrahedron colour space for tetrachromatic animals.
EMtetrahedron

Usage

EMtetrahedron(x, y, z, s.col = "red", f.col = "black",
            vnames = c("u", "s", "m", "l"), type = "p",
            radius = 0.01, add = F, xlab = "",
            ylab = "", zlab = "", box = F, axes = F,
            ylim = c(-0.75, 0.75), xlim = c(-0.75, 0.75),
            zlim = c(-0.75, 0.75), aspect = T, vectors=FALSE, ...)

Arguments

  x  x coordinate of points to be plotted
  y  y coordinate of points to be plotted
  z  z coordinate of points to be plotted
  s.col  Colour to be used for plotted items. See plot3d in the rgl package.
  f.col  Colour of tetrahedron lines. See plot3d in the rgl package.
  vnames  Vector names.
  type  see plot3d function in the rgl package.
  radius  see plot3d function in the rgl package.
  add  see plot3d function in the rgl package.
  xlab  see plot3d function in the rgl package.
  ylab  see plot3d function in the rgl package.
  zlab  see plot3d function in the rgl package.
  box  see plot3d function in the rgl package.
  axes  see plot3d function in the rgl package.
  ylim  see plot3d function in the rgl package.
  xlim  see plot3d function in the rgl package.
  zlim  see plot3d function in the rgl package.
  aspect  see plot3d function in the rgl package.
  vectors  Whether vectors representing direction of photoreceptor outputs should be plotted.
  ...  Other arguments passed to function plot3d in the rgl package.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References


See Also

EMtriangle, EMmodel
EMtriangle

Endler and Mielke (2005) triangle colour space

Description
Plots a triangle colour space for trichromatic Endler and Mielke (2005) colour vision model.

Usage
EMtriangle(x, y, type=c("length", "edge"), vnames=c("u","s","m"),
ylim=c(-0.9,0.9), xlim=c(-0.9,0.9),
pch=16, bty="n", yaxt="n", xaxt="n",
col="black", asp=1, ann=FALSE, vectors=FALSE, ...)

Arguments
x x-coordinate
y y-coordinate
type Whether the colour space should be built with a fixed vector length, or a fixed edge length (distance between vertices).
vnames Vector names
ylim see plot function.
xlim see plot function.
pch see par function.
bty see par function.
yaxt see par function.
xaxt see par function.
col see par function.
asp see plot function.
ann see par function.
vectors Whether vectors representing direction of photoreceptor outputs should be plotted.
... Other arguments passed to plot.

Details
The original model is available for tetrachromatic animals only. Trichromatic version is implemented in colourvision based on Pike (2012) formula. The triangle is built either with a vector length = 0.75 or a edge length = sqrt(3/2), to match the tetrahedron proposed by Endler and Mielke (2005). Doris Gomez derived a trichromatic version which is available in software AVICOL (Gomez, 2006) and was previously implemented here (colourvision v0.1).
energytoflux

Author(s)
Felipe M. Gawryszewski <f.gawry@gmail.com>

References
Gomez, D. 2006. AVICOL, a program to analyse spectrometric data. Last update october 2011. Free executable available at: http://sites.google.com/site/avicolprogram/ or from the author at <dodogomez@yahoo.fr>

See Also
EMtetrahedron, EMmodel

Examples
  EMtriangle(x=0,y=0, type="length", pch=16, col="red")

energytoflux  Irradiance from energy to quantum units.

Description
Convert Irradiance datum from energy units (uW/cm2/nm) to quantum flux units (umol/m2/s)

Usage
energytoflux(datum)

Arguments
datum  A data frame with first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in energy units (uW/cm2/nm).

Value
A data frame with first column corresponding to wavelength values and second column with irradiance values in umol/m2/s.

Author(s)
Felipe M. Gawryszewski <f.gawry@gmail.com>
GENmodel

**N-dimensional generic colour vision model**

### Description

A flexible function to build colour vision models based on any number of photoreceptor types (Gawryszewski 2018).

### Usage

```r
GENmodel(photo=ncol(C)>1, type="length", length=NA, edge=NA,
R, I, Rb=NA, C, vonKries = TRUE, func, unity=FALSE,
recep.noise=FALSE, noise.given=TRUE, e=NA, v=NA, n=NA,
interpolate=TRUE, nm=seq(300,700,1))
```

### Arguments

- **photo**: Number of photoreceptor types. Model accepts any number of photoreceptor types (>=2). For instance, dichromatic: `photo=2`; trichromatic: `photo=3`; tetrachromatic: `photo=4`, etc. Default gets number of photoreceptor types from `C` argument.
- **type**: Whether the colour space should be built with a fixed vector length (`type="length"`), or a fixed distance between vertices (`type="edge"`; e.g. Pike 2012 and Renoult et al. 2015).
- **length**: Vector length used to construct the colour space. Used when `type="length"`. Typically `length=1`.
- **edge**: Edge length used to construct the colour space. Used when `type="edge"`.
- **R**: Reflectance of observed objects. A data frame with first column corresponding to wavelength values and following columns with reflectance values. `R` must be in the same scale as `Rb`.
- **I**: Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.
- **Rb**: Background reflectance. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values. `Rb` must be in the same scale as in `R`.
- **C**: Photoreceptor sensitivity curves, from lowest to longest lambda-max. A data frame: first column corresponding to wavelength values and following columns with photoreceptor sensitivity values (see function `photor`).
- **vonKries**: Whether the von Kries transformation should be applied to photoreceptor photon catches. Calculates the ratio between stimulus and background photoreceptor photon catches.
- **func**: Transformation of photoreceptors inputs. For instance, `func=log` for a ln-transformation.
- **unity**: Whether sum of photoreceptor output should be normalized to 1, as in `EMmodel`.
recep.noise  Logical. Whether chromaticity distances should be calculated based on receptor noises.

noise.given Logical. Whether receptor noise is provided (noise.given = TRUE) or calculated from photoreceptor relative abundances (noise.given = FALSE). Used when recep.noise=TRUE.

e Receptor noise of each photoreceptor type. Used when recep.noise = TRUE and noise.given = TRUE.

n Relative number of each photoreceptor type in the retina. Used to calculate e when recep.noise = TRUE and noise.given = TRUE.

v Noise-to-signal ratio of a single photoreceptor. Used to calculate e when recep.noise = TRUE and noise.given = TRUE.

interpolate Whether data files should be interpolated before further calculations. See approx.

nm A sequence of numeric values specifying where interpolation is to take place. See approx.

Value

ei Noise of photoreceptor channels. Calculated only when recep.noise=TRUE

Qri Photoreceptor photon catch values from R. Relative Photoreceptor photon catches when vonKries=TRUE. See functions Q and Qr.

Ei Photoreceptor outputs.

Xi Stimulus colour locus coordinates in the colour space

deltaS Euclidean distance to the origin of the colour space. It represents the conspicuousness of the stimulus (R) in relation to the background (Rb)

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References


See Also

Q, Qr, CTTKmodel, EMMmodel, RNLmodel, colour_space
Examples

# A trichromatic colour vision model based on Endler and Mielke (2005)

## Photoreceptor sensitivity curves
## with lambda max at 350nm, 450nm and 550nm:
C <- photor(lambda.max = c(350, 450, 550))

## Gray background
## with 7 percent reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(7, length(300:700)))

## Read CIE D65 standard illuminant
data("D65")

## Reflectance data
## with a sigmoid spectrum and midpoint at 500nm and 550 nm
R1 <- logistic(x = seq(300, 700, 1), x0 = 500, L = 50, k = 0.04)
R2 <- logistic(x = seq(300, 700, 1), x0 = 550, L = 50, k = 0.04)
R <- cbind(R1, R2[, 2])
R[, 2] <- R[, 2] + 10
R[, 3] <- R[, 3] + 10

## Run model
model <- GENmodel(length = 0.75, R = R, I = D65, Rb = Rb, C = C,
                   func = log, unity = TRUE)

plot(model)

---

**GENplot**

*Generic model colour space 2D and 1D plot*

**Description**

Plots models based on the GENmodel() function for trichromatic and dichromatic animals.

**Usage**

```r
GENplot(model, photo, col.names = c("X1", "X2"),
        vectors = TRUE, vnames = TRUE, vsize = "auto",
        ylab = "y", xlab = "x", xlim = "auto", ylim = "auto", asp = 1, ...)
```

**Arguments**

- **model**: Output of a colour vision model.
- **photo**: Number of photoreceptor types.
- **col.names**: Column names to be plotted.
- **vectors**: Whether vectors representing direction of photoreceptor outputs should be plotted.
vnames  Whether vector names should be plotted.
vsize   Length of vectors to be plotted. Default calculates length automatically.
ylab    y-axis range. Default calculates range automatically. See `par` function.
xlab    x-axis range. Default calculates range automatically. See `par` function.
xlim    see `par` function.
ylim    see `par` function.
asp     see `plot` function.
...     Other arguments passed to `plot` function.

Author(s)
Felipe M. Gawryszewski <f.gawry@gmail.com>

See Also
`CTTKhexagon`, `CTTKhexagon3D`, `EMtriangle`, `EMtetrahedron`, `RNLplot`, `RNLplot3d`, `GENplot3d`, `plot.colourvision`, `plot3d.colourvision`

---

### Description

Plots models based on the GENmodel() function for tetrachromatic animals.

### Usage

```r
GENplot3d(model, col.names=c("X1","X2","X3"),
          vectors=TRUE, vnames=TRUE, vsize="auto",
          xlab="x", ylab="y", zlab="z",
          xlim="auto", ylim="auto", zlim="auto", asp=1, ...)
```

### Arguments

- **model**: Output of a colour vision model.
- **col.names**: Column names to be plotted.
- **vectors**: Whether vectors representing direction of photoreceptor outputs should be plotted.
- **vnames**: Whether vector names should be plotted.
- **vsize**: Length of vectors to be plotted. Default calculates length automatically.
- **xlab**: see `par3d` function in the rgl package.
- **ylab**: see `par3d` function in the rgl package.
- **zlab**: see `par3d` function in the rgl package.
logistic

Logistic curve

Description
Generates a logistic curve.

Usage
logistic(x = seq(300, 700, 1), x0, L, k)

Arguments
- x: Range.
- x0: Midpoint value.
- L: Maximum value.
- k: Steepness of the curve

Author(s)
Felipe M. Gawryszewski <f.gawry@gmail.com>

See Also
CTTKhexagon, CTTKhexagon3D, EMtriangle, EMtetrahedron, RNLplot, RNLplot3d, GENplot, plot.colourvision, plot3d.colourvision

References
https://en.wikipedia.org/wiki/Logistic_function

Examples
l<-logistic(x=seq(300,700,1), x0=650, L=50, k=0.04)
plot(l, type="l")
### noise_e

**Receptor noise**

Receptor noise either provided by the user or based on noise-to-signal ratio of a single photoreceptor and the relative abundance of photoreceptor types in the retina. This function is used internally in Receptor Noise Limited models.

**Usage**

```r
noise_e(noise, e, v, n)
```

**Arguments**

- `noise` Logical. Whether receptor noise is provided (`noise = TRUE`) or calculated from photoreceptor relative abundances (`noise = FALSE`)
- `e` Receptor noise of each photoreceptor type. It is used when `noise = TRUE`
- `v` Noise-to-signal ratio of a single photoreceptor.
- `n` Relative abundance of the photoreceptor in the retina.

**Author(s)**

Felipe M. Gawryszewski <f.gawry@gmail.com>

**References**


**See Also**

`RNLmodel`, `RNLthres`, `GENmodel`, `colour_space`

### photor

**Photoreceptor sensitivity spectra.**

Generates photoreceptor sensitivity spectra based on lambda-max values.

**Usage**

```r
photor(lambda.max, lambda = seq(300, 700, 1), beta.band = FALSE)
```
Arguments

lambda.max  Vector with photoreceptor wavelength at maximum sensitivity values, in increasing order.
lambda        Range and interval to calculate the sensitivity curves.
beta.band    Logical. Whether or not to include a beta-band in the sensitivity curve.

Value

A data frame with first column corresponding to wavelength values and following columns with photoreceptor sensitivity values.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References


See Also

CTTKmodel, EMmodel, RNLmodel, RNLthres

Examples

## Generates photoreceptor sensitivity
## values with lambda max at 350nm, 450nm and 550nm:
C<-photor(lambda.max=c(350,450,550))

plot(C[,2]~C[,1], type="l", col="violet")
lines(C[,3]~C[,1], type="l", col="blue")
lines(C[,4]~C[,1], type="l", col="green")

Description

Plotting method for objects of class colourvision. Plotting method for animals with two or three photoreceptor types.

Usage

## S3 method for class 'colourvision'
plot(x, ...)
Arguments

x: Object of class "colourvision".

Additional arguments passed to the plot function. See CTTKhexagon for Chittka (1992) model (CTTKmodel) plotting arguments; EMtriangle and EMline for Endler and Mielke (2005) model (EMmodel) plotting arguments; RNLplot for RNL plotting arguments; and GENplot for generic model (GENmodel) plotting arguments.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References


See Also

plot3d.colourvision, EMtriangle, CTTKhexagon, EMmodel, CTTKmodel, RNLmodel, RNLthres

Examples

#trichromatic
##Photoreceptor sensitivity curves
C<-photor(lambda.max=c(350,450,550))

##Gray background
Rb <- data.frame(300:700, rep(7, length(300:700)))

## Read CIE D65 standard illuminant
data("D65")

##Reflectance data
R1<-logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R1[,2]<-R1[,2]+10

##Run models
model<-EMmodel(photo=3,
          R=R1, I=D65, Rb=Rb, C=C)
plot(model)

model<-CTTKmodel(photo=3,
          R=R1, I=D65, Rb=Rb, C=C)
plot(model)

model<-RNLmodel(model="log", photo=3,
          R=R1, I=D65, Rb=Rb, C=C, noise=TRUE, e=c(0.13, 0.06, 0.12))
plot3d.colourvision

### Description

'plot3d' method for objects of class `colourvision`. Plotting method for animals with four photoreceptor types.

### Usage

```r
## S3 method for class 'colourvision'
plot3d(x, ...)
```

### Arguments

- `x`: Object of class "colourvision".
Additional arguments passed to the plot3d function. See `CTTKhexagon3D` for Chittka (1992) model (CTTKmodel) plotting arguments; `EMtetrahedron` for Endler and Mielke (2005) model (EMmodel) plotting arguments; `RNLplot3d` for RNL plotting arguments; and `GENplot3d` for generic model (GENmodel) plotting arguments.

Author(s)
Felipe M. Gawryszewski <f.gawry@gmail.com>

References

See Also
`plot.colourvision`, `EMtetrahedron`, `CTTKhexagon3D`, `EMmodel`, `CTTKmodel`, `RNLmodel`

---

Q  

Total photon capture

Description
Total photoreceptor photon capture for a given irradiance, reflectance and photoreceptor sensitivity curve. This function is used internally in colour vision models.

Usage

\[ Q(R, I, C, \text{interpolate}, \text{nm}) \]

Arguments

- **R**: Reflectance of observed object. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values.
- **I**: Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.
- **C**: Photoreceptor sensitivity curve. A data frame with two columns only: first column corresponding to wavelength values and second column with photoreceptor absorbance values.
- **interpolate**: Whether data files should be interpolated before further calculations. See `approx`.
- **nm**: A sequence of numeric values specifying where interpolation is to take place. See `approx`.  

Q Total photon capture

R Reflectance of observed object. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values.

I Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.

C Photoreceptor sensitivity curve. A data frame with two columns only: first column corresponding to wavelength values and second column with photoreceptor absorbance values.

interpolate Whether data files should be interpolated before further calculations. See `approx`.

nm A sequence of numeric values specifying where interpolation is to take place. See `approx`.  

Value

Gives the total photoreceptor photon capture.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References


See Also

Qr, CTTKmodel, EMmodel, RNLmodel, RNLthres, GENmodel

Qr

Photoreceptor relative quantum catch

Description

von Kries transformation. Photoreceptors are assumed to be adapted to the background. This function is used internally in colour vision models.

Usage

Qr(R, I, Rb, C, interpolate, nm)

Arguments

R

Reflectance of observed object. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values.

I

Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.

Rb

Background reflectance. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values. Photoreceptors are assumed to be adapted to the background reflectance.
Photoreceptor sensitivity curve. A data frame with two columns only: first column corresponding to wavelength values and second column with photoreceptor absorbance values.

interpolate

Whether data files should be interpolated before further calculations. See approx.

nm

A sequence of numeric values specifying where interpolation is to take place. See approx.

Details

For the von Kries transformation, first the quantum catches of the observed reflectance and the environmental background are calculated (see \( Q \)). Then:

\[
q_i = \frac{Q_i}{Q_{bi}}
\]

where \( Q_i \) is the quantum catch arising from the observed object and \( Q_{bi} \) is the quantum catch from the background, for each one of the photoreceptor types (i).

Value

Photoreceptor relative quantum catch.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

References


See Also

CTTKmodel, EMmodel, RNLmodel, RNLthres, GENmodel
radarplot

**Radar plot**

Description

Plots quantum catches or E-values (photoreceptor outputs) into a radar plot.

Usage

```r
radarplot(model, item=c("Qr", "E"), item.labels=FALSE, item.lwd=1,
border=NULL, radar.lwd=1, radar.col="grey",
length="auto", xlim="auto", ylim="auto",
xlab="", ylab="", asp=1, add = FALSE, ...)
```

Arguments

- **model** Output of a colour vision model.
- **item** Whether photoreceptor inputs (Qr) or outputs (E) should be plotted.
- **item.labels** Logical. Whether vector names should be plotted.
- **item.lwd** Width of lines connecting item values.
- **border** Colour of lines connecting item values. See `polygon`.
- **radar.lwd** Width of lines representing item vectors.
- **radar.col** Colour of lines representing item vectors.
- **length** Length of item vectors. Default calculates length automatically.
- **xlim** x-axis range. Default calculates range automatically. See `plot` function.
- **ylim** y-axis range. Default calculates range automatically. See `plot` function.
- **xlab** see `par` function.
- **ylab** see `par` function.
- **asp** see `par` function.
- **add** Logical. Whether data should be added to an existing plot.
- **...** Other arguments passed to `plot`.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

Examples

```r
##Photoreceptor sensitivity curves
##with lambda max at 350nm, 450nm and 550nm:
C<-photor(lambda.max=c(350,450,550))

## Grey background
```

## with 10 percent reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(10, length(300:700)))

## Read CIE D65 standard illuminant
data("D65")

## Reflectance data
## with a sigmoid spectrum and midpoint at 500nm
R1<-logistic(x=seq(300,700,1), x0=450, L=50, k=0.04)
R2<-logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R3<-logistic(x=seq(300,700,1), x0=550, L=50, k=0.04)
R<-cbind(R1,R2[,2],R3[,2])

## Run model
model<-CTTKmodel(photo=3, R=R, I=D65,
                  Rb=Rb, C=C)

# Radarplot
radarplot(model, border=c("violet", "red", "blue"), item="E", item.labels=TRUE)

---

### Rb

#### Brazilian savannah background reflectance spectrum.

**Description**

Brazilian savannah background reflectance spectrum calculated by the average reflectance of leaf, leaf litter, tree bark and twigs.

**Usage**

data("Rb")

**Format**

A data frame with 401 observations on the following 2 variables.

- **X300.700** a numeric vector
- **cerrado** a numeric vector

**Source**

RNLmodel

Receptor Noise Limited Models (Vorobyev & Osorio 1998)

Description

Receptor noise limited colour vision models (Vorobyev & Osorio 1998; Vorobyev et al. 1998) extended to any number of photoreceptor types.

Usage

```r
RNLmodel(model = c("linear", "log"), photo=ncol(C)-1,
R1, R2=Rb, Rb, I, C,
noise = FALSE, v=NA, n=NA, e=NA,
interpolate = TRUE, nm = seq(300, 700, 1),
coord="colourvision")
```

Arguments

- **model**: Linear ("linear"; Vorobyev & Osorio, 1998), or log-linear ("log"; Vorobyev et al. 1998) RNLmodel version. model="log" is preferred. model="linear" can be used only if comparison is to be made between two very similar colours.
- **photo**: Number of photoreceptor types. Model accepts any number of photoreceptor types (>=2). For instance, dichromatic: photo=2; trichromatic: photo=3; tetra-chromatic: photo=4, etc. Default gets number of photoreceptor types from C argument.
- **R1**: Reflectance of observed objects. A data frame with first column corresponding to wavelength values and following columns with reflectance values. R1 must be in the same scale as R2 and Rb (percentage or proportion).
- **R2**: Reflectance to be compared against R1. R2=Rb if comparison is to be made against the background. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values. R2 must be in the same scale as R1 and Rb (percentage or proportion).
- **Rb**: Background reflectance. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values. Rb must be in the same scale as R1 and R2 (percentage or proportion).
- **I**: Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.
- **C**: Photoreceptor sensitivity curves, from lowest to longest lambda-max. A data frame: first column corresponding to wavelength values and following columns with photoreceptor sensitivity values (see function `photor`).
- **noise**: Logical. Whether receptor noise is provided (noise = TRUE) or calculated from photoreceptor relative abundances (noise = FALSE).
Receptor noise of each photoreceptor type. Used when noise = TRUE

Relative number of each photoreceptor type in the retina. Usually increases with lambda-max. Used to calculate e when noise = FALSE.

Noise-to-signal ratio of a single photoreceptor. Used to calculate e when noise = FALSE.

Whether data files should be interpolated before further calculations. See approx.

A sequence of numeric values specifying where interpolation is to take place. See approx.

Whether colour locous coordinates should be calculated by the method (coord="colourvision") by Gawryszewski (2018), or by alternative (coord="alternative"; available for 2-4 photoreceptor types only) methods (Hempel de Ibarra et al. 2001; Renoult et al. 2017).

The receptor noise limited model was originally developed to calculate $\Delta S$ between two reflectance curves directly, without finding colour locus coordinates (e.g. $x, y$; Vorobyev and Osorio 1998). This function uses later formulae to find colour loci in a chromaticity diagram (similarly to Hempel de Ibarra et al. 2001; Renoult et al. 2015).

In lack of a direct measurement, receptor noise ($e_i$) can be estimated by the relative abundance of photoreceptor types in the retina, and a measurement of a single photoreceptor noise-to-signal ratio:

$$e_i = \frac{\nu}{\sqrt{\eta}}$$

where $\nu$ is the noise-to-signal ratio of a single photoreceptor, and $\eta$ is the relative abundance of photoreceptor $i$ in the retina. Alternatively, noise may be dependent of the intensity, but this possibility is not implement in colourvision yet. Noise dependent of intensity usually holds for low light conditions only (Vorobyev et al. 1998).

- **Value**
  - $e_i$: Noise of photoreceptor channels.
  - $Q_{ri, R1}$: Photoreceptor photon catch values from $R1$, after the von Kries transformation (see function $Qr$).
  - $Q_{ri, R2}$: Photoreceptor photon catch values from $R2$, after the von Kries transformation (see function $Qr$).
  - $E_{i, R1}$: Photoreceptor outputs from the stimulus ($R1$)
  - $E_{i, R2}$: Photoreceptor outputs from $R2$
  - $Xi_{i, R1}$: Coordinates in the colour space for $R1$
  - $Xi_{i, R2}$: Coordinates in the colour space for $R2$. Equals zero when $R1=Rb$
  - $\delta_{S}$: Euclidean distance from $R1$ to $R2$. It represents the conspicuousness of the stimulus ($R1$) in relation to the background when $R1=Rb$.

**Author(s)**

Felipe M. Gawryszewski <f.gawry@gmail.com>


### References


### See Also

`photor`, `RNLthres.CTTKmodel`, `EMmodel`, `GENmodel`

### Examples

```r
#1
## Photoreceptor sensitivity spectra
##with lambda max at 350nm, 450nm and 550nm:
C<-photor(lambda.max=c(350,450,550))

##Grey background
##with 7 percent reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(7, length(300:700)))

## Read CIE D65 standard illuminant:
data("D65")

##Reflectance data of R1 and R2
R1.1<-logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R1.2<-logistic(x=seq(300,700,1), x0=400, L=50, k=0.04)
w<-R1.1[,1]
R1.1<-R1.1[,2]+10
R1.2<-R1.2[,2]+10
R1<-data.frame(w=w, R1.1=R1.1, R1.2=R1.2)

R2<-logistic(x=seq(300,700,1), x0=550, L=50, k=0.04)
R2[,2]<-R2[,2]+10

## Run model
model<-RNLmodel(photo=3, model="log",
                 R1=R1, R2=R2, Rb=Rb, I=D65, C=C,
                 noise=TRUE, e = c(0.13, 0.06, 0.12))

#plot
plot(model)

#2
```
# Pentachromatic animal
## Photoreceptor sensitivity spectra
## with lambda max at 350,400,450,500, and 550nm:
C <- photor(lambda.max = c(350, 400, 450, 500, 550))

## Grey background
## with 7 percent reflectance from 300 to 700nm:
Rb <- data.frame(300:700, rep(7, length(300:700)))

## Read CIE D65 standard illuminant:
data("D65")

## Reflectance data of R1
R1 <- logistic(x = seq(300, 700, 1), x0 = 500, L = 50, k = 0.04)
R1[, 2] <- R1[, 2] + 10

# RNL model
RNLmodel(photo = 5, model = "log",
        R1 = R1, R2 = Rb, Rb = Rb, I = D65, C = C,
        noise = TRUE, e = c(0.13, 0.06, 0.12, 0.07, 0.08))

RNLplot

---

**Receptor noise limited model 2D and 1D plot**

**Description**

Plots receptor noise limited model (RNL) for trichromatic and dichromatic animals.

**Usage**

RNLplot(model, photo, item = "R1",
        vectors = TRUE, vnames = TRUE, vsize = "auto",
        xlab = "x", ylab = "y", xlim = "auto", ylim = "auto", asp = 1, ...)

**Arguments**

- **model**: Output of a colour vision model.
- **photo**: Number of photoreceptor types.
- **item**: Model output item to be plotted. Default plots stimulus data. See `RNLmodel`.
- **vectors**: Whether vectors representing direction of photoreceptor outputs should be plotted.
- **vnames**: Whether vector names should be plotted.
- **vsize**: Length of vectors to be plotted. Default calculates length automatically.
- **xlab**: x-axis range. Default calculates range automatically. See `par` function.
- **ylab**: y-axis range. Default calculates range automatically. See `par` function.
- **xlim**: see `par` function. Default calculates xlim automatically.
ylim  
see \texttt{par} function. Default calculates ylim automatically.

\texttt{asp}  
see \texttt{plot} function.

...  
Other arguments passed to \texttt{plot} function.

\textbf{Author(s)}

Felipe M. Gawryszewski \texttt{<f.gawry@gmail.com>}

\textbf{See Also}

\texttt{CTTKhexagon}, \texttt{CTTKhexagon3D}, \texttt{EMtriangle}, \texttt{EMtetrahedron}, \texttt{RNLplot3d}, \texttt{plot.colourvision}, \texttt{plot3d.colourvision}

\textbf{Examples}

\begin{verbatim}
# dichromat
C <- photor(lambda.max=c(450,550))
Rb <- data.frame(300:700, rep(7, length(300:700)))
data("D65")
R1.1 <- logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R1.2 <- logistic(x=seq(300,700,1), x0=400, L=50, k=0.04)
w <- R1.1[,1]
R1.1 <- R1.1[,2]+10
R1.2 <- R1.2[,2]+10
R1 <- data.frame(w=w, R1.1=R1.1, R1.2=R1.2)
model <- RNLmodel(model="log", 
                 R1=R1, Rb=Rb, I=D65, C=C, 
                 noise=TRUE, e = c(0.13, 0.06))
plot(model)

# trichromat
C <- photor(lambda.max=c(350,450,550))
Rb <- data.frame(300:700, rep(7, length(300:700)))
data("D65")
R1.1 <- logistic(x=seq(300,700,1), x0=500, L=50, k=0.04)
R1.2 <- logistic(x=seq(300,700,1), x0=400, L=50, k=0.04)
w <- R1.1[,1]
R1.1 <- R1.1[,2]+10
R1.2 <- R1.2[,2]+10
R1 <- data.frame(w=w, R1.1=R1.1, R1.2=R1.2)
model <- RNLmodel(model="log", 
                 R1=R1, Rb=Rb, I=D65, C=C, 
                 noise=TRUE, e = c(0.13, 0.06, 0.12))
plot(model)
\end{verbatim}
RNLplot3d

Receptor noise limited model 3D plot

Description

Plots receptor noise limited model (RNL) for tetrachromatic animals.

Usage

RNLplot3d(model, item="R1",
           vectors=TRUE, vnames=TRUE, vsize="auto",
           xlab="x", ylab="y", zlab="z",
           xlim="auto", ylim="auto", zlim="auto", asp=1, ...)

Arguments

model
  Output of a colour vision model.
item
  Model output item to be plotted. Default plots stimulus data. See RNLmodel.
vectors
  Whether vectors representing direction of photoreceptor outputs should be plotted.
vnames
  Whether vector names should be plotted.
vsize
  Length of vectors to be plotted. Default calculates length automatically.
xlab
  see par3d function in the rgl package.
ylab
  see par3d function in the rgl package.
zlab
  see par3d function in the rgl package.
xlim
  x-axis range. Default calculates range automatically. See par3d function in the rgl package.
ylim
  y-axis range. Default calculates range automatically. See par3d function in the rgl package.
zlim
  z-axis range. Default calculates range automatically. See par3d function in the rgl package.
asp
  see plot3d function in the rgl package.
...
  Other arguments passed to function plot3d in the rgl package.

Author(s)

Felipe M. Gawryszewski <f.gawry@gmail.com>

See Also

CTTKhexagon, CTTKhexagon3D, EMtriangle, EMtetrahedron, RNLplot, plot.colourvision, plot3d.colourvision
**RNLthres**

*Colour thresholds based on the Receptor Noise Limited Model (Vorobyev & Osorio 1998).*

---

**Description**

Colour thresholds based on receptor noise for any number of photoreceptor types (Vorobyev & Osorio 1998).

**Usage**

\[
\text{RNLthres}(\text{photo}=\text{ncol(C)}-1, \text{Rb}, \text{I}, \text{C}, \text{noise}=\text{TRUE}, v=\text{NA}, n=\text{NA}, e=\text{NA}, \text{interpolate}=\text{TRUE}, \text{nm}=\text{seq}(300,700,1))
\]

**Arguments**

- **photo**
  Number of photoreceptor types. Model accepts any number of photoreceptor types (>=2). For instance, dichromatic: photo=2; trichromatic: photo=3; tetrachromatic: photo=4, etc. Default gets number of photoreceptor types from C argument.

- **Rb**
  Background reflectance. A data frame with two columns only: first column corresponding to wavelength values and second column with reflectance values.

- **I**
  Irradiance spectrum. A data frame with two columns only: first column corresponding to wavelength values and second column with irradiance values. Irradiance values must be in quantum flux units.

- **C**
  Photoreceptor sensitivity curves, from lowest to longest lambda-max. A data frame: first column corresponding to wavelength values and following columns with photoreceptor sensitivity values (see function `photor`).

- **noise**
  Logical. Whether receptor noise is provided (noise = TRUE) or calculated from photoreceptor relative abundances (noise = FALSE).

- **e**
  Receptor noise of each photoreceptor type. Used when noise = TRUE.

- **n**
  Relative number of each photoreceptor type in the retina. Usually increases with lambda-max. Used to calculate e when noise = FALSE.

- **v**
  Noise-to-signal ratio of a single photoreceptor. Used to calculate e when noise = FALSE.

- **interpolate**
  Whether data files should be interpolated before further calculations. See `approx`.

- **nm**
  A sequence of numeric values specifying where interpolation is to take place. See `approx`.

**Details**

Colour thresholds based on receptor noise limited model as in Vorobyev and Osorio (1998). In lack of a direct measurement, receptor noise \((\epsilon_i)\) can be estimated by the relative abundance of photoreceptor types in the retina, and a measurement of a single photoreceptor noise-to-signal ratio:

\[
\epsilon_i = \frac{\nu}{\sqrt{T_h}}
\]
where \( \nu \) is the noise-to-signal ratio of a single photoreceptor, and \( \eta \) is the relative abundance of photoreceptor \( i \) in the retina. Alternatively, noise may be dependent of the intensity, but this possibility is not implement in colourvision yet. Noise dependent of intensity usually holds for low light conditions only (Vorobyev et al. 1998).

**Value**

A data.frame with the following columns:

- nm: Wavelength in nm.
- T: Colour threshold value.
- S: Log of sensitivity value (inverse of threshold).

**Author(s)**

Felipe M. Gawryszewski <f.gawry@gmail.com>

**References**


**See Also**

photor, RNLmodel

**Examples**

```r
###Bee photoreceptors normalized to max=1.
data("bee")
C<-bee
C[,2]<-C[,2]/max(C[,2])
C[,3]<-C[,3]/max(C[,3])
C[,4]<-C[,4]/max(C[,4])

##Grey background:
Rb <- data.frame(300:700, rep(0.1, length(300:700)))

## CIE D65 illuminant:
data("D65")

#Thresholds
thres<-RNLthres(photo=3, Rb=Rb, I=D65, C=C,
noise=TRUE, e = c(0.13, 0.06, 0.12))
plot(thres)
```
**Description**

Applies a `smooth.spline` for data frame containing spectrometric data.

**Usage**

```r
spec.denoise(specfiles, spar = 0.7, ...)
```

**Arguments**

- **specfiles**: A data frame with first column representing wavelength values and following columns with reflectance data.
- **spar**: see `smooth.spline`. Smoothing parameter, typically (but not necessarily) in (0,1].
- **...**: Other arguments passed to function `smooth.spline`.

**Value**

A data frame with first column representing wavelength values and following columns with reflectance data.

**Author(s)**

Felipe M. Gawryszewski <f.gawry@gmail.com>
Index

* datasets
  bee, 4
  D65, 12
  Rb, 34

* package
  colourvision-package, 2

approx, 10, 15, 22, 30, 32, 36, 41
bee, 4

colour_space, 5, 22, 26
colourvision (colourvision-package), 2
colourvision-package, 2
CTTKhexagon, 6, 9, 11, 24, 25, 28, 39, 40
CTTKhexagon3D, 7, 8, 11, 24, 25, 30, 39, 40
CTTKmodel, 6, 7, 9, 13, 17, 22, 27, 28, 30–32, 37

D65, 12
deltaS, 11, 13

EMline, 14, 17, 28
EMmodel, 6, 11, 13, 15, 18, 20–22, 27, 28, 30–32, 37
EMtetrahedron, 15, 17, 18, 20, 24, 25, 30, 39, 40
EMtriangle, 15, 17, 18, 19, 24, 25, 28, 39, 40
energytoflux, 20

GENmodel, 6, 13, 17, 21, 26, 28, 30–32, 37
GENplot, 23, 25, 28
GENplot3d, 24, 24, 30

logistic, 25

noise_e, 26

par, 7, 14, 19, 24, 33, 38, 39
photor, 10, 11, 15, 17, 21, 26, 35, 37, 41, 42
plot, 7, 14, 19, 24, 33, 39

plot.colourvision, 24, 25, 27, 30, 39, 40
plot.default, 7, 14
plot3d.colourvision, 24, 25, 28, 29, 39, 40
polygon, 33
Q, 6, 22, 30, 32
Qr, 6, 10, 11, 16, 22, 31, 31, 36
radarplot, 33
Rb, 34
RNmodel, 6, 11, 13, 17, 22, 26–28, 30–32, 35, 38, 40, 42
RNplot, 24, 25, 28, 38, 40
RNplot3d, 24, 25, 30, 39, 40
RNLthres, 26–28, 31, 32, 37, 41
smooth.spline, 43
spec.denoise, 43

44