


## ■ Methods

# The r4photobiology suite: spectral irradiance

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

ViPS, Department of Biosciences, University of Helsinki, Helsinki, Finland

DOI: 10.19232/uv4pb.2015.1.14 © 2015 The Author, licensed under 

## Introduction

R is a language designed for data analysis, and it has become in recent years the most popular software for advanced statistical analysis. With some delay, it is now also being incorporated into most university level curricula (?) and even in some cases being taught in high schools (Dennis 2013). Nowadays it is also widely used in financial institutions, industry and the big players in the cloud services. Microsoft has bought *Revolution Analytics* and has released its own distribution of R, called *Revolution R Open* (RRO). Currently R is world's most widely used statistics programming language (?).

I started using R more than 15 years ago. At those times R was not as popular as it is nowadays and most people considered it to be too difficult to be taught at the undergraduate level. By year 2001 I was already using exclusively R for data analyses related to my own research. Teaching SPSS felt awkward to me: why use in teaching software that I did not find suitable for my own research? This is how my first course specifically on R, and the use of R in my statistics courses started. At the time I was at the University of Jyväskylä.

Many scripts and bits of R code my students and myself were using for analysing spectral data accumulated over these nearly 20 years. By the time *Beyond the Visible: A handbook of best practice in plant UV photobiology* (2012) was published I had started organizing these bits of R code. Simultaneously with this handbook, I wrote an R package called *uvcalc*, which was described in it.

In several of the training schools organized by the UV4Growth COST action FA0906 "UV4Growth" we taught how to do calculations related to the quantification of UV radiation using R. This provided very good feedback that allowed me to design a much better user interface. In parallel, in a project I am collaborating with T. Matthew Robson, we need to process close to a million measured radiation spectra, which pushed me to optimize calculation speed. Early on it became clear that the use of the algorithms we were using was not limited to UV radiation or plants, which lead to a broadening of the aims of this development.

During these three years both the user interface and how calculations are implemented in the R code have been improved. In addition the range of calculations implemented has been greatly expanded. Given the amount code and data now available, they are now split between several different R packages, conforming a suite. I call it the R4Photobiology suite of packages (Aphalo 2015a). The core packages are already well tested and stable, so I have presented the suite at two conferences: a talk at the UseR! 2015 conference (Aphalo 2015b) and a poster at the European Society for Photobiology Congress (Aphalo 2015c). At the UseR! 2015 conference I received very useful comments that lead to the addition of new functionality into the packages.

The main focus of the suite is the processing of spectral irradiance data required for quantification of radiation in photobiological research. There is in addition support for calculations related to tri-chromic vision and the position of the sun, such as day and night length. In the section I describe the data flow and how it relates to the calculations implemented in the suite.

## Data flow

In this note I focus on spectral irradiance measurements, however, the suite includes functions for summarizing and combining different types of spectral data. It also has some support for estimating responses by organisms. These include photoreceptor-related calculations and vision in humans and other organisms. Functions for calculating the position of the sun, and day and night length are also included.

In broad terms the steps are A) acquisition of raw spectral data, B) processing into calibrated and validated spectral data, including exploratory analysis of spectra, C) summarizing of the spectral data, including spectral weighting, integration over wavelengths, smoothing, feature extraction; D) statistical analysis. The suite mainly focuses on steps B) and C) (Fig. 8.1).

**A** In the case of spectral irradiance, the first step is the acquisition of data, either with a spectroradiometer or by simulation with a model. The suite includes functions for reading output files produced by Ocean Optics instruments (OceanView, SpectraSuite), Macam instruments, Avantes instruments, and the old LI-COR spectroradiometer. Import of data output from TUV and LibRadTrans models is also possible. In addition, R can read natively .csv files (comma separated values) and with contributed packages out from MS-Excel and many other data formats. Usually calibrations are applied in the instrument firmware, or in software supplied with the instrument. At the moment support in the suite for these steps is limited to what we use ourselves because these steps tend to vary from instrument to instrument and in time.

**B** Classes for storing spectral data, definitions of operators and functions for mathematical operations for transforming spectra and for operations between spectra and between spectra and numerical data, and functions for automatic plotting of spectra are specially important for this step. Much of the code in the suite is needed to provide these facilities.

**C** “Apply” methods to apply arbitrary R functions to spectral data, functions for integration of spectral data over wavelengths, for smoothing and normalizing spectral data are supplied for this step. In addition to general-purpose functions for integration, functions for the calculation of specific summary quantities like irradiance and fluence are frequently used at this step.

**D** Statistical analysis is the core of R, so base R functions and those in other contributed R packages can be used at this step. The suite does not add to these.

## Design

I had two main aims guiding my design decisions for the suite: 1) robust data handling and reproducibility, 2) achieving easy of use through a consistent interface. The first aim is achieved by incorporating “sanity tests” for the data and by storing metadata, together with the data. So as to simplify the user interface, the data is always stored SI units without scale factors, e.g.  $\text{mol m}^{-2} \text{s}^{-1}$  instead of  $\mu\text{mol m}^{-2} \text{s}^{-1}$ . Algorithms have been carefully selected or designed so as to minimize rounding and interpolation errors, sometimes at the expense of performance.

Spectral data are stored in objects of classes defined in package *photobiology*. In these objects

data are stored in a consistent way—same quantity expressed in the same units—using always the same variable names for wavelengths and quantities. The hierarchy of classes used to store spectral data is shown in Figure 8.2. These classes are for individual spectra of a given type. Another hierarchy of classes is used to store collections of spectra—more precisely collections objects of classes defined to store individual spectra. They have related names. For example, in the case of spectral irradiance we have *source\_spct* for individual spectra and *source\_mspct* for collections of spectra of this type. Many methods are implemented with the same name for both individual spectra and for collections of spectra.

Another key concept used throughout is that of ‘waveband’. In the suite it is used with a broader meaning than usual: a ‘waveband’ always describes weights for a range of wavelengths, in the simplest case, weights are equal to one for observations within the range of interest, and zero otherwise. In the case of *effective* irradiance, weights can take arbitrary values, defined either by mathematical function or by tabulated multipliers. These weights are called in general *spectral weighting functions* (SWFs), or when derived from biological action spectra *biological spectral weighting functions* (BSWFs). The suite defines a class for storing such information, called *waveband*. Combining different spectral objects holding data, wavebands containing the weights to apply and functions defining integration procedures, it is possible to very flexibly quantify radiation. Furthermore, the user can create new flavours of any of these three types of objects. This approach gives consistency and orthogonality to the user interface, minimizing the number of ‘names’ that need to be remembered without sacrificing flexibility in what calculations can be done.

Additionally, the second aim is further supported by many different methods and functions having the same ‘argument signature’, in other words the position and names of formal arguments are the same across many functions. The use of classes, allows the coexistence of methods with the same name, which are automatically dispatched according to the type of spectral data. Finally implementing operators such as ‘+’ for spectra makes user code much simpler by removing the need of using loop constructs like *for* and *repeat* statements in user scripts.

## Packages

The suite consists in 12 packages, of which one is just a ‘loader’ of the other packages. All other packages depend on the one called *photobiology*. Other packages provide functionality or data specific to a subject area of research or a certain type

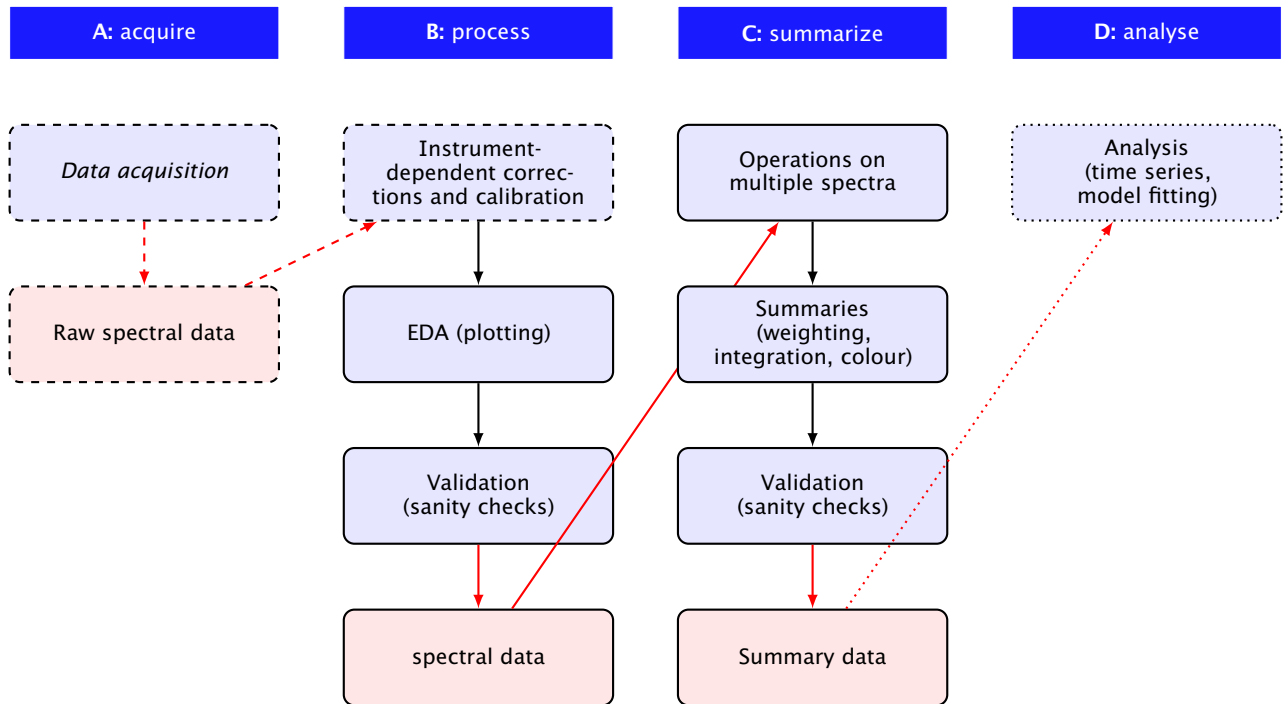


Figure 8.1: Diagram of the data flow for analysis of spectral irradiance.

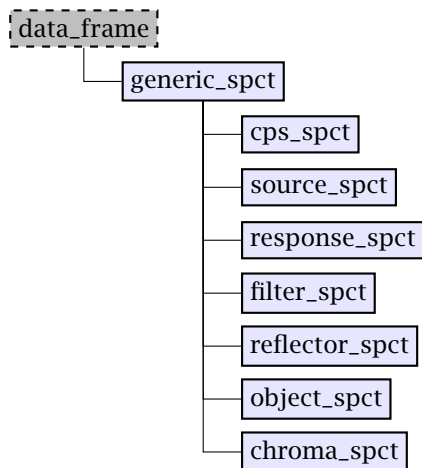


Figure 8.2: Classes for storage of data of different spectral quantities

of calculations. I expect that myself and other contributors will write new packages extending the suite.

The package at the core of the suite, called *photobiology*, implements generally useful data classes, methods and functions for photobiological calculations. It contains only the minimal amount of example data needed for examples in the documentation and code tests. As of current version 0.8.10, it exports 228 method and operator definitions—using less than 40 new names, 221 function definitions, and 17 classes.

Another broadly useful package is *photobiologygg* which implements plotting of objects of the spectral data classes defined in package *photobiology*. The plotting functions are extensions to Hadley Wickham’s package *ggplot2*, currently the most popular of R’s plotting systems. It defines only 9 new names, and defines special `plot()` methods for seven classes of spectral objects.

A third broadly useful package is *photobiologyWavebands* providing constructors for frequently used definitions of wavelength bands and BSWFs. Other packages are more specialized, for example *photobiologyPlants* providing functions and data useful in plant photobiology and *photobiologyFilters* containing a large collection of spectral transmittance data for filters and materials.

### Use examples

In this section we provide R-code examples for printing and summarizing spectral data (Box 8.1), plotting spectral irradiance (Box 8.2), calculating and plotting effective spectral irradiance (Box 8.3) and calculating and plotting effective spectral irradiance for polycarbonate-filtered solar radiation (Box 8.4, one final example exemplifies how to calculate daily effective exposures under different filters (Box 8.5. The intention of presenting these examples is to demonstrate the simplicity of the code needed to do some frequently used calculations. In these examples we have used only

data supplied by packages in the suite, which is a suitable approach for teaching or planning of experiments. For describing real experimental conditions one should use newly measured spectral data. A final example demonstrates how to read data from a file output by a Macam spectroradiometer (Box 8.6).

## Resources

A web site dedicated to the r4photobiology suite, located at <http://www.r4photobiology.info/> provides installation instructions. Each of the packages contains one or more vignettes like User Guides and/or catalogues of the included data examples, and the individual methods, functions, operators and data objects have been documented with help pages accessible through R's built-in documentation system. A handbook on *Photobiological calculations with R* is being written by myself, Andreas Albert, Titta Kotilainen and T. Matthew Robson. A draft version will be made available on-line in early 2016, and the final version published in the Autumn of 2016.

## Acknowledgements

The development of the suite has benefited from earlier work by many different people. From the point of view of R code development and coding, the packages and books written by Hadley Wickham have been of enormous importance. It is also necessary to acknowledge the contributors to the development of R itself, and the openness of the whole R community for sharing information and tips and their willingness to help through on-line forums. From the perspective of photobiological calculations themselves, many members of the UV4Growth COST Action have contributed 'problems' with their questions, and/or data and use examples that have been very useful for the design and testing of the suite. Some people need to be mentioned specially for their contributions related to algorithms used for calculations and discussions about reliability and reproducibility: Andy McLeod, Lars-Olof Björn, Lasse Yliantilla, T. Matthew Robson and Anders Lindfros where the main contributors on these aspect. Titta Kotilainen and T. Matthew Robson have been my *guinea pigs* always willing to give quick feedback on my sometimes not so wise design decisions and showing my new uses of my own code. Other members of the Action, specially the students in the training school have contributed very useful feedback, sometimes in writing and other times with their questions and facial expressions during training events. Several companies and researchers have allowed the inclusion of their data in the

suite. They are acknowledged in the documentation. There is one additional person to thank: Nigel Paul. A chat with him at one of the first UV4Growth meetings where we exchanged ideas about the need of making these type of calculations easier, and to improve the reliability and reproducibility of UV research with plants, was what lead to my decision of spending a considerable proportion of my work (and free) time on the development of the suite.

## References

- Aphalo, P. J. (2015a). *R packages for photobiology*. URL: <http://www.r4photobiology.info/> (visited on 11/07/2015).
- (2015b). *R4Photobiology: A Suite of R Packages*. Online at ResearchGate. Oral presentation at the User! 2015 conference at Aalborg, Denmark. DOI: [10.13140/RG.2.1.4240.1763](https://doi.org/10.13140/RG.2.1.4240.1763).
  - (2015c). *Using R for photobiological calculations*. On-line at ResearchGate. Poster presented at the ESP Congress 2015 in Aveiro Portugal. DOI: [10.13140/RG.2.1.2566.3447](https://doi.org/10.13140/RG.2.1.2566.3447).
- 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).
- Dennis, B. (2013). *The R student companion*. Boca Raton, FL: CRC Press, Taylor & Francis Group. ISBN: 1439875405.

**Box 8.1:** Example code for printing and summarizing spectral data.

Print spectral data for sunlight included in package photobiology. Compared to the usual R print-out we included, in part thanks to package dplyr, additional information including time and geolocation of measurement when available.

```
print(sun.spct)

## Object: source_spct [522 x 3]
## Wavelength range 280 to 800 nm, step 0.9230769 to 1 nm
## Measured on 2010-06-22 09:51:00 UTC
## Measured at 60.21 N, 24.96 E
## Time unit 1s
##
##   w.length s.e.irrad s.q.irrad
##   <dbl>    <dbl>    <dbl>
## 1    280.0         0         0
## 2    280.9         0         0
## 3    281.8         0         0
## 4    282.8         0         0
## 5    283.7         0         0
## ..      ...      ...      ...
```

The `summary()` method for spectra also outputs additional information compared to R's `summary()` method for data frames.

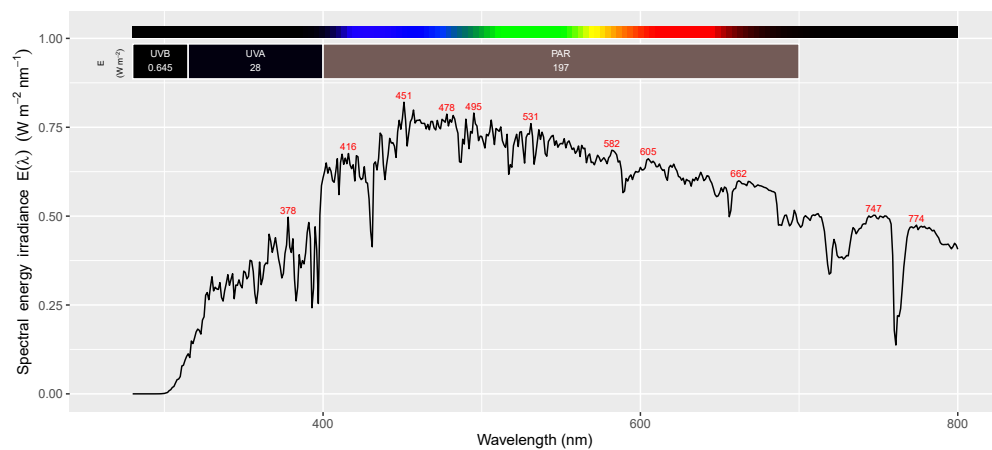
```
summary(sun.spct)

## Summary of object: source_spct [522 x 3]
## Wavelength range 280 to 800 nm, step 0.9230769 to 1 nm
## Measured on 2010-06-22 09:51:00 UTC
## Measured at 60.21 N, 24.96 E
## Time unit: 1s
##
##   w.length      s.e.irrad      s.q.irrad
## Min.   :280   Min.   :0.000   Min.   :0.00e+00
## 1st Qu.:409   1st Qu.:0.411   1st Qu.:1.98e-06
## Median :540   Median :0.580   Median :2.93e-06
## Mean   :540   Mean   :0.516   Mean   :2.41e-06
## 3rd Qu.:670   3rd Qu.:0.666   3rd Qu.:3.15e-06
## Max.   :800   Max.   :0.821   Max.   :3.37e-06
```

**Box 8.2:** Example code for plotting of spectral irradiance and spectral transmittance.

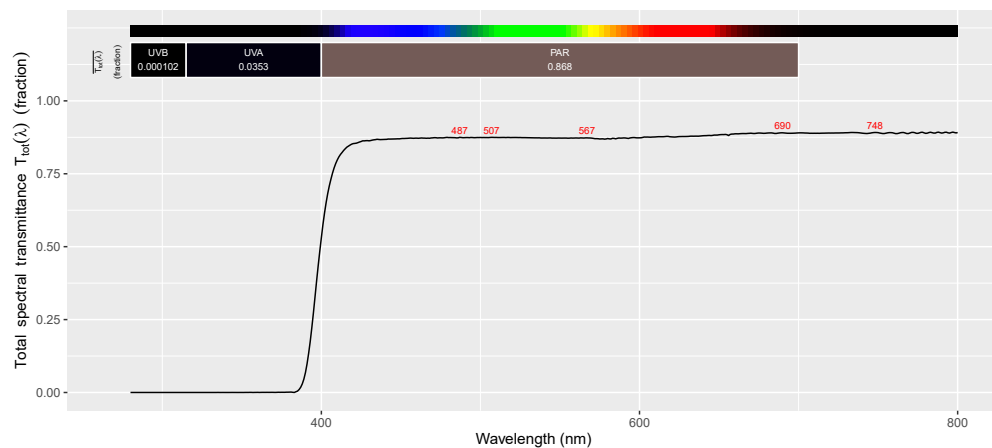
Plot of spectral irradiance of sunlight, using example data included in package photobiology.

```
plot(sun.spct)
```



Plot of spectral transmittance of 3 mm-thick polycarbonate example data included in package photobiologyFilters. We trim the wavelength range of the filter data to match the range of the solar irradiance data, so that the  $x$ -axis is identical in the three figures.

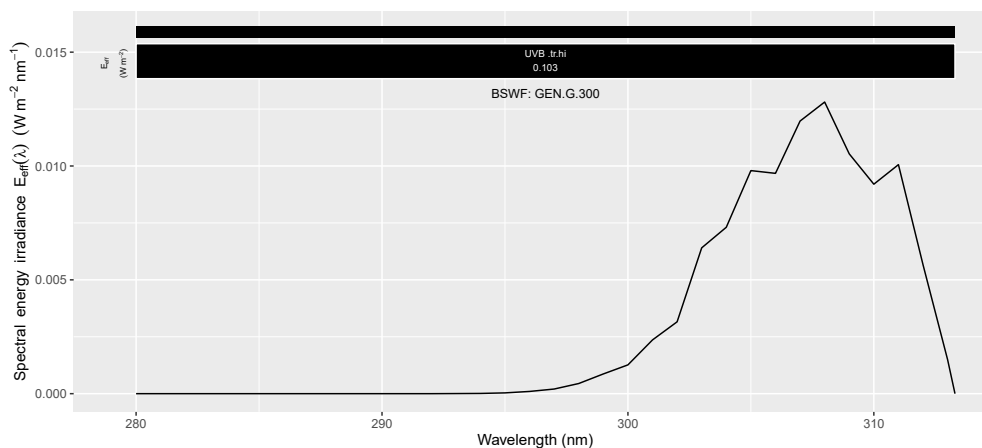
```
plot(trim_spct(foiltek.mspect$clear_PC, range=sun.spct))
```



**Box 8.3:** Example code for plotting biologically effective spectral irradiance.

Plot of spectral (energy) irradiance of sunlight weighted with the generalized plant action spectrum according to Green's formulation normalized to 300 nm, using example data included in package photobiology.

```
plot(sun.spct * GEN_G(300))
```

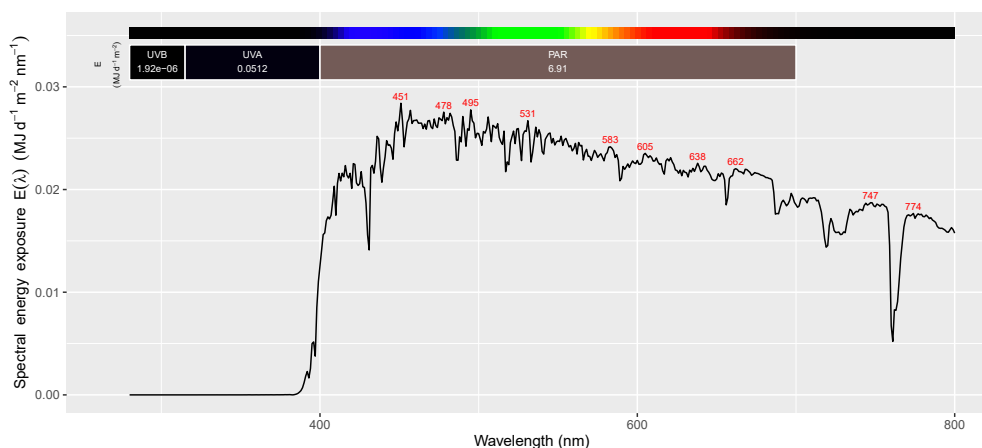


The labelling of the  $y$ -axis adjusts to the data and a label with the SWF is also added.

**Box 8.4:** Example code for estimation of spectral irradiance under a filter. Comparison of sunlight and sunlight filtered by a polycarbonate sheet.

We convolve the two spectra plotted in Box 8.2 using operator  $*$  to simulate spectral irradiance of filtered sunlight under the polycarbonate filter and plot the resulting spectrum with correctly labeled axes.

```
plot(sun.daily.spct * foiltek.mspct$Clear_PC)
```





**Box 8.5:** Example code for estimation of effective daily exposures. Comparison of sunlight and sunlight filtered by different filters.

The calculation of a effective irradiance value consists in applying a wavelength-dependent weight to spectral irradiance to obtain an effective irradiance spectrum, and then integrating this weighted spectrum over defined range of wavelengths. The equation below shows this expressed mathematically.

$$I_w = \int_{\lambda=\lambda_{\min}}^{\lambda=\lambda_{\max}} I(\lambda) \times w(\lambda) d\lambda \quad (8.1)$$

where  $I_w$  is an *effective* irradiance in *weighted*  $\text{W m}^{-2}$ ,  $\lambda$  is wavelength in nanometres,  $I(\lambda)$  is the spectral irradiance in  $\text{W m}^{-2} \text{nm}^{-1}$ ,  $w(\lambda)$  is a dimensionless **spectral weighting function** (SWF). The code chunk below applies equation 8.1 to solar spectral irradiance example data using the CIE98 spectral weighting function for erythema. Here we use `e_irrad()` to calculate energy irradiance. A function `q_irrad()` is available for photon irradiance. The predefined function `CIE()` returns a waveband object implementing the CIE98 weighting function. Other similar functions for other SWFs are also defined in package `photobiologyWavebands` and in addition users can define arbitrary definitions of both weighting functions or wavelength ranges.

```
e_irrad(sun.spct, CIE())
## CIE98.298.tr.lo
## 0.08182
## attr("time.unit")
## [1] "second"
## attr("radiation.unit")
## [1] "energy irradiance total"
```

We rarely need to calculate a single effective irradiance value, so methods in package `photobiology` allow the use of both collections of spectra and lists of waveband definitions in the same call. We can obtain a table (data frame) of effective and un-weighted daily exposures for three different action spectra, and three different bands of the spectrum, for unfiltered sunlight and sunlight filtered by three different filters in two statements. In the first statement (four top lines of code) we convolve the spectra and store the results in a 'collection of spectra'. In the second statement we calculate effective daily exposures, usually called effective doses, and also non-weighted daily energy exposures, saving the results to `results`. All results are in

```
spectra <- source_mspct(list(sun=sun.daily.spct,
  UVAB=sun.daily.spct * etola.mspct$clear_LD_PE_50um,
  UVA=sun.daily.spct * mcdermit.mspct$Autostat_CT5_125um,
  UV0=sun.daily.spct * rosco.mspct$UV_filter_EColour226))
result <- e_irrad(spectra, list(CIE=CIE(), GPAS=GEN_G(), PG=PG(),
  UVB=UVB(), UVA=UVA(), PAR=PAR()))
result
## Source: local data frame [4 x 7]
##
##   spct.idx   CIE      GPAS    PG      UVB      UVA      PAR
##   <fctr>    <dbl>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1     sun 2494.149 2.787e+03 23778.2 1.918e+04 1057512 7962841
## 2    UVAB 2173.369 2.412e+03 20979.7 1.667e+04  936883 7189580
## 3     UVA  595.829 1.848e+01 17257.1 5.043e+02  872743 7330173
## 4     UV0   9.875 2.787e-02  209.6 1.918e-01  66799 7027307
```

In the code above we have applied equation 8.1 24 times to obtain a table.



**Box 8.6:** Example code for reading a data file with spectral data output by Macam's software. We additionally print and plot the spectral irradiance data to demonstrate that also some metadata like units of expression as been also inferred or acquire from the file read.

Read, print, summarize and plot spectral data from a .DTA file from a Macam's spectroradiometer.

```
QP_UVB313.spct <- read_macam_dta("PLUS20-1.DTA")
print(QP_UVB313.spct)
```

```
## Object: source_spct [151 x 2]
## Wavelength range 250 to 400 nm, step 1 nm
## Label: File: PLUS20-1.DTA
## Measured on 1997-05-19 17:44:58 UTC
## Time unit 1s
##
##   w.length s.e.irrad
##   <dbl>    <dbl>
## 1      250         0
## 2      251         0
## 3      252         0
## 4      253         0
## 5      254         0
## ..      ...      ...
```

```
summary(QP_UVB313.spct)
```

```
## Summary of object: source_spct [151 x 2]
## Wavelength range 250 to 400 nm, step 1 nm
## Label: File: PLUS20-1.DTA
## Measured on 1997-05-19 17:44:58 UTC
## Time unit: 1s
##
##   w.length    s.e.irrad
## Min.   :250    Min.   :0.00000
## 1st Qu.:288    1st Qu.:0.00000
## Median :325    Median :0.00241
## Mean   :325    Mean   :0.01580
## 3rd Qu.:362    3rd Qu.:0.02746
## Max.   :400    Max.   :0.08619
```

```
plot(QP_UVB313.spct)
```

