

photobiologyPlants Version 0.0.1

User Guide

Pedro J. Aphalo

January 22, 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://r4photobiology.wordpress.com>.

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(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 $\text{W m}^{-2} \text{nm}^{-1}$ and spectral photon irradiances in $\text{mol m}^{-2} \text{s}^{-1} \text{nm}^{-1}$. 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

```
attach(sun.data)
```

```
Pr_P_ratio(w.length, s.e.irrad)
```

```
## [1] 0.68341
```

```
R.FR <- R_FR_ratio(w.length,s.e.irrad)
```

```
Pr_P_ratio_R_FR(R.FR)
```

```
## [1] 0.7035189
```

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. In such a case the photoequilibrium is only a very rough approximation, as this function is meant for calculation of the photoequilibrium under dichromatic radiation.

```
detach(sun.data)
```

Above we have used `attach` and `detach` as we wanted to keep the code lines short and simple. It is also possible, and recommended to use `with` instead:

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

```
## [1] 0.68341
```

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 Pr:P ratio. For monochromatic light irradiance is irrelevant for the photoequilibrium.

```
Pr_P_ratio(660)

## [1] 0.869649

Pr_P_ratio(735)

## [1] 0.01749967

Pr_P_ratio(435)

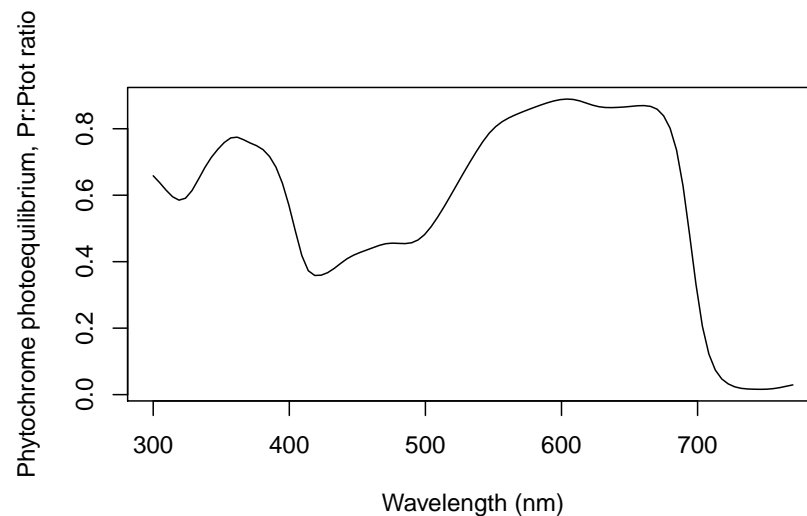
## [1] 0.3859998
```

We can also plot Pr:P as a function of wavelength (nm) of monochromatic light.

```
ex5.data <- data.frame(w.length=seq(300, 770, length.out=100), pr.p=numeric(100))
attach(ex5.data)

ex5.data$pr.p <- Pr_P_ratio_mono(ex5.data$w.length)

plot(pr.p ~ w.length, xlab="Wavelength (nm)",
     ylab="Phytochrome photoequilibrium, Pr:Ptot ratio",
     type="l", data=ex5.data)
rm(ex5.data)
```



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.

```
Pr_P_ratio_R_FR(1.15)

## [1] 0.6919699
```

```
Pr_P_ratio_R_FR(0.01)

## [1] 0.04747996

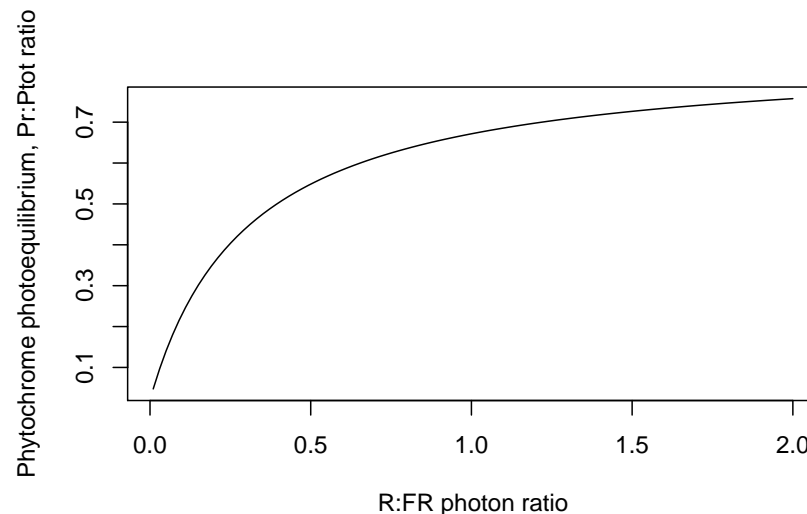
Pr_P_ratio_R_FR(c(1.15,0.01))

## [1] 0.69196990 0.04747996
```

Of course it is also easy to plot Pr: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, 2.0, length.out=100), pr.p=numeric(100))
ex6.data$pr.p <- Pr_P_ratio_R_FR(ex6.data$r.fr)

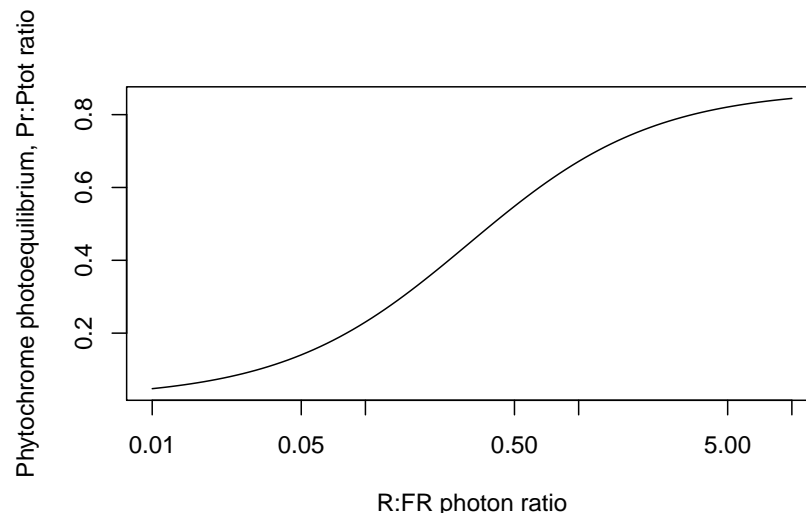
plot(pr.p ~ r.fr, xlab="R:FR photon ratio",
      ylab="Phytochrome photoequilibrium, Pr:Ptot ratio",
      type="l", data=ex6.data)
rm(ex6.data)
```



Here 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)), pr.p=numeric(100))
ex7.data$pr.p <- Pr_P_ratio_R_FR(ex7.data$r.fr)

plot(pr.p ~ r.fr, xlab="R:FR photon ratio", log="x",
      ylab="Phytochrome photoequilibrium, Pr:Ptot ratio",
      type="l", data=ex7.data)
rm(ex7.data)
```



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

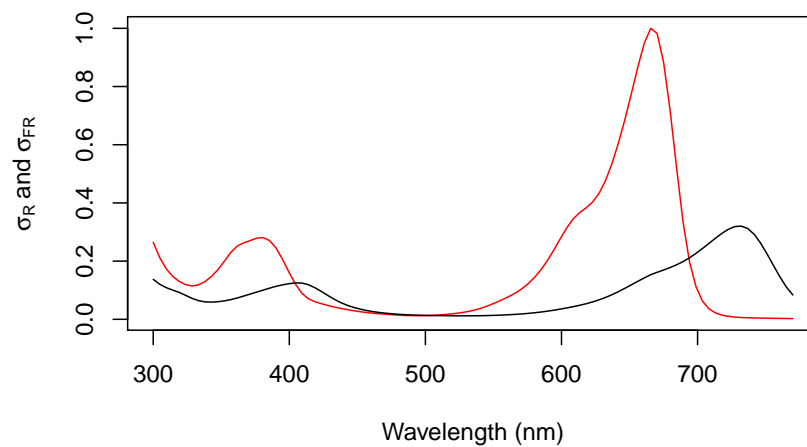
The phytochrome photoequilibrium cannot be calculated from the absorbance spectra of Pr and Pfr, because Pr and Pfr have different quantum yields for the respective phototransformations. We need to use action spectra, which in this context are usually called ‘absorption cross-sections’. They can be calculated 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 the Cryptochrome effective irradiances

```

attach(sun.spct)

## The following object is masked from ex5.data:
##
##   w.length

photon_irradiance(w.length, s.e.irrad, CRY2.Abs(previous="darkness"))

## CRY2.Abs.darkness
##   2.735279e-05

```

```
photon_irradiance(w.length, s.e.irrad, CRY2.Abs(previous="light"))

## CRY2.Abs.light
##      2.87217e-05
```

Here we calculated the effective photon irradiance weighted with the absorbance spectrum, first using the spectrum for “dark-adapted” CRY2 and then for “blue-light-adapted” CRY2.

```
energy_irradiance(w.length, s.e.irrad, CRY2.Abs(previous="darkness"))

## CRY2.Abs.darkness
##      7.700883
```

```
energy_irradiance(w.length, s.e.irrad, CRY2.Abs(previous="light"))

## CRY2.Abs.light
##      7.773691
```

Here we calculated the effective (energy) irradiance weighted with the absorbance spectrum, first using the spectrum for “dark-adapted” CRY2 and then for “blue-light-adapted” CRY2.

```
energy_irradiance(w.length, s.e.irrad, CRY2.Abs(norm=400, previous="light"))

## CRY2.Abs.light
##      104.6284
```

As shown above, it is also possible to indicate a normalization wavelength, in this case 400 nm. If no normalization wavelength is given, the absorbance spectrum data is used as is.

```
detach(sun.spct)
```

We have used here **attach** and **detach** as we used the same data in several examples, and we wanted to keep the code lines short and simple. It is also possible, and even recommended to use **with** instead:

```
with(sun.spct, photon_irradiance(w.length, s.e.irrad, CRY2.Abs()))

## CRY2.Abs.light
##      2.87217e-05
```

8 Calculating the CRY absorbance at given wavelengths

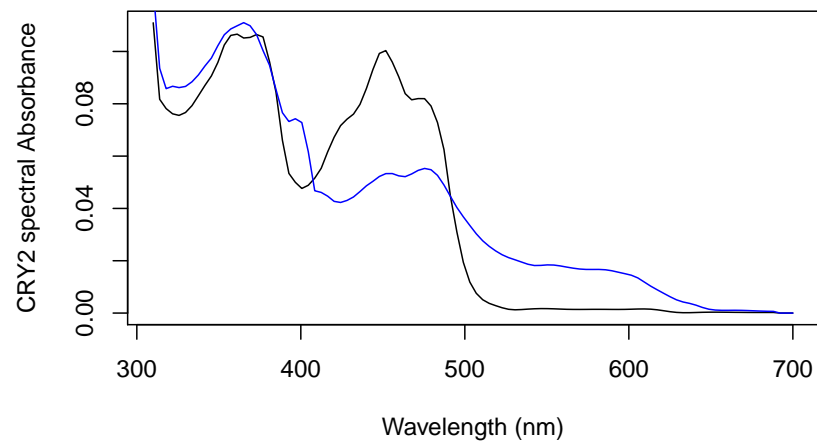
Here we try to reproduce Figure 1.B from Banerjee et al. (2007).

```

ex7.data <- data.frame(w.length=seq(310, 700, length.out=100),
                      A.dark=numeric(100),
                      A.light=numeric(100))
ex7.data$A.dark <- CRY2_Abs_dark_fun(ex7.data$w.length)
ex7.data$A.light <- CRY2_Abs_light_fun(ex7.data$w.length)

plot(A.dark ~ w.length, xlab="Wavelength (nm)",
     ylab="CRY2 spectral Absorbance",
     type="l", data=ex7.data)
lines(A.light ~ w.length, data=ex7.data, col="blue")
rm(ex7.data)

```



Part III

UVR8

9 Coming soon!