photobiologyWavebands Version 0.4.2
User Guide

Pedro J. Aphalo

January 8, 2017

1 Introduction

This package provides constructors for `waveband` objects, a class defined in package photobiology. These are convenience functions that allow definition of ranges of wavelengths and biological spectral weighting functions (BSWFs) following definitions in common use, CIE recommendations and ISO standards.

2 Waveband constructors for ranges

Functions for several colour bands, in some cases according to different optional definitions, are listed in Tables 1, 2, 3, and 4.

```
Red()

## Red.ISO
## low (nm) 610
## high (nm) 760
## weighted none

PAR()

## PAR
## low (nm) 400
## high (nm) 700
## weighted none

UV()

## UV.ISO
## low (nm) 100
## high (nm) 400
## weighted none

IR()

## IR.ISO
## low (nm) 780
## high (nm) 1e+06
## weighted none
```

Table 1: Functions in R package photobiologyWavebands used for constructing descriptors of ultraviolet wavebands used for calculation of irradiances or exposures. The boundaries of the band given as wavelengths in nm (λ). Definition according to ISO-21348 "ISO" is the default for all functions except `UVA1()` and `UVA2()` for which there is only one definition in common use "CIE" which is the default.

Waveband	Source	Dose or irrad.	Waveband (nm)
Ultraviolet			
UV	ISO-21348	`UV("ISO")`	$100 \leq \lambda < 400$
UV-C	ISO-21348	`UVC("ISO")`	$100 \leq \lambda < 280$
UV-C	n.a.	`UVC("medical")`	$220 \leq \lambda < 290$
UV-C	n.a.	`UVC("none")`	$200 \leq \lambda < 280$
UV-B	ISO-21348	`UVB("ISO")`	$280 \leq \lambda < 315$
UV-B	n.a.	`UVB("none")`	$280 \leq \lambda < 320$
UV-A	ISO-21348	`UVA("ISO")`	$315 \leq \lambda < 400$
UV-A	CIE	`UVA("CIE")`	$315 \leq \lambda < 400$
UV-A	n.a.	`UVA("none")`	$320 \leq \lambda < 400$
UV-A1	CIE	`UVA1("CIE")`	$315 \leq \lambda < 340$
UV-A2	CIE	`UVA2("CIE")`	$340 \leq \lambda < 400$
Arbitrary	n.a.	`new_waveband(lo,hi)`	$lo \leq \lambda < hi$

Constructors of lists of waveband definitions frequently used together are also defined in the package (Table 5).

```
UV_bands()

## [[1]]
## UVC.ISO
## low (nm) 100
## high (nm) 280
## weighted none
##
## [[2]]
## UVB.ISO
## low (nm) 280
## high (nm) 315
## weighted none
##
## [[3]]
## UVA.ISO
## low (nm) 315
## high (nm) 400
## weighted none
```

Table 2: Functions in R package photobiologyWavebands used for constructing descriptors of visible wavebands used for calculation of irradiances or exposures. The boundaries of the band given as wavelengths in nm (λ). Definition according to ISO-21348 "ISO" is the default for all functions except `PAR()` for which there is only one definition in common use "Plant" which is the default.

Waveband	Source	Dose or irrad.	Waveband (nm)
Visible			
Visible	ISO-21348	`VIS("ISO")`	$380 \leq \lambda < 760$
Photosynthesis	n.a.	`PAR("Plant")`	$400 \leq \lambda < 700$
Purple	ISO-21348	`Purple("ISO")`	$360 \leq \lambda < 450$
Blue	Sellaro	`Blue("Sellaro")`	$420 \leq \lambda < 490$
Blue	ISO-21348	`Blue("ISO")`	$450 \leq \lambda < 500$
Green	Sellaro	`Green("Sellaro")`	$500 \leq \lambda < 570$
Green	ISO-21348	`Green("ISO")`	$500 \leq \lambda < 570$
Yellow	ISO-21348	`Yellow("ISO")`	$570 \leq \lambda < 591$
Orange	ISO-21348	`Orange("ISO")`	$591 \leq \lambda < 610$
Red	ISO-21348	`Red("ISO")`	$610 \leq \lambda < 760$
Red	Smith	`Red("Smith10")`	$655 \leq \lambda < 665$
Red	Smith ?	`Red("Smith20")`	$650 \leq \lambda < 670$
Red	Inada	`Red("Inada")`	$600 \leq \lambda < 700$
Red	Warrington	`Red("Warrington")`	$625 \leq \lambda < 675$
Red	Sellaro	`Red("Sellaro")`	$620 \leq \lambda < 680$
Far-red	Smith10	`Far_red("Smith")`	$725 \leq \lambda < 735$
Far-red	Smith20	`Far_red("Smith")`	$720 \leq \lambda < 740$
Far-Red	Inada	`Red("Inada")`	$700 \leq \lambda < 800$
Far-Red	Warrington	`Red("Warrington")`	$700 \leq \lambda < 850$
Far-red	Sellaro	`Far_red("Sellaro")`	$700 \leq \lambda < 750$
Far-red	BTV	`Far_red("BTV")`	$700 \leq \lambda < 760$
Arbitrary	n.a.	`new_waveband(lo,hi)`	$lo \leq \lambda < hi$

Table 3: Functions in R package photobiologyWavebands used for constructing descriptors of infrared wavebands used for calculation of irradiances or exposures. The boundaries of the band are given as wavelengths in nm (λ). Definition according to ISO-20473, "ISO" is the default for all functions except `IRA()`, `IRB()`, `IRC()` are defined based on CIE recommendations.

Waveband	Source	Dose or irrad.	Waveband (nm)
Infrared			
IR	ISO-20473	`IR("ISO")`	$780 \leq \lambda < 1 \times 10^6$
NIR	ISO-20473	`NIR("ISO")`	$780 \leq \lambda < 3000$
MIR	ISO-20473	`MIR("ISO")`	$3000 \leq \lambda < 50000$
FIR	ISO-20473	`FIR("ISO")`	$5000 \leq \lambda < 1 \times 10^6$
IR	CIE	`IR("CIE")`	$700 \leq \lambda < 1 \times 10^6$
IR-A	CIE	`IRA("CIE")`	$780 \leq \lambda < 1400$
IR-B	CIE	`IRB("CIE")`	$1400 \leq \lambda < 3000$
IR-C	CIE	`IRC("CIE")`	$3000 \leq \lambda < 1 \times 10^6$
SWIR	n.a.	`SWIR()`	$1400 \leq \lambda < 3000$
Arbitrary	n.a.	`new_waveband(lo,hi)`	$lo \leq \lambda < hi$

```r
VIS_bands()

## [[1]]
## Purple.ISO
## low (nm) 360
## high (nm) 450
## weighted none
##
## [[2]]
## Blue.ISO
## low (nm) 450
## high (nm) 500
## weighted none
##
## [[3]]
## Green.ISO
## low (nm) 500
## high (nm) 570
## weighted none
##
## [[4]]
## Yellow.ISO
## low (nm) 570
## high (nm) 591
## weighted none
##
## [[5]]
## Orange.ISO
## low (nm) 591
## high (nm) 610
## weighted none
```

Table 4: Functions in R package photobiologyWavebands used for constructing descriptors of Landsat-imager wavebands. The boundaries of the band are given as wavelengths in nm (λ). Definitions follow NASA and USGS documentation for boundaries at half maximum (HM).

Waveband	imager	constructor	Wavelengths (nm)
Landsat			
Purple/Coastal	OLI	`Purple("LandsatOLI")`	$433 \leq \lambda < 453$
Blue	TM	`Blue("LandsatTM")`	$450 \leq \lambda < 520$
Blue	ETM	`Blue("LandsatETM")`	$450 \leq \lambda < 520$
Blue	OLI	`Blue("LandsatOLI")`	$450 \leq \lambda < 515$
Green	RBV	`Green("LandsatRBV")`	$480 \leq \lambda < 580$
Green	MSS	`Green("LandsatMSS")`	$500 \leq \lambda < 600$
Green	TM	`Green("LandsatTM")`	$520 \leq \lambda < 600$
Green	ETM	`Green("LandsatETM")`	$520 \leq \lambda < 600$
Green	OLI	`Green("LandsatOLI")`	$525 \leq \lambda < 600$
Red	RBV	`Red("LandsatRBV")`	$580 \leq \lambda < 680$
Red	MSS	`Red("LandsatMSS")`	$600 \leq \lambda < 700$
Red	TM	`Red("LandsatTM")`	$630 \leq \lambda < 690$
Red	ETM	`Red("LandsatETM")`	$630 \leq \lambda < 690$
Red	OLI	`Red("LandsatOLI")`	$630 \leq \lambda < 680$
NIR	RBV	`NIR("LandsatRBV")`	$700 \leq \lambda < 830$
NIR	MSS	`NIR("LandsatMSS")`	$800 \leq \lambda < 1100$
NIR	TM	`NIR("LandsatTM")`	$760 \leq \lambda < 900$
NIR	ETM	`NIR("LandsatETM")`	$760 \leq \lambda < 900$
NIR	OLI	`NIR("LandsatOLI")`	$845 \leq \lambda < 885$
SWIR1	TM	`SWIR("LandsatTM")`	$1550 \leq \lambda < 1750$
SWIR1	ETM	`SWIR("LandsatETM")`	$1550 \leq \lambda < 1750$
SWIR1	OLI	`SWIR("LandsatOLI")`	$1560 \leq \lambda < 1660$
SWIR2	TM	`SWIR("LandsatTM")`	$2080 \leq \lambda < 2350$
SWIR2	ETM	`SWIR("LandsatETM")`	$2080 \leq \lambda < 2350$
SWIR2	OLI	`SWIR("LandsatOLI")`	$2100 \leq \lambda < 2300$
TIR1	TIRS	`TIR1("LandsatTIRS")`	$10600 \leq \lambda < 11200$
TIR2	TM	`TIR2("LandsatTM")`	$10400 \leq \lambda < 12500$
TIR2	ETM	`TIR2("LandsatETM")`	$10400 \leq \lambda < 12500$
TIR2	TIRS	`TIR2("LandsatTIRS")`	$11500 \leq \lambda < 12500$

```
##
## [[6]]
## Red.ISO
## low (nm) 610
## high (nm) 760
## weighted none
```

```
IR_bands()

## [[1]]
## NIR.ISO
## low (nm) 780
## high (nm) 3000
## weighted none
##
## [[2]]
## MIR.ISO
## low (nm) 3000
## high (nm) 50000
## weighted none
##
## [[3]]
## FIR.ISO
## low (nm) 50000
## high (nm) 1e+06
## weighted none
```

Additional constructors are provided for Landsat missions, for example the list of wavebands used in mission Landsat 1 can be created directly.

```
Landsat_bands("L1")

## [[1]]
## Green.LandsatRBV
## low (nm) 480
## high (nm) 580
## weighted none
##
## [[2]]
## Red.LandsatRBV
## low (nm) 580
## high (nm) 680
## weighted none
##
## [[3]]
## NIR.LandsatRBV
## low (nm) 700
## high (nm) 830
## weighted none
##
## [[4]]
## Green.LandsatMSS
## low (nm) 500
## high (nm) 600
## weighted none
```

Table 5: Functions in R package photobiologyWavebands used for constructing lists descriptors of wavebands used for calculation of irradiances or exposures.

Waveband	Source	Function	std values used
UV defs.	ISO-21348	`UV_bands("ISO")`	"ISO"
UV defs.	CIE	`UV_bands("CIE")`	"CIE"
UV defs.	n.a.	`UV_bands("medical")`	"medical"
UV defs.	n.a.	`UV_bands("none")`	"none"
VIS defs.	ISO-21348	`VIS_bands("ISO")`	"ISO"
IR defs.	ISO-20473	`IR_bands("ISO")`	"ISO"
IR defs.	CIE	`IR_bands("CIE")`	"CIE"
Plant sens.	n.a.	`Plant_bands("sensory20")`	"ISO", "Sellaro" and "Smith20"
Plant sens.	n.a.	`Plant_bands("sensory")`	"ISO", "Sellaro" and "Smith20"
Plant sens.	n.a.	`Plant_bands("sensory10")`	"ISO", "Sellaro" and "Smith10"
Plant sens.	n.a.	`Plant_bands("")`	"ISO" and "McCree"
Plant sens.	n.a.	`Plant_bands("energy")`	"ISO" and "McCree"
Plant sens.	n.a.	`Plant_bands("photon")`	"ISO" and "McCree"
Plant sens.	n.a.	`Plant_bands("ISO")`	"ISO" and "McCree"
Plant sens.	n.a.	`Plant_bands("CIE")`	"CIE" and "McCree"
Landsat	NASA/USGS	`Landsat_bands("L8")`	Landsat 8 mission
Landsat	NASA/USGS	`Landsat_bands("L`x`")`	Landsat $x = 1 \ldots 8$ missions

```
##
## [[5]]
## Red.LandsatMSS
## low (nm) 600
## high (nm) 700
## weighted none
##
## [[6]]
## NIR.LandsatMSS
## low (nm) 800
## high (nm) 1100
## weighted none
```

3 Calculating irradiances

An example using `sun_spct` included in package `photobiology`. As the input spectral irradiance is units of $\mathrm{W\,m^{-2}\,nm^{-1}}$ the output is in $\mathrm{mol\,m^{-2}\,s^{-1}}$ or $\mathrm{W\,m^{-2}}$.

```
e_irrad(sun.spct, UV()) # W m-2

##  UV.ISO.tr.lo
##      28.62872
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy irradiance total"

q_irrad(sun.spct, UV()) * 1e6 # umol s-1 m-2
```

```
##  UV.ISO.tr.lo
##      86.49506
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "photon irradiance total"
```

Irradiances for different wavebands can be grouped into a list of any length. If the list has named members, then these names are used instead of the default ones.

```
e_irrad(sun.spct, list(Blue(), VIS()))

##  Blue.ISO   VIS.ISO
##  37.55207 231.86345
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy irradiance total"

e_irrad(sun.spct, list(B = Blue(), VIS()))

##  Blue.ISO   VIS.ISO
##  37.55207 231.86345
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy irradiance total"
```

A few functions for generating coherent lists of wavebands are also defined (Table 5).

```
e_irrad(sun.spct, VIS_bands())

##  Purple.ISO    Blue.ISO   Green.ISO  Yellow.ISO
##    47.75529    37.55207    49.26860    13.67971
##  Orange.ISO     Red.ISO
##    12.00432    79.38159
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy irradiance total"
```

4 Photon ratios

Photon ratios can be calculated from any pair of waveband objects. This a convenient and very flexible way of doing this type of calculations.

```
q_ratio(sun.spct, Blue(), VIS())

##  Blue.ISO: VIS.ISO(q:q)
```

```
##              0.1371157
## attr(,"radiation.unit")
## [1] "q:q ratio"
```

5 Effective irradiances and exposures

The waveband definitions and SWFs are stored in `waveband` objects, that can be created with function `waveband`. The same functions described above for unweighted irradiances are used to calculate effective irradiances and doses.

Currently functions for constructing `waveband` objects describing several BSWFs are implemented (see Table 6). These functions take three arguments in most cases as they have been used and continue to be used inconsistently in the scientific literature. By supplying these arguments different variations of the BSWFs can be obtained. The defaults used are those values which we consider best, usually the most frequently used ones, except in cases when we consider the use of those values problematic for the reliability of the calculations.

```
e_irrad(sun.spct, CIE())

##  CIE98.298.tr.lo
##       0.08181583
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy irradiance total"
```

6 Action spectra at given wavelengths

The functions available for calculating action spectra take as argument a vector of wavelengths, and return a vector of effectiveness (either quantum/photon or energy based) depending on how the original source describes them. These functions are listed in Table 7, and an example of their use follows.

```
# at 1 nm intervals
wavelengths1 <- 285:400
action.spectrum1 <- CIE_e_fun(wavelengths1)
```

All functions accept a wavelengths vector with variable and arbitrary step sizes, with the condition that the wavelengths are sorted in strictly increasing order.

In practice these functions are mainly used internally by the package, and very rarely in user code, as the same output can by obtained by multiplication of `source_spct` objects by `waveband` objects.

Compare the following two plots,

Table 6: Functions in R package photobiologyWavebands used for constructing `waveband` objects describing BSWFs used for calculation of effective irradiances or doses. The functions for BSWFs available in this package, are by default as in the original source. Optionally they can be normalized to any wavelength within their non-zero range by providing the `norm` argument with a wavelength in nm. The range of wavelengths included when calculating integrals is given by `w.low` and `w.high`. The values in the table below are the defaults.

Action spectrum	Formulation	Constructor function	norm λ (nm)	w.low λ (nm)	w.high λ (nm)
Gen. plant action	Green	`GEN_G(norm, w.low, w.high)`	300	275	313.3
Gen. plant action	Thimijan	`GEN_T(norm, w.low, w.high)`	300	275	345.0
Gen. plant action	Micheletti	`GEN_M(norm, w.low, w.high)`	300	275	313.3
Plant growth	Flint & Caldwell	`PG(norm, w.low, w.high)`	300	275	390.0
Erythemal	CIE98	`CIE(norm, w.low, w.high)`	298	250	400.0
ICNIRP	ICNIRP2004	`ICNIRP(norm, w.low, w.high)`	270	210	400.0
'Naked' DNA	TUV, from Setlow	`DNA_N(norm, w.low, w.high)`	300	250	400.0
'Naked' DNA	Green & Miller	`DNA_GM(norm, w.low, w.high)`	300	250	400.0
'Plant' DNA	Musil, from Quaite	`DNA_P(norm, w.low, w.high)`	300	250	400.0
Flavonoid	Ibdah	`FLAV(norm, w.low, w.high)`	300	275	346.0

Table 7: Biological spectral weighting functions predefined in R package photobiologyWavebands. The functions for BSWFs available in this package, implement the functions as defined in the original publications. When the original 'definition' is available as tabulated data, or we have tabulated it by digitizing a figure, the values returned are calculated by spline interpolation.

Action spectrum	Formulation	Function	Norm. λ (nm)
Gen. plant action	Green	`GEN_G_q_fun(w.length)`	280
Gen. plant action	Thimijan	`GEN_T_q_fun(w.length)`	300
Gen. plant action	Micheletti	`GEN_M_q_fun(w.length)`	300
Plant growth	Flint & Caldwell	`PG_q_fun(w.length)`	300
Erythemal	CIE98	`CIE_e_fun(w.length)`	298
ICNIRP	ICNIRP2004	`ICNIRP_e_fun(w.length)`	270
'Naked' DNA	TUV, from Setlow	`DNA_N_q_fun(w.length)`	n.a.
'Naked' DNA	Green & Miller	`DNA_GM_q_fun(w.length)`	n.a.
'Plant' DNA	Musil, from Quaite	`DNA_P_q_fun(w.length)`	290
Flavonoid	Ibdah	`FLAV_q_fun(w.length)`	300

```
sun.spct * CIE()

## Object: source_spct [122 x 2]
## Wavelength range 280 to 400 nm, step 0.9230769 to 1 nm
## Time unit 1s
## Data weighted using 'CIE98.298' BSWF
##
## # A tibble: 122 × 2
##    w.length s.e.irrad
##       <dbl>     <dbl>
## 1  280.0000         0
## 2  280.9231         0
## 3  281.8462         0
## 4  282.7692         0
## 5  283.6923         0
## 6  284.6154         0
## 7  285.5385         0
## 8  286.4615         0
## 9  287.3846         0
## 10 288.3077         0
## # ... with 112 more rows
```

7 Luminous flux

The luminous flux per unit are in lux can be calculated as follows using the original luminous efficiency function for the human eye using for defining the lumen. As we start with spectral irradiance we obtain luminous flux per unit area expressed in lux.

```
e_response(sun.spct * CIE1924_lef.spct) * photopic_sensitivity

##    Total
## 49579.93
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy response total"
```

The luminous flux per unit area in lux can be calculated as follows using the latest luminous efficiency function for the human eye.

```
e_response(sun.spct * CIE2008_lef2deg.spct) * photopic_sensitivity

##    Total
## 53057.78
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy response total"
```

As the luminous efficiency functions vary slightly in the wavelength at which the maximum is located, and the wavelength used for the sensitivity constant is fixed by the definition of the Lumen, a small correction is need for exact results.

```
e_response(sun.spct * CIE2008_lef2deg.spct) * photopic_sensitivity *
                       interpolate_spct(CIE2008_lef2deg.spct, 555)$s.e.response

##    Total
## 53910.01
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy response total"
```

An equivalent quantity can be calculated for scotopic vision, using the corresponding function and constant.

```
e_response(sun.spct * 1e-6 * CIE1951_scotopic_lef.spct) * scotopic_sensitivity

##     Total
## 0.1186256
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy response total"
```