


■ Methods

Exploring temporal and latitudinal variation in the solar spectrum at ground level with the TUV model

Pedro J. Aphalo, ORCID: 0000-0003-3385-972X

ViPS, Organismal and Evolutionary Biology, University of Helsinki, Helsinki, Finland

DOI: 10.19232/uv4pb.2018.2.14 © 2018 The Author, licensed under 

The simulation of solar spectra at ground level

Radiation transfer models use as input the extraterrestrial solar spectrum. The spectrum at ground level depends on atmospheric composition, path length through the atmosphere and the presence of clouds and/or aerosols. The path length is determined by the solar elevation above the horizon and the position of the observed with respect to ground level and the elevation of the ground above sea level. Atmospheric composition can vary most significantly with respect to water vapour and ozone “columns”, but these data are not difficult to come by. In contrast, data on cloudiness expressed as cloud depth is difficult to obtain, mainly because it changes very rapidly. Consequently, for non-specialists the use of the TUV model is best restricted to simulations under clear sky conditions. In an earlier article in this Bulletin the model libRadtran (Emde et al. 2016) was introduced (Brelsford 2017). In the present article we will use the TUV model developed by Sasha Madronich, more specifically the web interface to this model called Quick TUV Calculator (http://cprm.acom.ucar.edu/Models/TUV/Interactive_TUV/).

The TUV model is a well known model of atmospheric chemistry that also simulates the solar spectrum within the atmosphere and at ground level using a radiation transfer approach. The model is written in FORTRAN and available for local use. When we need only to compute a few spectra with no special conditions, it is much easier to use the on-line interface provided by the National Centre of Atmospheric Research (NCAR) under the name of Quick TUV Calculator (http://cprm.acom.ucar.edu/Models/TUV/Interactive_TUV/) than to install the model programme.

The “quick” interface is menu/dialogue based and easy to use once one understands the different options and required input data. An annotated video captured during use of the Quick TUV calculator is included as supplementary material with permission of Sasha Madronich, creator of the TUV model.

The Quick TUV Calculator accepts as input either a solar zenith angle or geographic plus time coordinates. In the examples presented here I used zenith angles. The ‘photobiology’ package also includes functions for the calculation of the zenith and other angles for the sun position (see our previous article Aphalo 2016 and the R notebook in the supplemental files). The interface with the fields set to the values used for the simulations is shown in Figure 10.1.

NCAR | National Center for Atmospheric Research
UCAR | Atmospheric Chemistry Observations & Modeling

Modeling
Master Mechanism
TUV
WRF-Chem

QUICK TUV CALCULATOR

This web page runs the 5.3 version of the TUV model. You can run the model for a specified latitude, longitude and time (input option 1), or for a given solar zenith angle (input option 2). In either case, you must also specify the additional parameters in the second column. Also, you may select to print out the photolysis rates and/or the solar actinic flux spectrum at a given altitude above the surface (output option 1), or the erythemal UV and/or solar irradiance at that altitude (output option 2). For any problem, or to send comments, email [TUV administrators](#).

| Wavelength | OTHER INPUT PARAMETERS | Sunlight |
|--|--|--|
| Start: 280 End: 780 Increments: 500 | OVERHEAD OZONE COLUMN: 300 (du) SURFACE ALBEDO (0-1): 0.1 GROUND ELEVATION (km asl): 0 MEASUREMENT ALTITUDE (km asl): 0 | Direct beam: 1.0 Diffuse down: 1.0 Diffuse up: 0.0 |
| <input type="radio"/> INPUT OPTION 1 LATITUDE (deg): 0 LONGITUDE (deg): 0 DATE (YYYYMMDD): 20150630 TIME (hh:mm:ss, GMT): 12:00:00 <input checked="" type="radio"/> INPUT OPTION 2 SOLAR ZENITH ANGLE (deg): 0 | Clouds Opt. Depth: 0.00 Base: 4.00 Top: 5.00 Aerosols Opt. Depth: 0.235 S-S Alb: 0.990 Alpha: 1.000 | OUTPUT OPTION 1 (for Atmospheric Science) <input checked="" type="checkbox"/> MOLECULAR PHOTOLYSIS FREQUENCIES (s ⁻¹) <input type="checkbox"/> ACTINIC FLUX, SPECTRAL (quanta s ⁻¹ cm ⁻² nm ⁻¹) OUTPUT OPTION 2 (for Biology) <input type="checkbox"/> IRRADIANCE, WEIGHTED (W m ⁻²) <input checked="" type="checkbox"/> IRRADIANCE, SPECTRAL (W m ⁻² nm ⁻¹) |

RADIATION TRANSFER MODEL

Pseudo-spherical 2 streams (faster, less accurate)
 Pseudo-spherical discrete ordinate 4 streams (slower, more accurate)

GO!
[RESET](#)

The **wavelength** parameters are used to specify the operational range of the TUV Calculator. You can use wavelength start and end to focus on a particular chemical species. Set increments to **-156** for an auto-generated wavelength grid. For more detailed information, please read the [TUV Instructions](#).

If your usage of the TUV calculator is high-volume (greater than ~100 requests per day), we suggest that you [download the TUV FORTRAN code](#) and build an executable to perform the calculations locally on your own computer.

For video instructions on using the TUV Calculator, you may watch this external video [Using the Quick TUV Calculator](#), created by Pedro J. Aphalo at the University of Helsinki.

Figure 10.1: Quick TUV Calculator. User interface with settings as used to generate the data used for the examples shown in this article. To obtain the different spectra the value of the zenith angle was changed.

Using spectral data output by the Quick TUV Calculator

I will show how to import into R, compute summaries and plot the output from this model using the packages from the R for Photobiology suite. Package ‘photobiologyInOut’ was recently updated (versions $\geq 0.4.15$) to support the import of spectral data from files returned by the *Quick TUV Calculator*. We will read solar spectra into R, plot and calculate summary quantities from them. The examples chosen will be useful as reference data also to those not interested in doing new simulations. When high quality spectral measurements are lacking, simulated spectra can be very useful for the interpretation of results from field experiments, the design of realistic treatments or the assessment of the relevance of results of past experiments to specific exposure conditions outdoors.

Two aspects of the use of the packages in the R for Photobiology suite have been described in previous articles published in this Bulletin: the calculation of summaries from spectra (Aphalo 2015) and computations related to the position of the sun (Aphalo 2016). Here we will make use, in addition to the functions previously described, of a function from package ‘photobio-

logyInOut' that implements the import into R of the spectral data returned by the Quick TUV Calculator. We also use the plotting functions from package 'ggspectra'. Many functions in the packages earlier described now accept collections of spectra as arguments, which simplifies plotting or computations on multiple spectra. All the R packages used are available through public CRAN repository (<https://cran.r-project.org/>) and further information and on-line documentation can be found at the website <https://www.r4photobiology.info/>.

In the article itself we provide only partial examples of the R code, while fully reproducible source code and output is provided as supplementary material in the form of an R notebook readable with most web browsers. This R notebook includes the source code embedded which allows users to rebuild the HTML file and edit the source code as needed by opening the R notebook file in RStudio (<https://www.rstudio.com/>).

We start by loading the packages we will be using, and setting photon quantities as the default to use.

```
library(photobiology)
library(photobiologywavebands)
library(photobiologyInOut)
library(photobiologySun)
library(dplyr)
library(ggplot2)
library(ggalt)
library(ggspectra)
library(knitr)

theme_set(theme_bw(13))
photon_as_default()
```

Importing spectral data

When running a simulation in the Quick TUV Calculator, the spectral data and/or summaries are returned as a simple web page. A web page is a text file, and can be saved from the web browser. Depending on the browser it may be possible to save the page as a text file without HTML header and footer and using .txt and file ending. In all cases it should be possible to save the output as an HTML file. In either case the files can be imported into R with function `read_qtuv_txt()` just by providing the file name as argument. As much metadata as possible is extracted from the file, while the HTML header is stripped if present before attempting the decoding of data and meta-data. If summaries are present in addition to the spectral data in the Quick TUV Calculator output, these are also discarded.

```
qtuv.spct <-
  read_qtuv_txt("tuv-zenith-00-03-300.html")
head(qtuv.spct, 4)

## Object: source_spct [4 x 7]
## Wavelength range 280.5 to 283.5 nm, step 1 nm
## Label: Quick TUV spectral simulation File: tuv-zenith-00-03-300.html
## Measured on 2015-06-30 UTC
## Time unit 1s
```

```
##
## # A tibble: 4 x 7
##   w.length s.e.irrad s.e.irrad.dir s.e.irrad.diff.~ s.e.irrad.diff.~ angle
## *   <dbl>   <dbl>         <dbl>         <dbl>         <dbl> <dbl>
## 1     280.  3.07e-15      1.76e-15      1.31e-15      3.07e-16  0
## 2     282.  1.17e-13      6.59e-14      5.14e-14      1.17e-14  0
## 3     282.  1.84e-12      1.01e-12      8.23e-13      1.84e-13  0
## 4     284.  9.92e-12      5.42e-12      4.51e-12      9.92e-13  0
## # ... with 1 more variable: date <dtm>

cat(comment(qtuv.spct))

## SPECTRAL IRRADIANCE (W m-2 nm-1)
## from file:  tuv-zenith-00-03-300.html  generated by Quick TUV on
## 2018-12-15 07:41:23 ozone column (DU) =
## zenith angle (degrees) = 0
## altitude (km) = 0
## observer elev. = 0
```

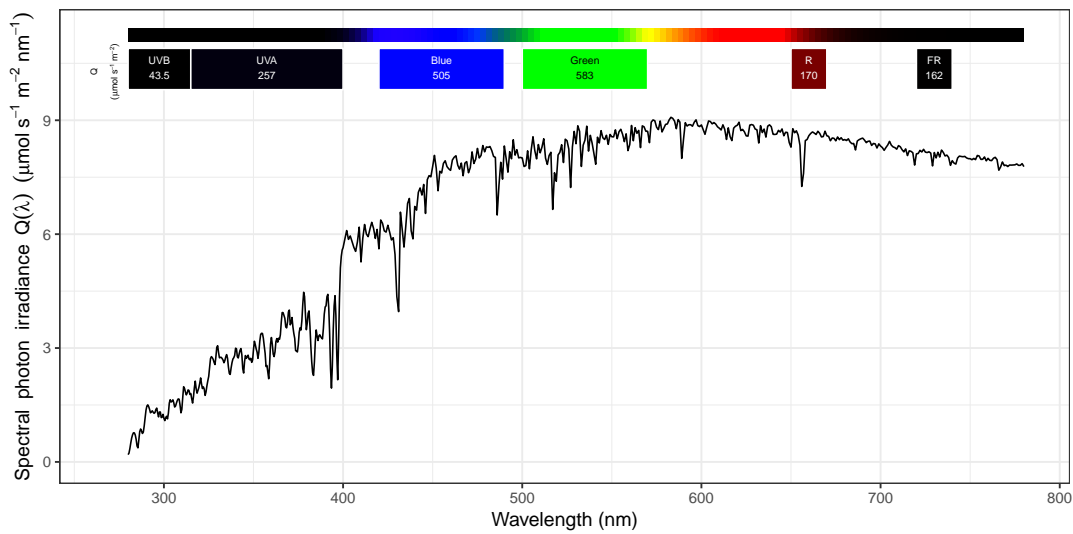
The example in the supplementary R notebook shows how to read multiple files into a collection of spectra using iteration. The result of running that code was saved to an R data file, which will be loaded below.

Plotting

In addition to simulations produced ad-hoc, existing measured spectra or standardized spectra can be useful as references and/or as examples. For example, it can be useful to compare the extraterrestrial solar spectrum and spectra at ground level. Package ‘photobiologySun’ provides spectral data for both of these types of spectra, with several measured and simulated sets of spectral data. We use these data only for the initial example but later rely on new simulations done with the Quick TUV Calculator. Although not shown here, package ‘photobiologySun’ also contains example data measured in the understory of different forests.

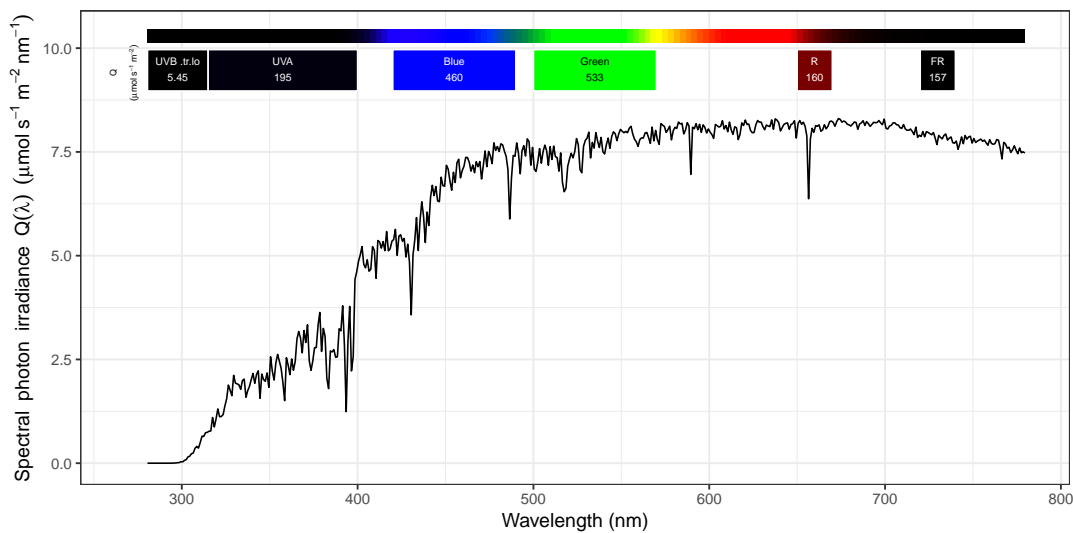
We here plot the extraterrestrial solar spectrum, which is an input to all radiation transfer models. The data used for this figure are from package ‘photobiologySun’.

```
sun_ET.spct <- sun_reference.mspct[["Gueymard.AM0"]]
plot(sun_ET.spct,
     range = c(250, 780), annotations = c("-", "peaks"),
     w.band = c(Plant_bands(), IR_bands())) +
  theme(legend.position = "top")
```



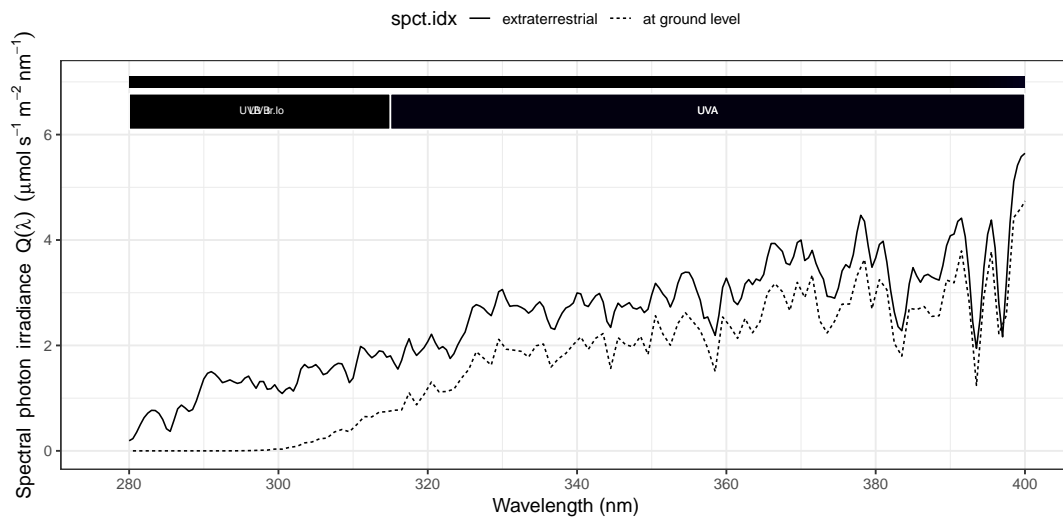
We can plot the simulated spectrum read above from the Quick TUV Calculator output in the previous section.

```
plot(qtuv.spct,
     range = c(250, 780),
     annotations = c("-", "peaks"),
     w.band = c(Plant_bands(), IR_bands())) +
theme(legend.position = "top")
```



To highlight the effect of the atmosphere and solar elevation we can add to the plot the extraterrestrial spectrum, while also reducing the range of wavelengths plotted.

```
plot(source_mspct(list(extraterrestrial = sun_ET.spct, "at ground level" = qtuv.spct)),
     range = c(250, 400),
     annotations = c("-", "peaks"),
     w.band = c(Plant_bands(), IR_bands())) +
theme(legend.position = "top")
```



We can use models to simulate the solar spectrum at a specific geographical location and time point. We next load data for several spectra simulated with the Quick TUV Calculator and originally imported into R as demonstrated in the previous section and re-saved into an .Rda R data file (see the supplementary R Notebook for details).

```
load("QTUV-spectra.Rda")
length(sun_tuv.mspct)
```

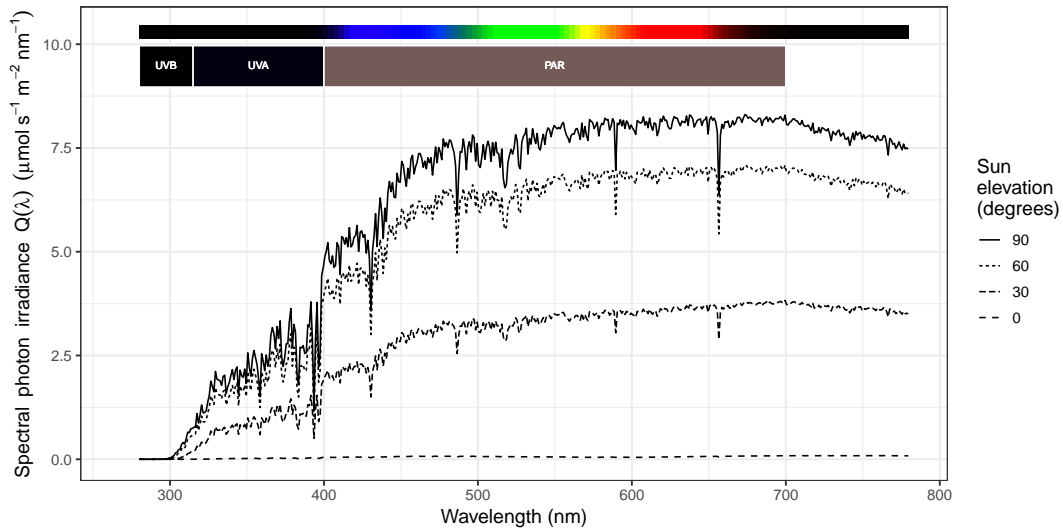
```
## [1] 29
```

In the next figure we present spectral irradiance on a horizontal surface for the sun at different elevations above the horizon, assuming an ozone column of 300 DU (Dobson units). In the TUV model the zenith angle is used to describe solar elevation. The collection of spectra `sun_tuv.mspct` contains 29 spectra, and we use subscripting by name to select those we want to plot. To save typing we create two vectors with the names of the spectra we will use in different examples. We will use these vectors to select subsets of spectra from the collection of spectra.

```
spectra7elevations <- c("tuv_zenith_00_03_300",
                       "tuv_zenith_15_03_300",
                       "tuv_zenith_30_03_300",
                       "tuv_zenith_45_03_300",
                       "tuv_zenith_60_03_300",
                       "tuv_zenith_75_03_300",
                       "tuv_zenith_90_03_300")
the7elevations <- 90 - c(0, 15, 30, 45, 60, 75, 90)

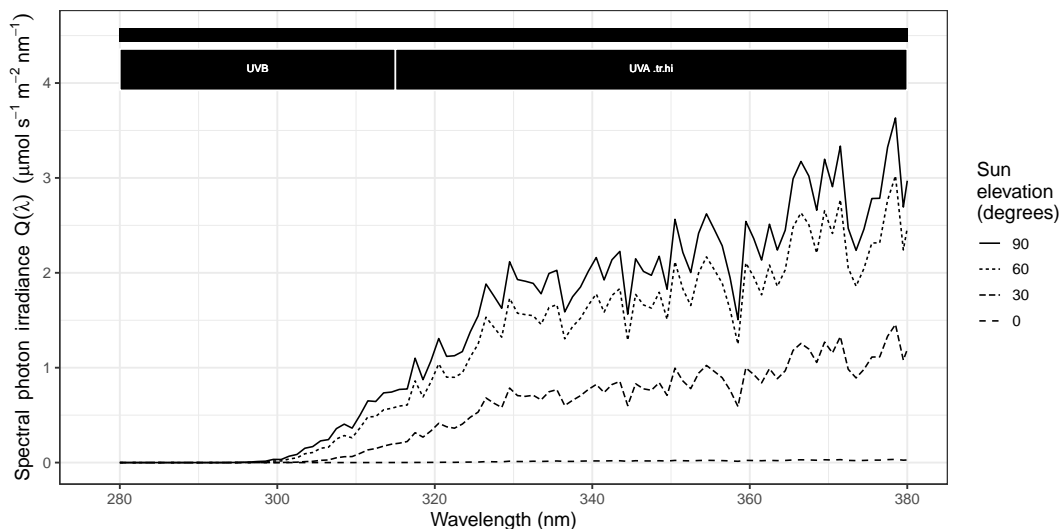
spectra4elevations <- c("tuv_zenith_00_03_300",
                       "tuv_zenith_30_03_300",
                       "tuv_zenith_60_03_300",
                       "tuv_zenith_90_03_300")
the4elevations <- 90 - c(0, 30, 60, 90)
```

```
plot(sun_tuv.mspct[spectra4elevations],
     annotations = c("-", "peaks")) +
scale_linetype(labels = the4elevations,
               name = "Sun\nelevation\n(degrees)")
```



To obtain a detailed plot of the UV region we pass the desired range of wavelengths in nanometres range = c(280, 380). Except for this the code below is the same as in the previous chunk.

```
plot(sun_tuv.mspct[spectra4elevations],
     range = c(280, 380),
     annotations = c("-", "peaks")) +
scale_linetype(labels = the4elevations,
               name = "Sun\nelevation\n(degrees)")
```

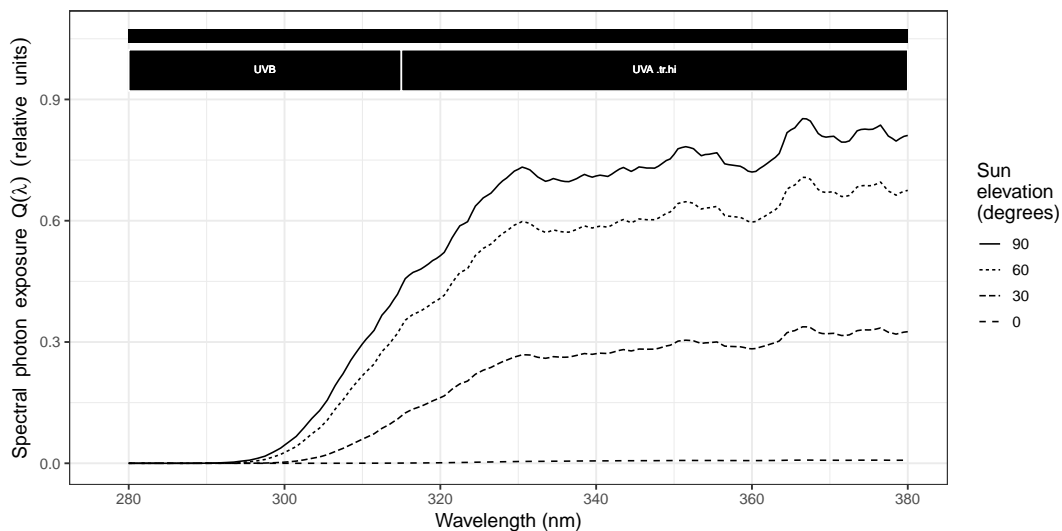


Although plotting the spectra as shown above illustrates the overall change in spectral composition, these plots are difficult to read. We will modify the previous figure expressing the irradiance at ground level as a fraction of the extraterrestrial one. There is a problem, though:

the spectra have different wavelength resolution, something that would lead to artifacts when ratios are computed. Consequently, we first smooth the spectra. We do this in separate steps for clarity.

```
tuv_smoothed.mspect <-
  mmsply(sun_tuv.mspect[spectra4elevations],
    smooth_spct, method = "supsmu")
extraterrestrial_smoothed.spct <-
  smooth_spct(sun_ET.spct,
    method = "supsmu")
sun_relative <-
  convolve_each(tuv_smoothed.mspect,
    extraterrestrial_smoothed.spct,
    oper = `/`)
sun_relative <-
  mmsply(sun_relative, setScaled, scaled = TRUE)
```

```
plot(sun_relative,
  range = c(280, 380),
  annotations = c("-", "peaks")) +
  scale_linetype(labels = the4elevations,
    name = "Sun\nelevation\n(degrees)")
```



Computing summary quantities

Photon irradiance

```
irradiances.tb <-
  q_irrad(sun_tuv.mspect[spectra7elevations],
    w.band = list("UV-B" = UVB(), "UV-A" = UVA(), PAR = PAR()),
    scale.factor = 1e6)
```



```
irradiances.tb$elevation <- the7elevations
```

```
kable(irradiances.tb, digits = c(0, 2, 0, -1, 1))
```

| spct.idx | UV-B | UV-A | PAR | elevation |
|----------------------|------|------|------|-----------|
| tuv_zenith_00_O3_300 | 5.45 | 195 | 2220 | 90 |
| tuv_zenith_15_O3_300 | 5.04 | 186 | 2130 | 75 |
| tuv_zenith_30_O3_300 | 3.91 | 161 | 1880 | 60 |
| tuv_zenith_45_O3_300 | 2.40 | 122 | 1480 | 45 |
| tuv_zenith_60_O3_300 | 1.01 | 76 | 970 | 30 |
| tuv_zenith_75_O3_300 | 0.19 | 31 | 420 | 15 |
| tuv_zenith_90_O3_300 | 0.00 | 2 | 20 | 0 |

Biologically effective irradiance

```
be_irradiances.tb <-
```

```
  e_irrad(sun_tuv.mspct[spectra7elevations],
    w.band = list(GPAS = GEN_G(), PG = PG(), CIE98 = CIE()))
```

```
be_irradiances.tb$elevation <- the7elevations
```

```
kable(be_irradiances.tb, digits = 2)
```

| spct.idx | GPAS | PG | CIE98 | elevation |
|----------------------|------|------|-------|-----------|
| tuv_zenith_00_O3_300 | 0.45 | 1.67 | 0.30 | 90 |
| tuv_zenith_15_O3_300 | 0.40 | 1.58 | 0.27 | 75 |
| tuv_zenith_30_O3_300 | 0.29 | 1.31 | 0.20 | 60 |
| tuv_zenith_45_O3_300 | 0.15 | 0.95 | 0.12 | 45 |
| tuv_zenith_60_O3_300 | 0.05 | 0.55 | 0.05 | 30 |
| tuv_zenith_75_O3_300 | 0.01 | 0.21 | 0.01 | 15 |
| tuv_zenith_90_O3_300 | 0.00 | 0.01 | 0.00 | 0 |

Or expressed as a fraction relative to the value for the sun at the zenith.

```
be_irradiances.tb %>%
```

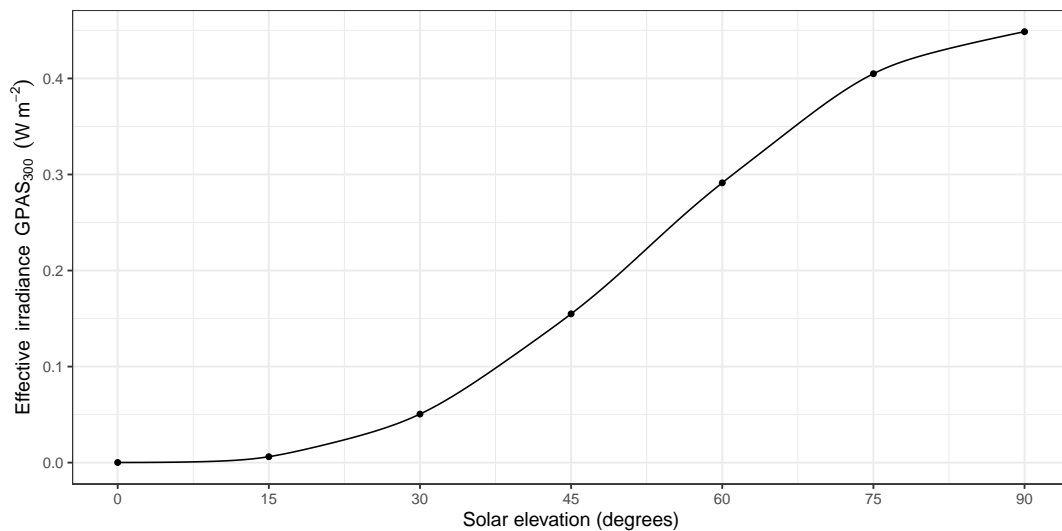
```
  mutate(GPAS = GPAS / max(GPAS), PG = PG / max(PG), CIE98 = CIE98 / max(CIE98)) %>%
```

```
  kable(digits = 2)
```

| spct.idx | GPAS | PG | CIE98 | elevation |
|----------------------|------|------|-------|-----------|
| tuv_zenith_00_O3_300 | 1.00 | 1.00 | 1.00 | 90 |
| tuv_zenith_15_O3_300 | 0.90 | 0.94 | 0.91 | 75 |
| tuv_zenith_30_O3_300 | 0.65 | 0.79 | 0.69 | 60 |
| tuv_zenith_45_O3_300 | 0.35 | 0.57 | 0.41 | 45 |
| tuv_zenith_60_O3_300 | 0.11 | 0.33 | 0.17 | 30 |
| tuv_zenith_75_O3_300 | 0.01 | 0.13 | 0.04 | 15 |
| tuv_zenith_90_O3_300 | 0.00 | 0.01 | 0.00 | 0 |

We can also plot the values for one of the BSWFs, here GPAS.

```
ggplot(be_irradiances.tb, aes(x = elevation, y = GPAS)) +
  geom_point() +
  geom_xspline(spline_shape = -0.5) +
  scale_x_continuous(name = "Solar elevation (degrees)",
                    breaks = the7elevations) +
  scale_y_continuous(name =
    expression(Effective~irradiance~GPAS[300]~(W~m^{-2})))
```



Photon ratio

```
uvb_par_ratio.tb <-
  q_ratio(sun_tuv.mspect[spectra7elevations],
          w.band.num = UVB(),
          w.band.denom = PAR())

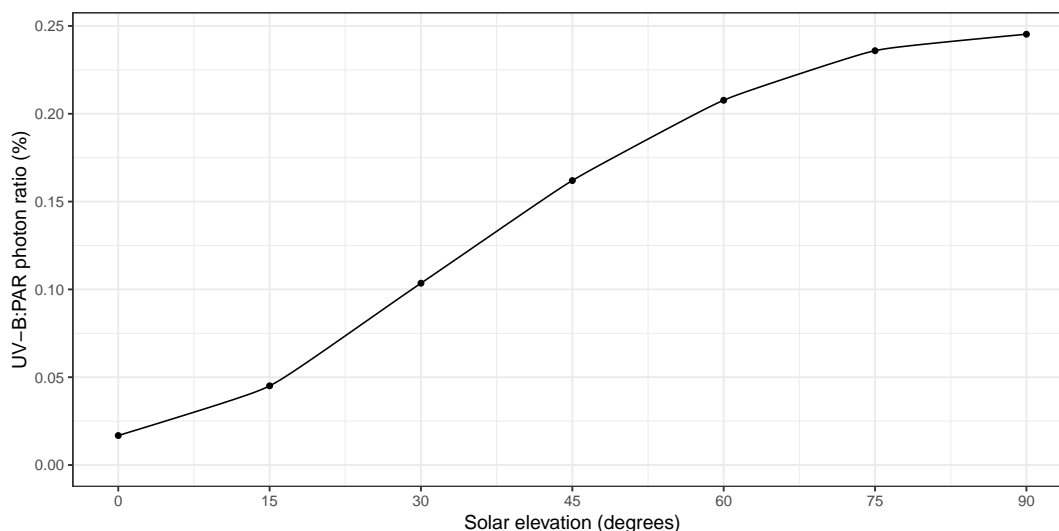
head(uvb_par_ratio.tb)

## # A tibble: 6 x 2
##   spct.idx      `q_ratio_UVB.ISO:PAR(q:q)`
##   <fct>          <dbl>
## 1 tuv_zenith_00_03_300      0.00245
## 2 tuv_zenith_15_03_300      0.00236
## 3 tuv_zenith_30_03_300      0.00208
## 4 tuv_zenith_45_03_300      0.00162
## 5 tuv_zenith_60_03_300      0.00104
## 6 tuv_zenith_75_03_300      0.000451

uvb_par_ratio.tb$elevation <- the7elevations

names(uvb_par_ratio.tb)[2] <- "UV-B:PAR"
```

```
ggplot(uvb_par_ratio.tb,
  aes(x = elevation,
      y = `UV-B:PAR` * 100)) +
  geom_point() +
  geom_xspline() +
  scale_x_continuous(name = "Solar elevation (degrees)",
    breaks = the7elevations) +
  scale_y_continuous(name = "UV-B:PAR photon ratio (%)", limits = c(0, NA))
```



Conclusions and caveats

Using the Quick TUV Calculator it is easy to obtain spectral data for clear-sky conditions. Including the effect of clouds is more difficult, as cloud depth data is rarely easily available. Having spectral data, as opposed to UVI or other summary data, allows the calculation of diverse derived quantities and summaries. Such calculations are best done using scripts to ensure reproducibility. The R packages in the R for photobiology suite aim at easing these computations while providing maximum flexibility. In the case of biologically effective exposures and irradiances, even for a single BSWF several alternative mathematical formulations are in use as well as different approaches to extrapolation. The defaults used in the 'photobiology' package differ from those used by the TUV model for the calculation of these values, which can result in apparent discrepancies. For example the formulation for GPAS in TUV is that of Micheletti which differs slightly from Green's formulation, but both of these differ quite much from Thimijan's formulation (see Aphalo et al. 2012, for details, equations and original references).

Supplementary material

File QTUV-video.mp4: An instructional video on how to use the Quick TUV web interface is provided to help new users. File QTUV-notebook.html: An R notebook with additional code examples and their output, viewable with a web browser. The embedded source code can be

most easily accessed and edited using RStudio. File `QTUV-simulations.Rda`: Spectral data generated with Quick TUV Calculator, imported into R and re-saved as R objects.

The *R notebook* file contains worked out examples of reading into R several spectra downloaded from the Quick TUV site, plotting these spectra, their manipulation and the calculation of summary quantities. The HTML file contains brief text explanations, readable code listings and the rendered output. The actual source file used to produce the HTML file is embedded and can be extracted, modified and reused.

References

- Aphalo, P. J., A. Albert, L. O. Björn, A. R. McLeod, T. M. Robson, and E. Rosenqvist, eds. (2012). *Beyond the Visible: A handbook of best practice in plant UV photobiology*. 1st ed. COST Action FA0906 "UV4growth". Helsinki: University of Helsinki, Department of Biosciences, Division of Plant Biology, pp. xxx + 174. ISBN: ISBN 978-952-10-8363-1 (PDF), 978-952-10-8362-4 (paperback).
- Aphalo, P. J. (2015). "The r4photobiology suite: spectral irradiance". In: *UV4Plants Bulletin* 2015.1, pp. 19-27. ISSN: ISSN 2343-323X. DOI: [10.19232/uv4pb.2015.1.14](https://doi.org/10.19232/uv4pb.2015.1.14).
- (2016). "The r4photobiology suite: sun angles and day length". In: *UV4Plants Bulletin* 2016.1, pp. 29-39. ISSN: 2343-323X. DOI: [10.19232/uv4pb.2016.1.15](https://doi.org/10.19232/uv4pb.2016.1.15).
- Brelsford, C. C. (2017). "Radiative transfer theory and modelling with libRadtran". In: *UV4Plants Bulletin* 2016.2, pp. 45-50. ISSN: 2343-323X. DOI: [10.19232/uv4pb.2016.2.15](https://doi.org/10.19232/uv4pb.2016.2.15).
- Emde, C., R. Buras-Schnell, A. Kylling, B. Mayer, J. Gasteiger, U. Hamann, J. Kylling, B. Richter, C. Pause, T. Dowling, et al. (2016). "The libRadtran software package for radiative transfer calculations (version 2.0.1)". In: *Geoscientific Model Development* 9.5, pp. 1647-1672. DOI: [10.5194/gmd-9-1647-2016](https://doi.org/10.5194/gmd-9-1647-2016).

Editorial-board-reviewed article.

Published on-line on 2018-12-27.

Edited by: Titta Kotilainen.