photobiologySensors Version 0.1.6
Catalogue of Sensors

Pedro J. Aphalo

June 7, 2014

1 Introduction

We will plot the spectral response of the different sensors for which data is provided in the pacakge. We plot side-by-side the response to energy (i.e. the electrical output that would be expected at each wavelengths with a source emitting equal spectral energy irradiance at all wavelengths) and the response to photons (i.e. as above but with a source emitting equal spectral photon irradiance at all wavelengths). All responses are normalized to an area of one under the whole curve.

```
library(ggplot2)
library(photobiologygg)

## Loading required package:  photobiology
## Loading required package:  data.table
## Loading required package:  lubridate
##
## Attaching package:  'lubridate'
##
## The following objects are masked from 'package:data.table':
##
##    hour, mday, month, quarter, wday, week, yday, year
## Warning:  replacing previous import by 'lubridate::hour' when loading 'photobiology'
## Warning:  replacing previous import by 'lubridate::mday' when loading 'photobiology'
## Warning:  replacing previous import by 'lubridate::month' when loading 'photobiology'
## Warning:  replacing previous import by 'lubridate::quarter' when loading 'photobiology'
## Warning:  replacing previous import by 'lubridate::wday' when loading 'photobiology'
## Warning:  replacing previous import by 'lubridate::week' when loading 'photobiology'
## Warning:  replacing previous import by 'lubridate::yday' when loading 'photobiology'
## Warning:  replacing previous import by 'lubridate::year' when loading 'photobiology'
## Loading required package:  proto
## Loading required package:  splus2R
## Loading required package:  plyr
##
## Attaching package:  'plyr'
##
## The following object is masked from 'package:lubridate':
##
##    here
```

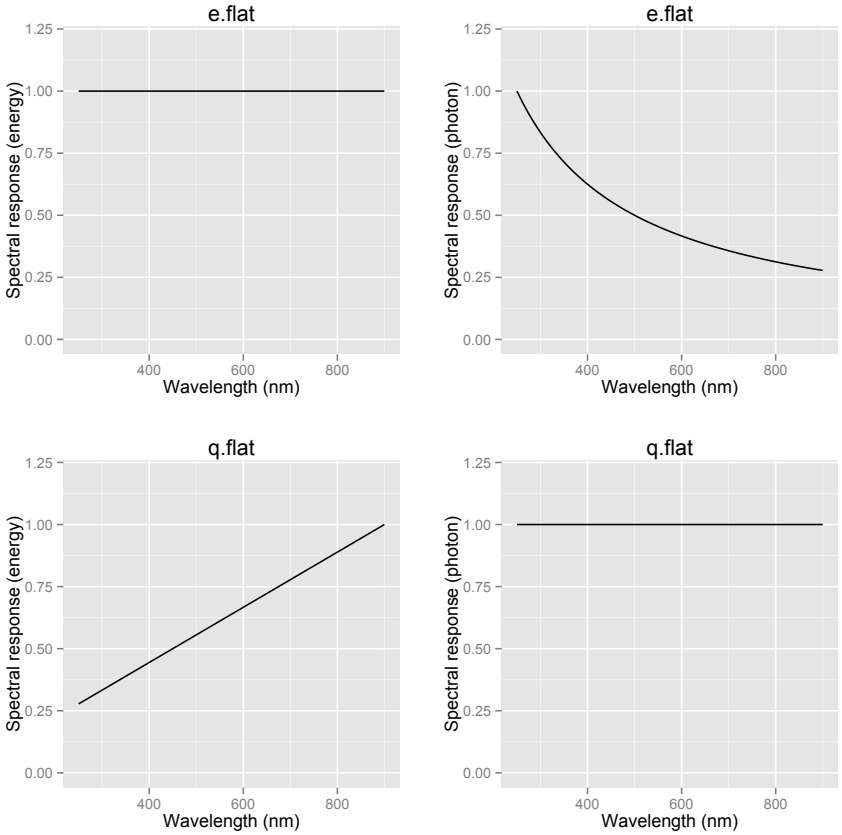
```
library(photobiology)
library(photobiologySensors)
```

We define a function to do the actual plotting so as to not repeat code, and to make changes easier in the future.

```
sensor.plotter <-
  function(sensor.name, w.low=250.0, w.high=900.0, scaled="peak"){
    w.length.out <-
      seq(from=w.low, to=w.high, length.out=300)
    e.spectrum.data <-
      calc_sensor_multipliers(w.length.out=w.length.out,
                              sensor.name=sensor.name,
                              unit.out="energy", scaled=scaled)
    q.spectrum.data <-
      calc_sensor_multipliers(w.length.out=w.length.out,
                              sensor.name=sensor.name,
                              unit.out="photon", scaled=scaled)
    e.spectrum.data <-
      na.omit(e.spectrum.data)
    q.spectrum.data <-
      na.omit(q.spectrum.data)
    fig_energy <-
      ggplot(aes(x=w.length, y=response), data=e.spectrum.data) +
      xlim(w.low, w.high) + ylim(0.0, 1.2) +
      labs(x="Wavelength (nm)", y="Spectral response (energy)",
           title=sensor.name) +
      geom_line() +
      stat_peaks(hjust=-0.5, angle=90, span=5,
                 ignore_threshold=0.1)
    fig_photon <-
      ggplot(aes(x=w.length, y=response), data=q.spectrum.data) +
      xlim(w.low, w.high) + ylim(0.0, 1.2) +
      labs(x="Wavelength (nm)", y="Spectral response (photon)",
           title=sensor.name) +
      geom_line() +
      stat_peaks(hjust=-0.5, angle=90, span=5,
                 ignore_threshold=0.1)
    print(fig_energy)
    print(fig_photon)
  }
```

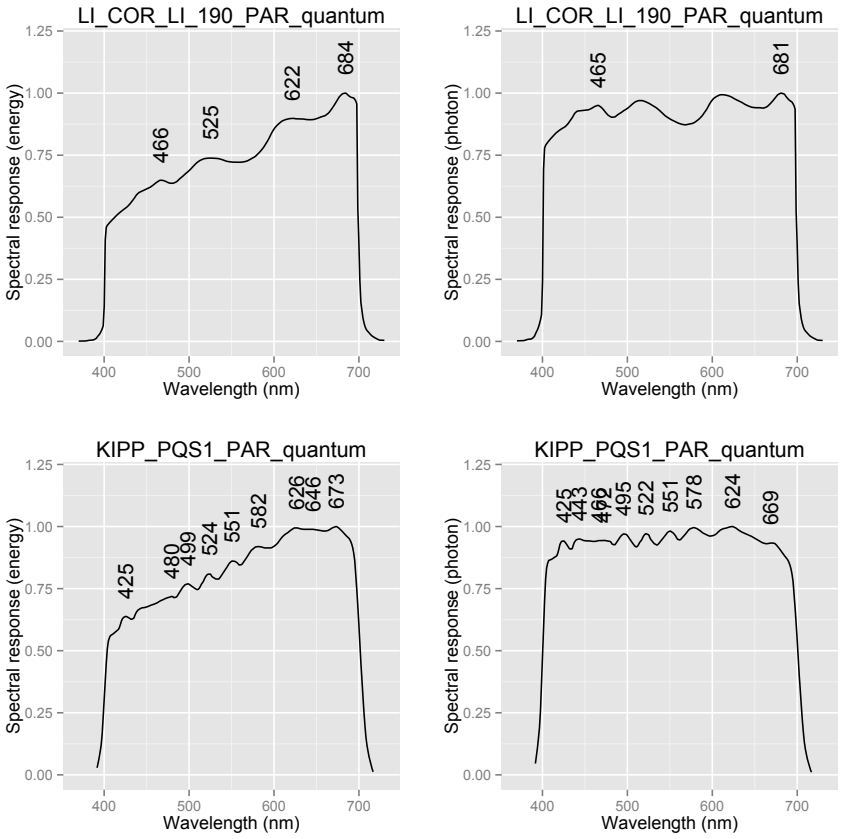
2 Flat responses

```
sensor.plotter("e.flat")
sensor.plotter("q.flat")
```



3 Quantum PAR sensors

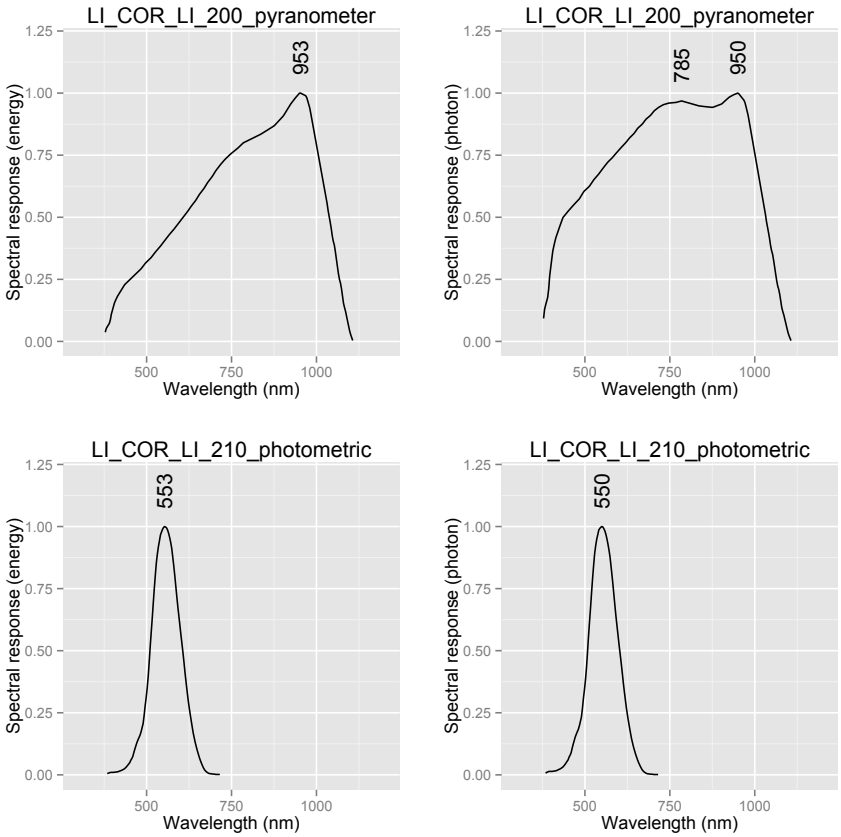
```
par.sensors <- c("LI_COR_LI_190_PAR_quantum", "KIPP_PQS1_PAR_quantum")
for (sensor in par.sensors) {
  sensor.plotter(sensor.name=sensor, w.low=370.0, w.high=730.0)
}
```



4 Other sensors

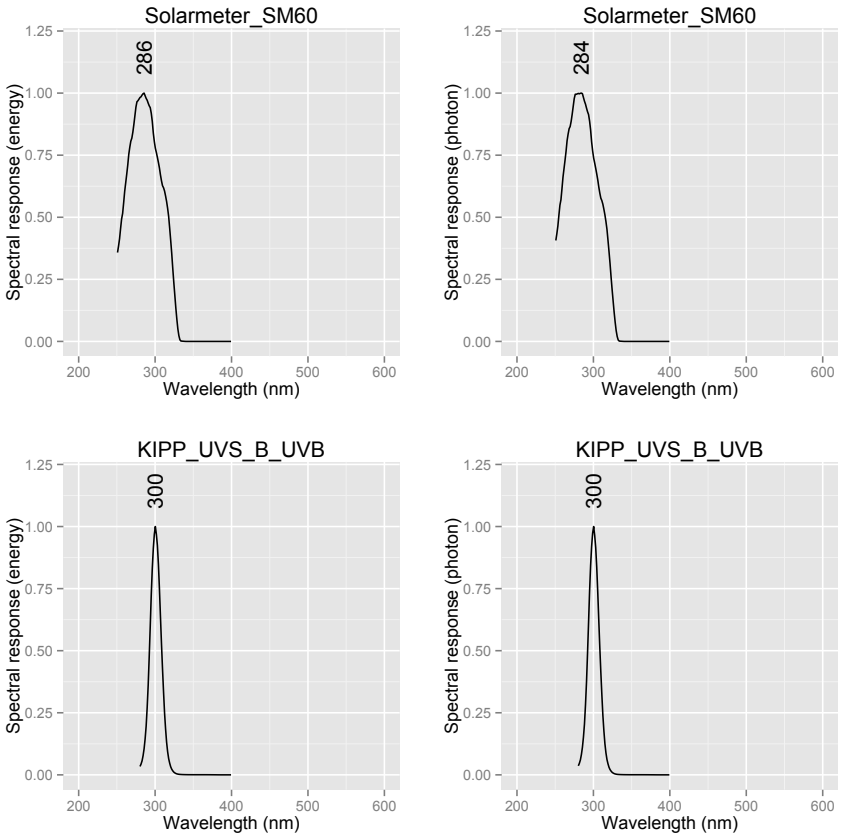
```
other.sensors <- c("LI_COR_LI_200_pyranometer", "LI_COR_LI_210_photometric")
for (sensor in other.sensors) {
  sensor.plotter(sensor.name=sensor, w.low=300.0, w.high=1200.0)
}
```

```
## Warning:  Warning:  wavelength values should be in nm
##  data contains values < 200 nm and/or > 1000 nm
## Warning:  Warning:  wavelength values should be in nm
##  data contains values < 200 nm and/or > 1000 nm
## Warning:  Warning:  wavelength values should be in nm
##  data contains values < 200 nm and/or > 1000 nm
## Warning:  Warning:  wavelength values should be in nm
##  data contains values < 200 nm and/or > 1000 nm
```



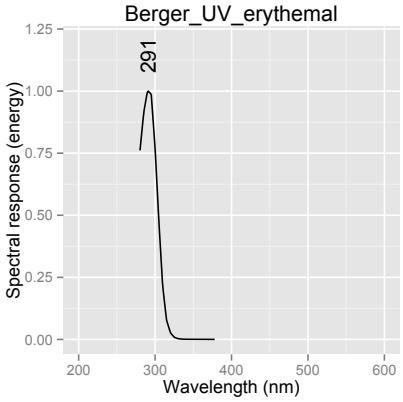
5 UVB sensors

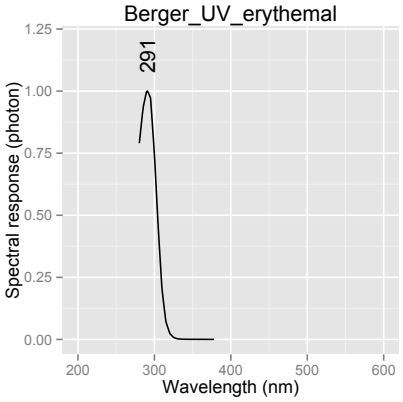
```
uvb.sensors <- c("Solarmeter_SM60", "KIPP_UVS_B_UVB")
for (sensor in uvb.sensors) {
  sensor.plotter(sensor, w.low=200, w.high=600)
}
```

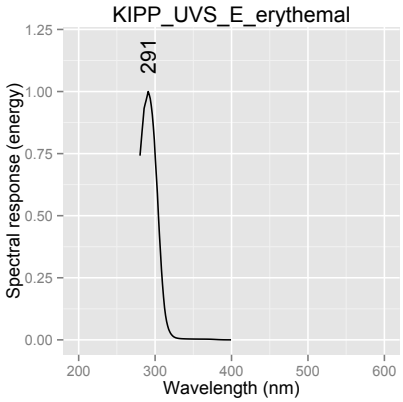


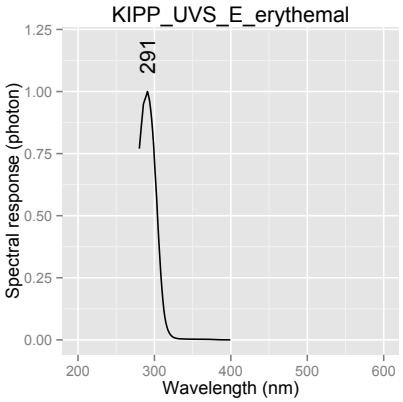
6 Erythemal UV sensors

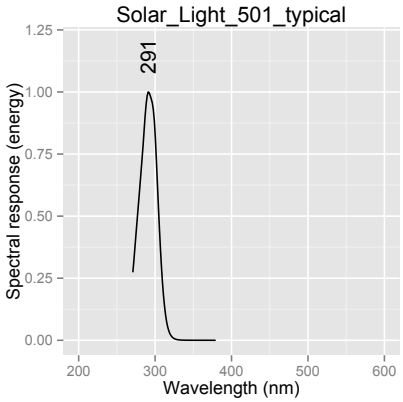
```
uv_ery.sensors <- c("Berger_UV_erythemal", "KIPP_UVS_E_erythemal",
                    "Solar_Light_501_typical", "Solar_Light_501_high_UVA", "Solar_Light_501_low_UVA",
                    "Vital_BW_20")
for (sensor in uv_ery.sensors) {
  sensor.plotter(sensor, w.low=200, w.high=600)
}
```

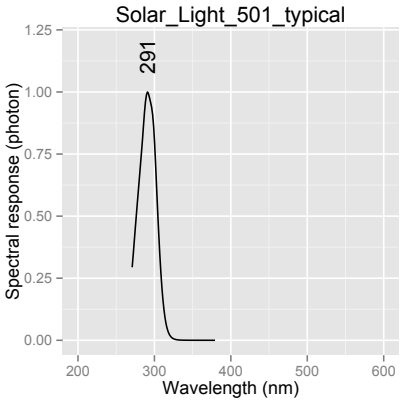

Berger_UV_erythemal
291
Spectral response (energy)
Wavelength (nm)

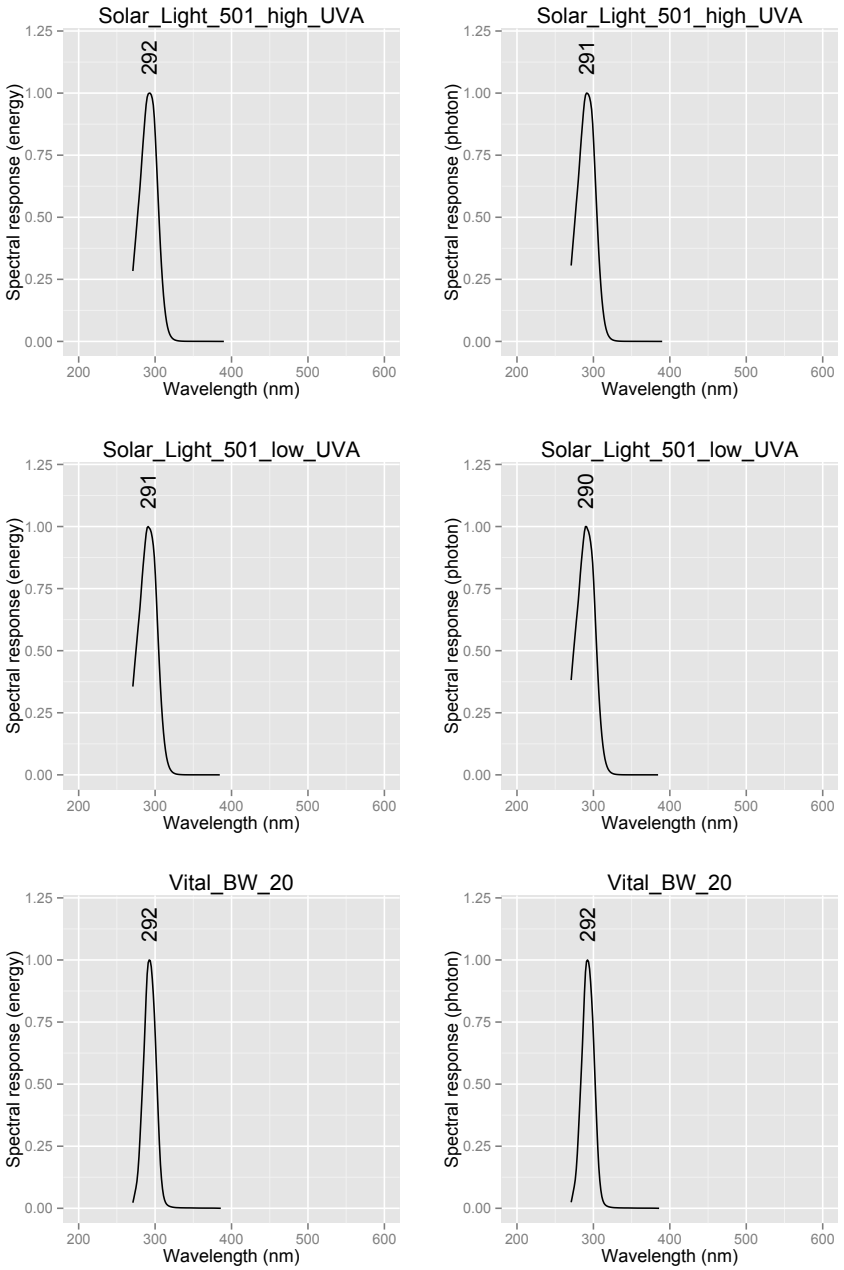

Berger_UV_erythemal
291
Spectral response (photon)
Wavelength (nm)


KIPP_UVS_E_erythemal
291
Spectral response (energy)
Wavelength (nm)


KIPP_UVS_E_erythemal
291
Spectral response (photon)
Wavelength (nm)

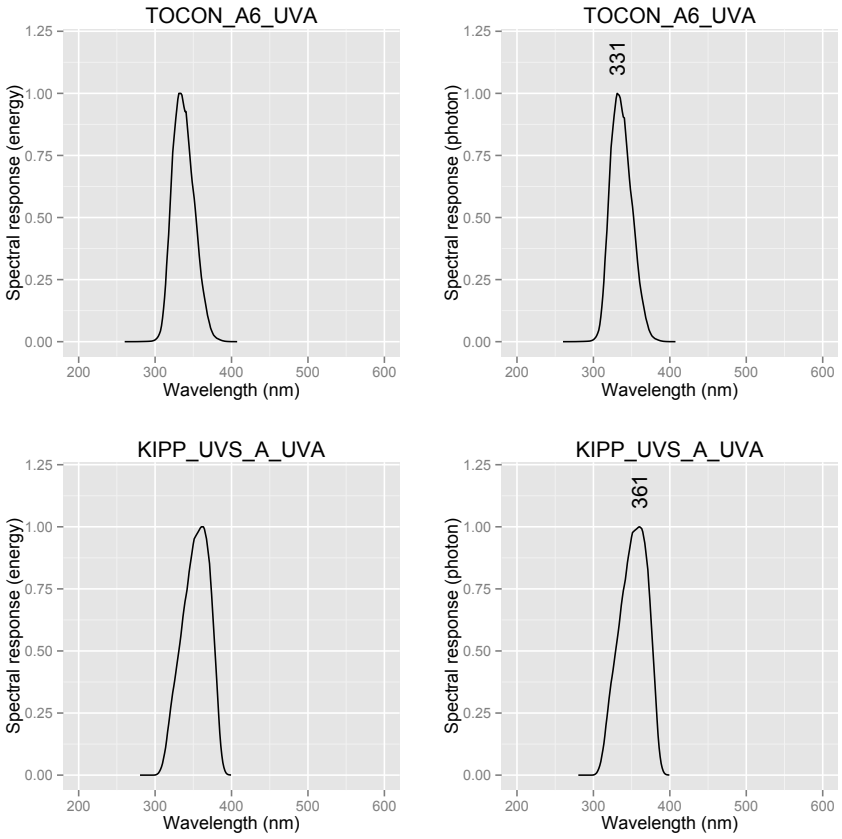

Solar_Light_501_typical
291
Spectral response (energy)
Wavelength (nm)


Solar_Light_501_typical
291
Spectral response (photon)
Wavelength (nm)



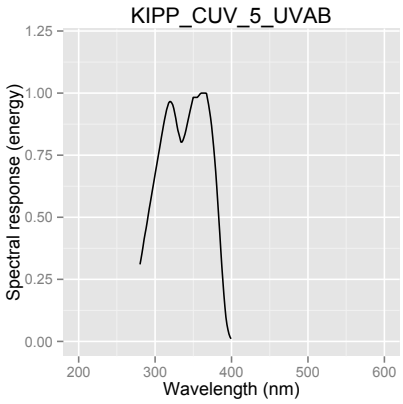
7 UVA sensors

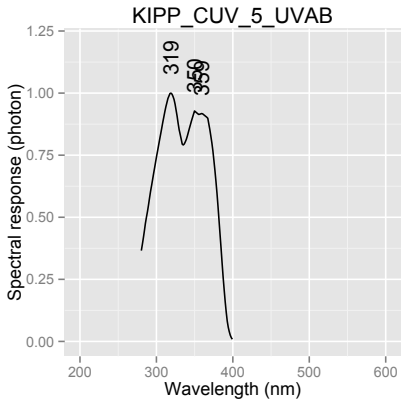
```
uva.sensors <- c("TOCON_A6_UVA", "KIPP_UVS_A_UVA")
for (sensor in uva.sensors) {
  sensor.plotter(sensor, w.low=200, w.high=600)
}
```



8 Broadband UV sensors

```
uvab.sensors <- c("KIPP_CUV_5_UVAB")
for (sensor in uvab.sensors) {
  sensor.plotter(sensor, w.low=200, w.high=600)
}
```





9 Blue sensors

```
b.sensors <- c("TOCON_blue4")
for (sensor in b.sensors) {
  sensor.plotter(sensor, w.low=200, w.high=600)
}
```