photobiologyPlants Version 0.3.3
User Guide

Pedro J. Aphalo

December 21, 2015

1 Introduction

We have developed a set of packages to facilitate the calculation of many different quantities that can be derived from spectral irradiance data. The basic package is called photobiology, and the package described here is an extension of the basic facilities for quantification of Phytochrome, Chryptochrome and UVR8 (plant photoreceptors). It will be submitted to CRAN (Comprehensive R archive network), it is meanwhile available from `http://www.r4photobiology.info`.

2 Installation and use

The functions in the package photobiologyPlants are made available by installing the packages `photobiologyPlants` (once) and loading it from the library when needed.

To load package photobiologyPlants into the workspace of the current R session we use `library(photobiologyPlants)`.

```
library(ggplot2)
library(ggspectra)
library(photobiology)
library(photobiologyPlants)
library(photobiologyWavebands)
```

3 Spectral data

See the User Guide for package photobiology for instructions on how to work with spectral data.

> All functions in the photobiology suite expect wavelength in nanometres (nm), spectral energy irradiances in $\mathrm{W\,m^{-2}\,nm^{-1}}$ and spectral photon irradiances in $\mathrm{mol\,m^{-2}\,s^{-1}\,nm^{-1}}$. They do not rescale the data.

It is very important to make sure that the wavelengths are in nanometers as this is what the functions expect. If the wavelengths supplied as arguments are in the wrong units, the returned values will be wrong.

Below we use the solar spectral data included in package photobiology as a data frame in object `sun.data`.

Part I
PHY

4 Calculating the Phytochrome photoequilibrium

Photoequilibrium from a `source_spct` object:

```
Pfr_Ptot(sun.spct)

## [1] 0.68341
```

Red:far-red ratio:

```
R_FR(sun.spct)

##  Red.Smith10: FarRed.Smith10(q:q)
##                          1.266704
## attr(,"radiation.unit")
## [1] "q:q ratio"
```

which is equivalent to:

```
q_ratio(sun.spct, Red("Smith"), Far_red("Smith"))

##  Red.Smith10: FarRed.Smith10(q:q)
##                          1.266704
## attr(,"radiation.unit")
## [1] "q:q ratio"
```

We can, and should whenever spectral data are available, calculate the photoequilibrium as above from them. It is possible to obtain and approximation in case of the solar spectrum and other broad spectra, using the red:far-red photon ratio. The calculation, however, is only stricly valid, for di-chromatic illumination with red plus far-red light.

```
Pfr_Ptot_R_FR(R_FR(sun.spct))

##  Red.Smith10: FarRed.Smith10(q:q)
##                         0.7051691
```

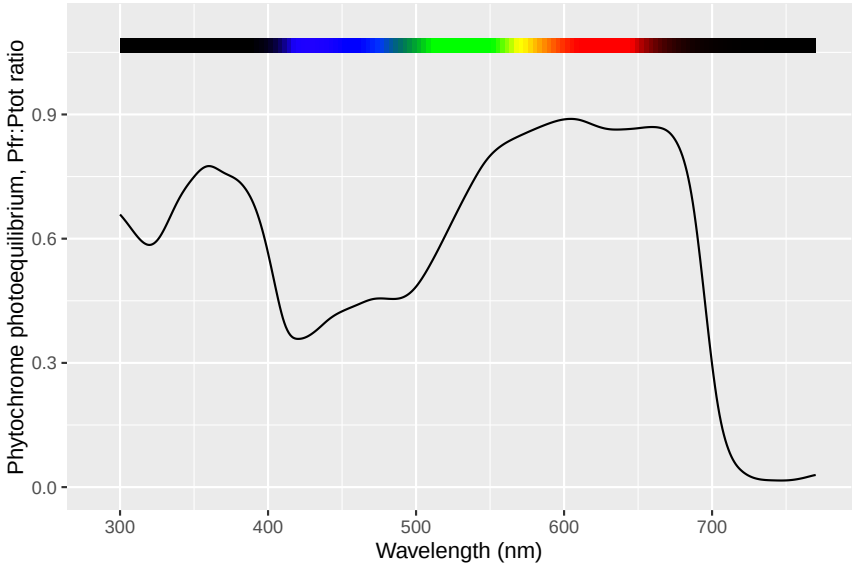
Here we calculated the R:FR ratio from spectral data, but in practice one would use this function only when spectral data is not available as when a R plus FR sensor is used. We can see tha In such a case the photoequilibrium calcualted is only a very rough approximation. For sunlight, in the example above when using spectral data we obtained a value of 0.683 in contrast to 0.705 when using the R:FR photon ratio. For other light sources differences can be much larger.

In the case of monochromatic light we can still use the same functions, as the defaults are such that we can use a single value as the 'w.length' argument, to obtain the Pfr:P ratio. For monchromatic light irradiance is irrelevant for the photoequilibrium.

```
Pfr_Ptot(660)

## [1] 0.869649

Pfr_Ptot(735)

## [1] 0.01749967

Pfr_Ptot(c(660, 735))

## [1] 0.86964902 0.01749967

Pfr_Ptot(435)

## [1] 0.3859998
```

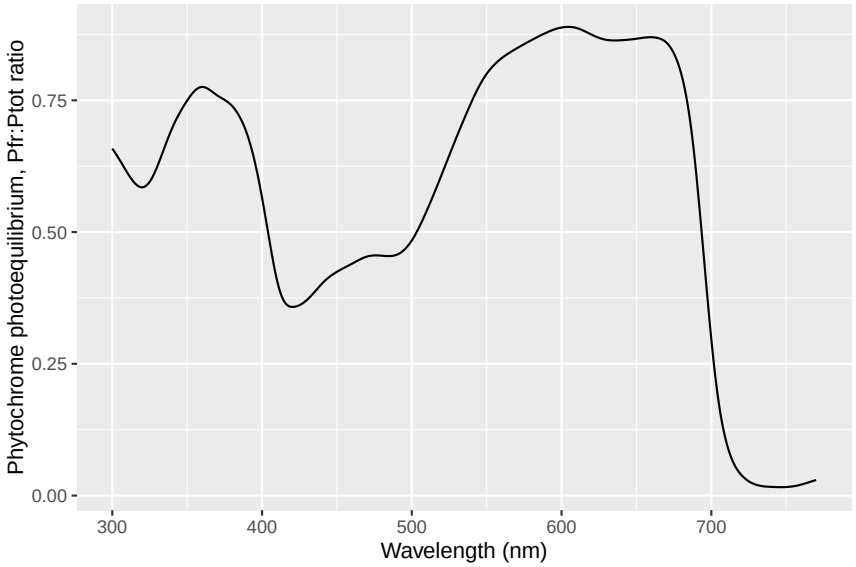
We can also plot Pfr:Ptot as a function of wavelength (nm) of monochromatic light. The default is to return a vector for short input vectors, and a `response_spct` object otherwise, but this can be changed through argument `spct.out`.

```
plot(Pfr_Ptot(300:770), norm = NULL, unit.out = "photon",
     w.band = Plant_bands(),
     annotations = c("colour.guide", "labels", "boxes")) +
  labs(y = "Phytochrome photoequilibrium, Pfr:Ptot ratio")
```



It is, of course, also possible to use base R plotting functions, or as shown here `ggplot` functions:

```
ggplot(data=Pfr_Ptot(300:770), aes(w.length, s.q.response)) +
  geom_line() +
  labs(x = "Wavelength (nm)",
     y = "Phytochrome photoequilibrium, Pfr:Ptot ratio")
```

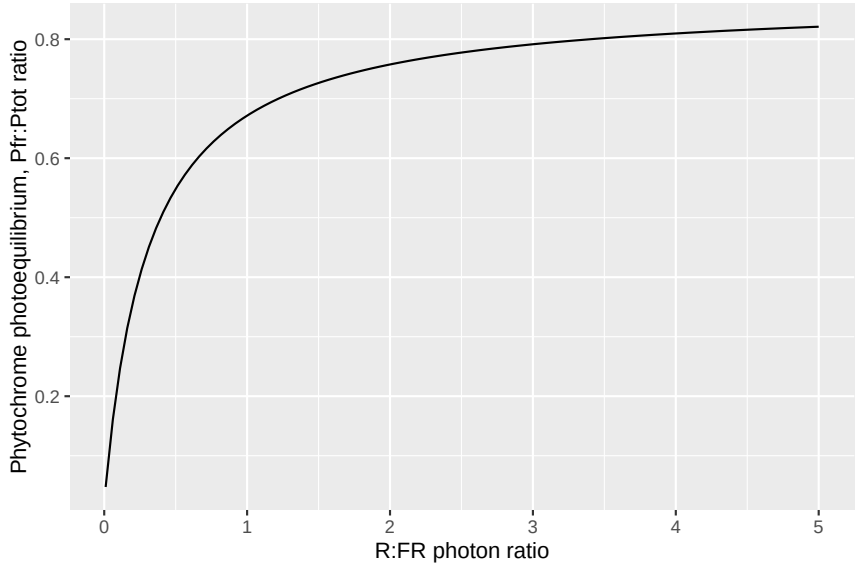


In the case of dichromatic illumination with red (660 nm) and far-red (730 nm) light, we can use a different function that takes the R:FR photon ratio as argument.

```
Pfr_Ptot_R_FR(1.15)

## [1] 0.6919699

Pfr_Ptot_R_FR(0.01)

## [1] 0.04747996

Pfr_Ptot_R_FR(c(1.15,0.01))

## [1] 0.69196990 0.04747996
```

Of course it is also easy to plot Pfr:P ratio as a function of R:FR photon ratio. However we have to remember that such values are exact only for dichromatic light, and only a rough approximation for wide-spectrum light sources. For light spectrum light sources, the photoequilibrium should, if possible, be calculated from spectral irradiance data.

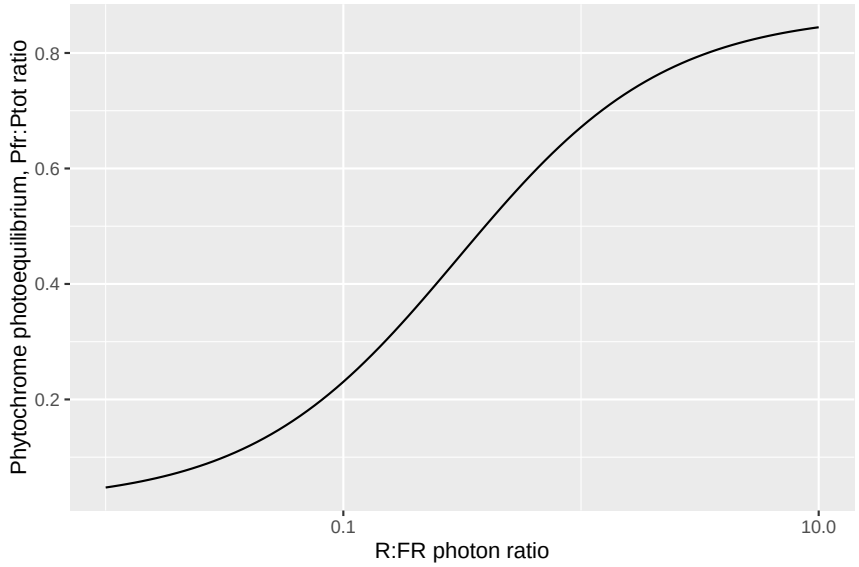
```
ex6.data <- data.frame(r.fr=seq(0.01, 5.0, length.out=100), Pfr.p=numeric(100))
ex6.data$Pfr.p <- Pfr_Ptot_R_FR(ex6.data$r.fr)
ggplot(data=ex6.data, aes(r.fr, Pfr.p)) +
  geom_line() +
    labs(x ="R:FR photon ratio",
         y = "Phytochrome photoequilibrium, Pfr:Ptot ratio")
```



Below we try to reproduce figure 11 from Mancinelli (1994), where he uses a logarithmic scale for the R:FR ratio.

```
ex7.data <- data.frame(r.fr = 10^seq(log10(0.01), log10(10.0), length.out=100),
                       Pfr.p = numeric(100))
ex7.data$Pfr.p <- Pfr_Ptot_R_FR(ex7.data$r.fr)
```

```
ggplot(data=ex7.data, aes(r.fr, Pfr.p)) +
  geom_line() +
    scale_x_log10() +
      labs(x ="R:FR photon ratio",
           y = "Phytochrome photoequilibrium, Pfr:Ptot ratio")
```



5 Calculating reaction rates

```
with(sun.data, Phy_reaction_rates(w.length, s.e.irrad))

## $k1
## [1] 1.259332
##
## $k2
## [1] 0.5833862
##
## $nu
## [1] 1.842718
```

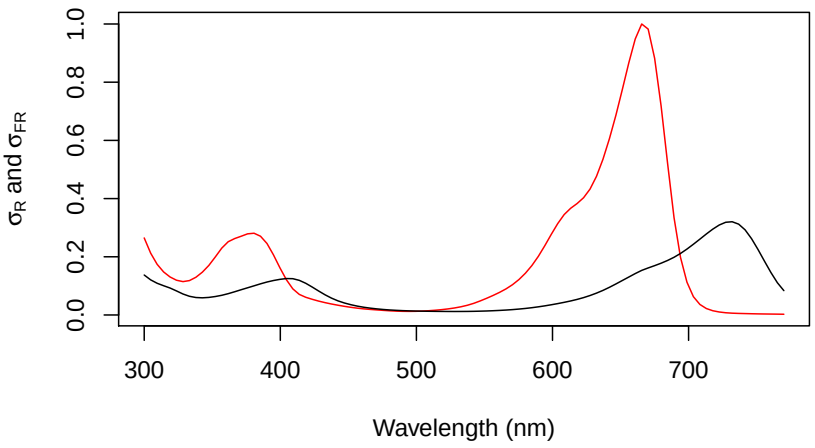
6 Calculating the absorption cross section at given wavelengths

The phytochrome photoequilibrium cannot be calculated from the absportance spectra of Pr and Pfr, because Pr and Pfr have different quantum yields for the respective phototransformations. We need to use action spectra, which in this

context are usually called 'absorption cross-sections'. They can be calculated as the product of absorbance and quantum yield. The values in these spectra, in the case of Phy are called 'Sigma'.

Here we reproduce Figure 3 in Mancinelli (1994), which gives the 'Relative photoconversion cross-sections' of Pr (σ_R) and Pfr (σ_{FR}). The values are expressed relative to σ_R at its maximum at $\lambda = 666$ nm.

```
ex7.data <- data.frame(w.length=seq(300, 770, length.out=100))
ex7.data$sigma.r <- Phy_Sigma_R(ex7.data$w.length)
ex7.data$sigma.fr <- Phy_Sigma_FR(ex7.data$w.length)
ex7.data$sigma <- Phy_Sigma(ex7.data$w.length)
plot(I(sigma.r/ max(sigma.r)) ~ w.length, data=ex7.data, type="l", col="red",
     xlab="Wavelength (nm)", ylab=expression(sigma[R]~and~sigma[FR]))
lines(I(sigma.fr/max(sigma.r)) ~ w.length, data=ex7.data)
rm(ex7.data)
```



Part II
CRY

7 Calculating energy or photons absorbed by Crytochrome 2

```
options(photobiology.filter.qty = "absorbance")
```

```
q_response(sun.spct * CRY2_dark.spct)
```

```
##   Total
## 2207091
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "photon response total"
```

```
e_response(sun.spct * CRY2_light.spct)
```

```
##    Total
## 7.778865
## attr(,"time.unit")
## [1] "second"
## attr(,"radiation.unit")
## [1] "energy response total"
```

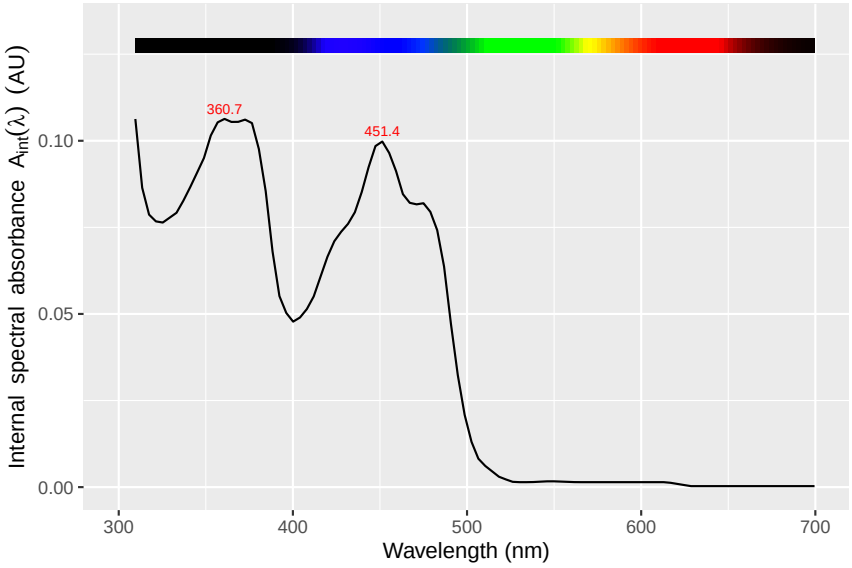
8 Calculating the CRY absorbance at given wavelengths

Here we try to reproduce Figure 1.B from Banerjee et al. (2007).

```
interpolate_wl(CRY2_dark.spct, 300:500)
```

```
## Object: filter_spct [202 x 2]
## Wavelength (nm): range 300 to 500, step 0.467 to 1
##
##    w.length     A
##       (dbl) (dbl)
## 1       300    NA
## 2       301    NA
## 3       302    NA
## 4       303    NA
## 5       304    NA
## 6       305    NA
## 7       306    NA
## 8       307    NA
## 9       308    NA
## 10      309    NA
## ..      ...   ...
```

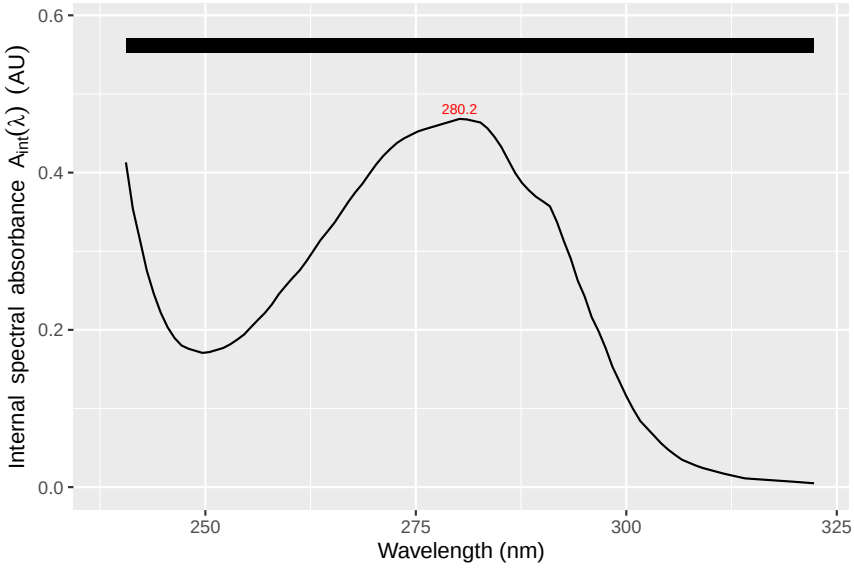
```
plot(CRY2_dark.spct, range = c(300,700))
```



Part III
UVR8

9 Plot

```
plot(UVR8_Glasgow.spct)
```



10 Coming soon!