

photobiologyFilters Version 0.4.3

User Guide

Pedro J. Aphalo

August 4, 2016

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 core package is called **photobiology**, and the package described here adds transmittance data for some frequently used filters.

2 Calculating average transmittance

```
uv.226.new.spct <- rosco.mspct$UV_filter_EColour226
transmittance(uv.226.new.spct, UV_bands())

##      UVC.ISO.tr.lo      UVB.ISO      UVA.ISO
## 0.0001290556 0.0000100000 0.0476859412
## attr(,"Tfr.type")
## [1] "total"
## attr(,"radiation.unit")
## [1] "transmittance average"
```

```
transmittance(uv.226.new.spct, PAR())

##      PAR
## 0.8819025
## attr(,"Tfr.type")
## [1] "total"
## attr(,"radiation.unit")
## [1] "transmittance average"
```

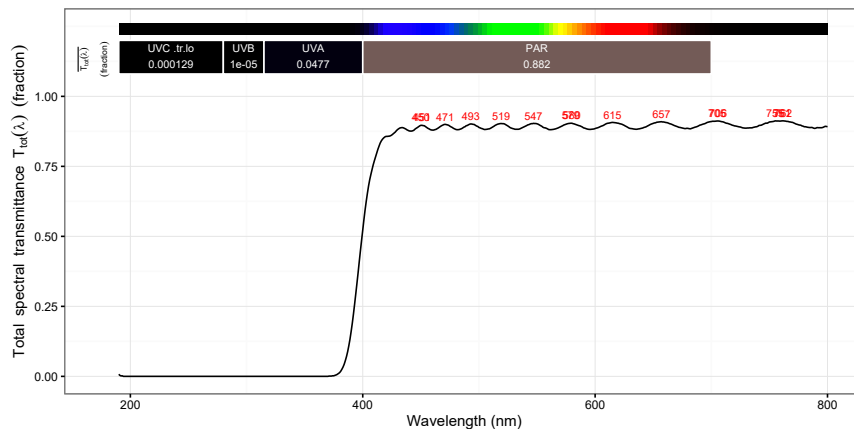
```
absorbance(uv.226.new.spct, PAR())

##      PAR
## 0.05526775
## attr(,"Tfr.type")
## [1] "total"
## attr(,"radiation.unit")
## [1] "absorbance average"
```

3 Plotting the transmittance spectrum of filters

We can plot the data as is, accepting all defaults. As the returned value is a ggplot object, we can modify and add to it as needed.

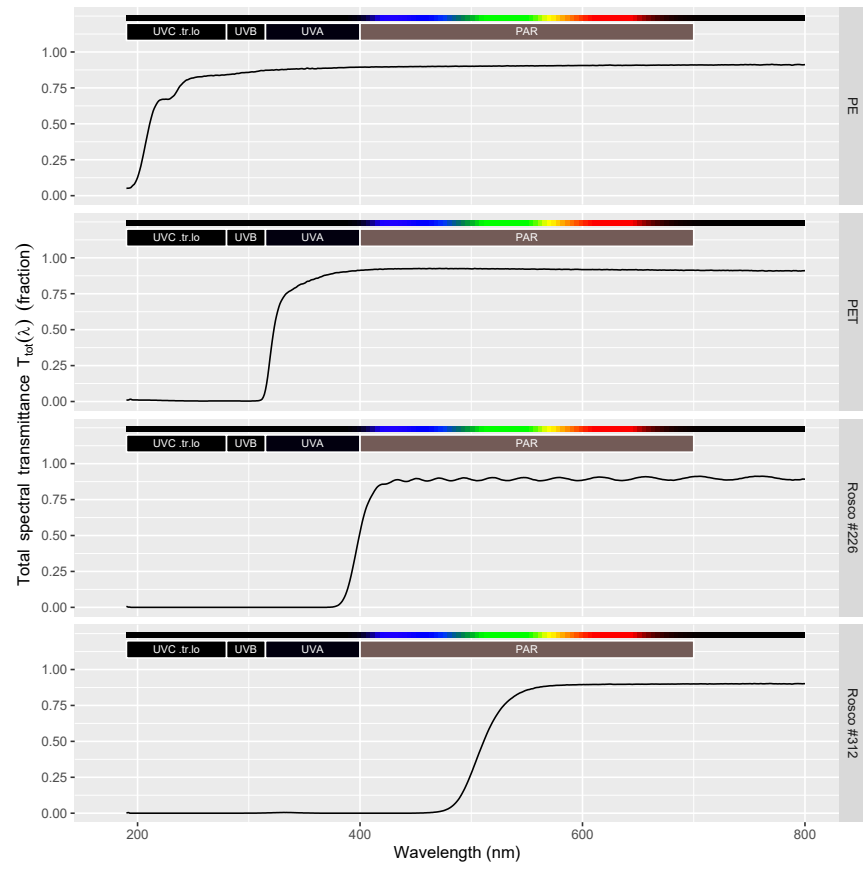
```
plot(uv.226.new.spct) + theme_bw(10)
```



We can also row-bind several spectra and plot them in panels.

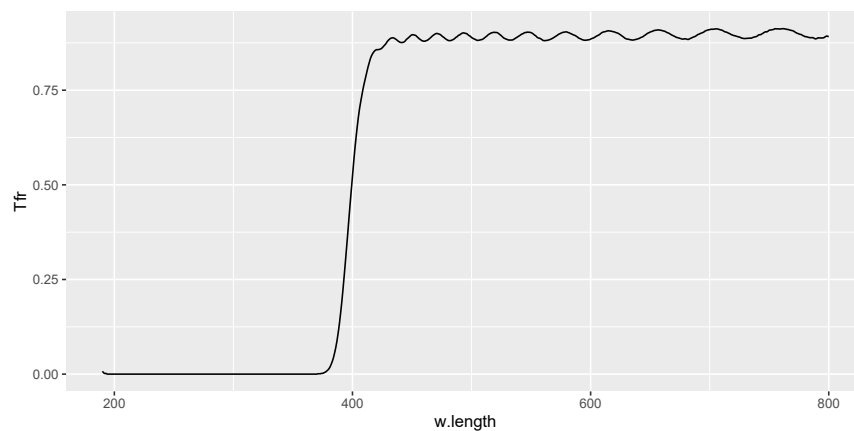
```
filters.spct <- rbindspct(list(PE = etola.mspct$Clear_LD_PE_50um,
                              PET = mcdermit.mspct$Autostat_CT5_125um,
                              'Rosco #226' = rosco.mspct$UV_filter_EColour226,
                              'Rosco #312' = rosco.mspct$Canary_Supergel312),
                          idfactor = "filter")
```

```
plot(filters.spct,
      annotations = c("colour_guide", "boxes", "labels")) +
  facet_grid(filter~.)
```



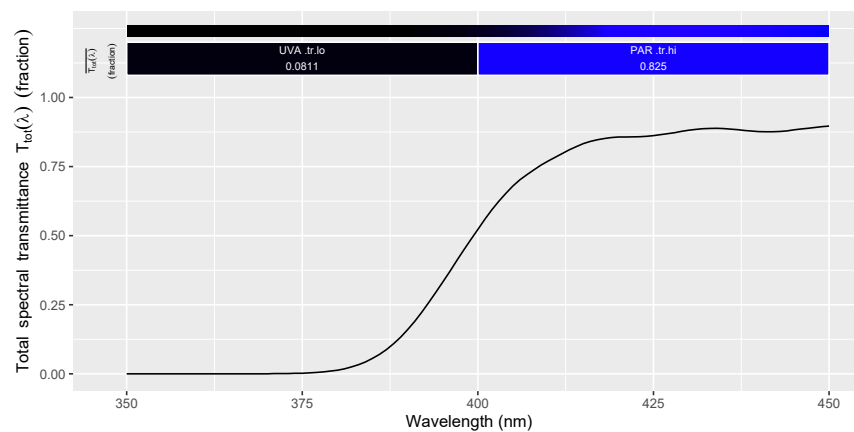
We can also use ggplot (or lattice or base graphics functions) to plot the spectrum.

```
ggplot(data=uv.226.new.spct, aes(x=w.length, y=Tfr)) + geom_line()
```



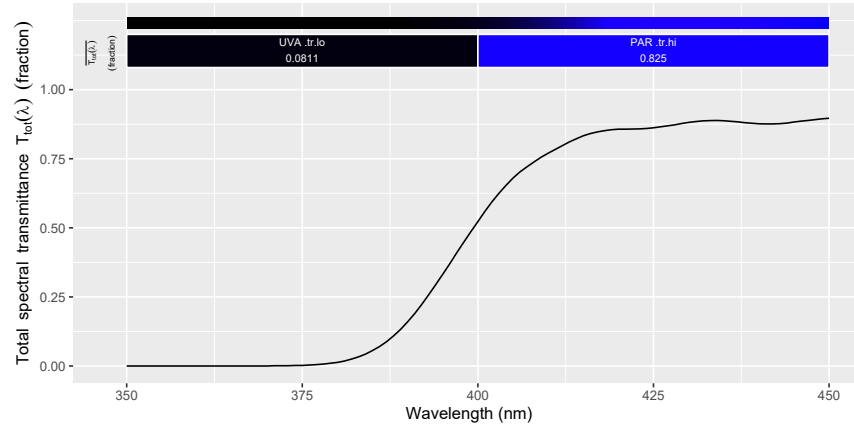
If we would like to use a different range of wavelengths for several calculations, then we can build a new filter.spct by trimming.

```
my.uv.226.spct <-  
  trim_spct(uv.226.new.spct, range = c(350,450), byref = FALSE)  
plot(my.uv.226.spct)
```



However, if we only would like to limit the range of wavelengths in the plot, it is even easier to use the **range** formal argument of function **plot** to obtain the same plot.

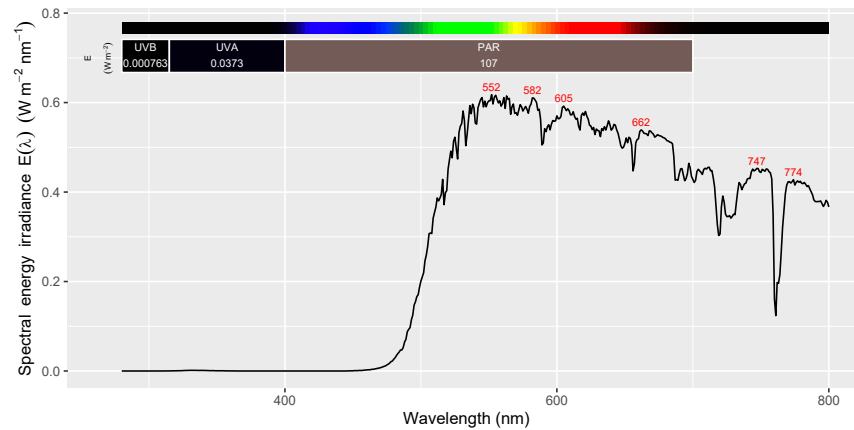
```
plot(uv.226.new.spct, range = c(350, 450))
```



4 Convoluting spectral irradiance with filter transmittance

To simulate the spectral irradiance of a filtered light source we convolute the bare source's irradiance spectrum with the filter transmittance spectrum.

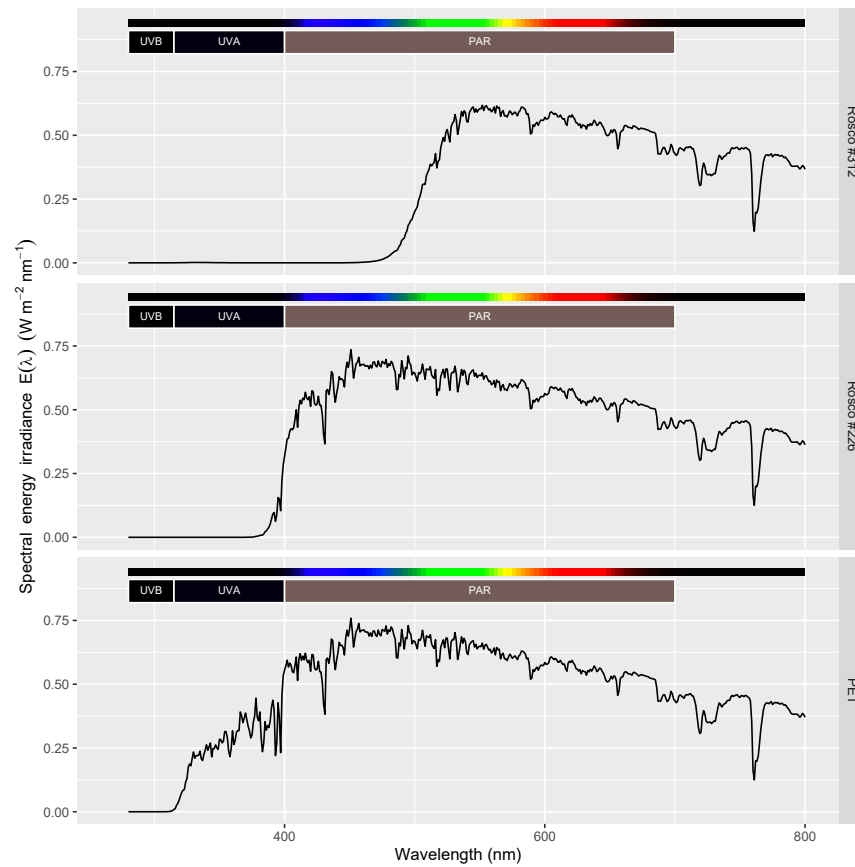
```
plot(sun.spct * rosco.mspct$Canary_Supergel312)
```



If we want to compare the effect of several filters on the spectrum of a light source we can create a combined spectral object by row-binding the source.spect objects obtained as a result of the convolutions.

```
all.spect <-
  rbindspect(list('Rosco #312' = sun.spect * rosclo.mspect$Canary_Supergel312,
    'Rosco #226' = sun.spect * rosclo.mspect$UV_filter_EColour226,
    PET = sun.spect * mcdermit.mspect$Autostat_CT5_125um),
    idfactor = "filter")

plot(all.spect, annotations = c("colour.guide", "boxes", "labels")) + facet_grid(filter~.)
```



If we would like the different spectra plotted on a single panel, we can use, for example, package `ggplot2`.

```
ggplot(data=all.spct, aes(x=w.length, y=s.e.irrad, colour=filter)) + geom_line()
```

