Package: eemR (via r-universe)

August 25, 2024

Type Package

Title Tools for Pre-Processing Emission-Excitation-Matrix (EEM) Fluorescence Data

Version 1.0.1.9000

Description Provides various tools for preprocessing Emission-Excitation-Matrix (EEM) for Parallel Factor Analysis (PARAFAC). Different methods are also provided to calculate common metrics such as humification index and fluorescence index.

License GPL (>= 2)

BugReports https://github.com/PMassicotte/eemR/issues

URL https://github.com/PMassicotte/eemR

Depends R (>= 3.2.1)

LazyData TRUE

Imports stringr, dplyr, R.matlab, pracma, stats, rlist, viridis, purrr, assertthat

RoxygenNote 7.1.2

Suggests cdom, magrittr, testthat (>= 2.1.0), knitr, rmarkdown, ggplot2, plot3D, extrafont, tidyr, shiny, DT, MBA, here, covr

VignetteBuilder knitr

Encoding UTF-8

Repository https://pmassicotte.r-universe.dev

RemoteUrl https://github.com/pmassicotte/eemr

RemoteRef HEAD

RemoteSha 04e9f6b76624d5e2dc9d3db64e5993ede0a5ebf9

Contents

`absorbance` *CDOM absorbance data.*

Description

Simple absorbance spectra used to test package's functions.

Usage

```
data("absorbance")
```

Format

A data frame with 711 rows and 4 variables

Details

- wavelength. Wavelengths used for measurements (190-900 nm.)
- absorbance

Examples

```
data("absorbance")

plot(absorbance$wavelength, absorbance$sample1,
  type = "l",
  xlab = "Wavelengths", ylab = "Absorbance per meter"
)
lines(absorbance$wavelength, absorbance$sample2, col = "blue")
lines(absorbance$wavelength, absorbance$sample3, col = "red")
```

`eem` *eem constructor*

Description

eem constructor

Usage

```
eem(data)
```

Arguments

`data`	A list containing "file", "x", "em", "ex".

Value

An object of class `eem` containing:

- sample The sample name of the eem.
- file The filename of the eem.
- location Directory of the eem.
- x A matrix with fluorescence values.
- em Emission vector of wavelengths.
- ex Excitation vector of wavelengths.

eem_bind *Bind eem or eemlist*

Description

Function to bind EEMs that have been loaded from different folders or have been processed differently.

Usage

```
eem_bind(...)
```

Arguments

...	One or more object of class eemlist.

Value

An object of eemlist.

Examples

```
file <- system.file("extdata/cary/scans_day_1/", "sample1.csv", package = "eemR")
eem <- eem_read(file, import_function = "cary")

eem <- eem_bind(eem, eem)
```

eem_biological_index *Calculate the biological fluorescence index (BIX)*

Description

The biological fluorescence index (BIX) is calculated by dividing the fluorescence at excitation 310 nm and emission at 380 nm (ex = 310, em = 380) by that at excitation 310 nm and emission at 430 nm (ex = 310, em = 430).

Usage

```
eem_biological_index(eem, verbose = TRUE)
```

Arguments

eem	An object of class eemlist.
verbose	Logical determining if additional messages should be printed.

Value

An object of class `eemlist`.

A data frame containing the biological index (BIX) for each eem.

Interpolation

Different excitation and emission wavelengths are often used to measure EEMs. Hence, it is possible to have mismatchs between measured wavelengths and wavelengths used to calculate specific metrics. In these circumstances, EEMs are interpolated using the `interp2` function from the `parcma` library. A message warning the user will be prompted if data interpolation is performed.

References

Huguet, A., Vacher, L., Relexans, S., Saubusse, S., Froidefond, J. M., & Parlanti, E. (2009). Properties of fluorescent dissolved organic matter in the Gironde Estuary. Organic Geochemistry, 40(6), 706-719.

doi:10.1016/j.orggeochem.2009.03.002

See Also

interp2

Examples

```
file <- system.file("extdata/cary/scans_day_1/", package = "eemR")
eem <- eem_read(file, import_function = "cary")

eem_biological_index(eem)
```

`eem_coble_peaks` *Extract fluorescence peaks*

Description

Extract fluorescence peaks

Usage

```
eem_coble_peaks(eem, verbose = TRUE)
```

Arguments

`eem`	An object of class `eemlist`.
`verbose`	Logical determining if additional messages should be printed.

Details

According to Coble (1996), peaks are defined as follow:

Peak B: ex = 275 nm, em = 310 nm

Peak T: ex = 275 nm, em = 340 nm

Peak A: ex = 260 nm, em = 380:460 nm

Peak M: ex = 312 nm, em = 380:420 nm

peak C: ex = 350 nm, em = 420:480 nm

Given that peaks A, M and C are not defined at fix emission wavelength, the maximum fluorescence value in the region is extracted.

Value

An object of class `eemlist`.

A data frame containing peaks B, T, A, M and C for each eem. See details for more information.

Interpolation

Different excitation and emission wavelengths are often used to measure EEMs. Hence, it is possible to have mismatchs between measured wavelengths and wavelengths used to calculate specific metrics. In these circumstances, EEMs are interpolated using the `interp2` function from the `parcma` library. A message warning the user will be prompted if data interpolation is performed.

References

Coble, P. G. (1996). Characterization of marine and terrestrial DOM in seawater using excitation-emission matrix spectroscopy. Marine Chemistry, 51(4), 325-346.

doi:10.1016/03044203(95)000623

See Also

interp2

Examples

```
file <- system.file("extdata/cary/scans_day_1/", "sample1.csv", package = "eemR")
eem <- eem_read(file, import_function = "cary")

eem_coble_peaks(eem)
```

eem_cut *Cut emission and/or excitation wavelengths from EEMs*

Description

Cut emission and/or excitation wavelengths from EEMs

Usage

```
eem_cut(eem, ex, em, exact = TRUE, fill_with_na = FALSE)
```

Arguments

eem	An object of class eemlist.
ex	A numeric vector of excitation wavelengths to be removed.
em	A numeric vector of emission wavelengths to be removed.
exact	Logical. If TRUE, only wavelengths matching em and/or ex will be removed. If FALSE, all wavelengths in the range of em and/or ex will be removed.
fill_with_na	Logical. If TRUE, fluorescence values at specified wavelengths will be replaced with NA. If FALSE, these values will be removed.

Value

An object of class eemlist.

Examples

```
# Open the fluorescence eem
file <- system.file("extdata/cary/scans_day_1/", "sample1.csv", package = "eemR")

eem <- eem_read(file, import_function = "cary")
plot(eem)

# Cut few excitation wavelengths
eem <- eem_cut(eem, ex = c(220, 225, 230, 230))
plot(eem)

eem <- eem_read(file, import_function = "cary")
eem <- eem_cut(eem, em = 350:400, fill_with_na = TRUE)
plot(eem)
```

`eem_export_matlab` *Export EEMs to Matlab*

Description

Export EEMs to Matlab

Usage

```
eem_export_matlab(file, ...)
```

Arguments

`file`	The .mat file name where to export the structure.
`...`	One or more object of class `eemlist`.

Details

The function exports EEMs into PARAFAC-ready Matlab `.mat` file usable by the drEEM toolbox.

Value

A structure named `OriginalData` is created and contains:

nSample The number of eems.

nEx The number of excitation wavelengths.

nEm The number of emission wavelengths.

Ex A vector containing excitation wavelengths.

Em A vector containing emission wavelengths.

X A 3D matrix (nSample X nEx X nEm) containing EEMs.

`sample_name` The list of sample names (i.e. file names) of the imported EEMs.

Known bug in export

`eemR` uses `R.Matlab` to export the the fluorescence data into a matfile. However, there is currently a bug in the latter package that require the user to reshape the exported data once in Matlab. This can be done using the following command: `load('OriginalData.mat'); OriginalData.X = reshape(OriginalData.X, OriginalData.nSample, OriginalData.nEm, OriginalData.nEx)`

Examples

```
file <- system.file("extdata/cary/", package = "eemR")
eem <- eem_read(file, recursive = TRUE, import_function = "cary")

export_to <- paste(tempfile(), ".mat", sep = "")
eem_export_matlab(export_to, eem)
```

`eem_extract` *Extract EEM samples*

Description

Extract EEM samples

Usage

```
eem_extract(eem, sample, keep = FALSE, ignore_case = FALSE, verbose = TRUE)
```

Arguments

`eem`	An object of class `eemlist`.
`sample`	Either numeric of character vector. See `details` for more information.
`keep`	logical. If TRUE, the specified `sample` will be returned. If FALSE, they will be removed.
`ignore_case`	Logical, should sample name case should be ignored (TRUE) or not (FALSE). Default is FALSE.
`verbose`	Logical determining if removed/extracted eems should be printed on screen.

Details

`sample` argument can be either numeric or character vector. If it is numeric, samples at specified index will be removed.

If `sample` is character, regular expression will be used and all sample names that have a partial or complete match with the expression will be removed. See `examples` for more details.

Value

An object of class `eemlist`.

Examples

```
folder <- system.file("extdata/cary/scans_day_1", package = "eemR")
eems <- eem_read(folder, import_function = "cary")

eems

# Remove first and third samples
eem_extract(eems, c(1, 3))

# Remove everything except first and third samples
eem_extract(eems, c(1, 3), keep = TRUE)

# Remove all samples containing "3" in their names.
eem_extract(eems, "3")
```

```
# Remove all samples containing either character "s" or character "2" in their names.
eem_extract(eems, c("s", "2"))

# Remove all samples containing "blank" or "nano"
eem_extract(eems, c("blank", "nano"))
```

`eem_extract_blank` *Extract blank EEM*

Description

Extract blank EEM

Usage

```
eem_extract_blank(eem, average = TRUE)
```

Arguments

`eem`	An object of class `eemlist`.
`average`	Logical. If TRUE blank EEMs will be averaged

Value

An object of class `eemlist`.

`eem_fluorescence_index`
Calculate the fluorescence index (FI)

Description

Calculate the fluorescence index (FI)

Usage

```
eem_fluorescence_index(eem, verbose = TRUE)
```

Arguments

`eem`	An object of class `eemlist`.
`verbose`	Logical determining if additional messages should be printed.

Value

An object of class `eemlist`.

A data frame containing fluorescence index (FI) for each eem.

Interpolation

Different excitation and emission wavelengths are often used to measure EEMs. Hence, it is possible to have mismatchs between measured wavelengths and wavelengths used to calculate specific metrics. In these circumstances, EEMs are interpolated using the `interp2` function from the `parcma` library. A message warning the user will be prompted if data interpolation is performed.

References

https://doi.wiley.com/10.4319/lo.2001.46.1.0038

See Also

interp2

Examples

```
file <- system.file("extdata/cary/scans_day_1/", "sample1.csv", package = "eemR")
eem <- eem_read(file, import_function = "cary")

eem_fluorescence_index(eem)
```

`eem_humification_index`

Calculate the fluorescence humification index (HIX)

Description

The fluorescence humification index (HIX), which compares two broad aromatic dominated fluorescence maxima, is calculated at 254 nm excitation by dividing the sum of fluorescence intensities between emission 435 to 480 nm by the the sum of fluorescence intensities between 300 to 345 nm.

Usage

```
eem_humification_index(eem, scale = FALSE, verbose = TRUE)
```

Arguments

`eem`	An object of class `eemlist`.
`scale`	Logical indicating if HIX should be scaled, default is FALSE. See details for more information.
`verbose`	Logical determining if additional messages should be printed.

Value

An object of class `eemlist`.

A data frame containing the humification index (HIX) for each eem.

Interpolation

Different excitation and emission wavelengths are often used to measure EEMs. Hence, it is possible to have mismatchs between measured wavelengths and wavelengths used to calculate specific metrics. In these circumstances, EEMs are interpolated using the `interp2` function from the `parcma` library. A message warning the user will be prompted if data interpolation is performed.

References

Ohno, T. (2002). Fluorescence Inner-Filtering Correction for Determining the Humification Index of Dissolved Organic Matter. Environmental Science & Technology, 36(4), 742-746.

doi:10.1021/es0155276

See Also

interp2

Examples

```
file <- system.file("extdata/cary/scans_day_1/", package = "eemR")
eem <- eem_read(file, import_function = "cary")

eem_humification_index(eem)
```

---

eem_inner_filter_effect

Inner-filter effect correction

Description

Inner-filter effect correction

Usage

```
eem_inner_filter_effect(eem, absorbance, pathlength = 1)
```

Arguments

`eem`	An object of class `eemlist`.
`absorbance`	A data frame with: **wavelength** A numeric vector containing wavelengths. **...** One or more numeric vectors containing absorbance spectra.
`pathlength`	A numeric value indicating the pathlength (in cm) of the cuvette used for absorbance measurement. Default is 1 (1cm).

Details

The inner-filter effect correction procedure is assuming that fluorescence has been measured in 1 cm cuvette. Hence, absorbance will be converted per cm. Note that absorbance spectra should be provided (i.e. not absorption).

Value

An object of class `eemlist`.

An object of class `eem` containing:

- sample The file name of the eem.
- x A matrix with fluorescence values.
- em Emission vector of wavelengths.
- ex Excitation vector of wavelengths.

Names matching

The names of `absorbance` variables are expected to match those of the eems. If the appropriate absorbance spectrum is not found, an uncorrected eem will be returned and a warning message will be printed.

Sample dilution

Kothawala et al. 2013 have shown that a 2-fold dilution was required for sample presenting total absorbance > 1.5 in a 1 cm cuvette. Accordingly, a message will warn the user if total absorbance is greater than this threshold.

References

Parker, C. a., & Barnes, W. J. (1957). Some experiments with spectrofluorometers and filter fluorimeters. The Analyst, 82(978), 606. doi:10.1039/an9578200606

Kothawala, D. N., Murphy, K. R., Stedmon, C. A., Weyhenmeyer, G. A., & Tranvik, L. J. (2013). Inner filter correction of dissolved organic matter fluorescence. Limnology and Oceanography: Methods, 11(12), 616-630. doi:10.4319/lom.2013.11.616

Examples

```
library(eemR)
data("absorbance")

folder <- system.file("extdata/cary/scans_day_1", package = "eemR")
eems <- eem_read(folder, import_function = "cary")
eems <- eem_extract(eems, "nano") # Remove the blank sample

# Remove scattering (1st order)
eems <- eem_remove_scattering(eems, "rayleigh")

eems_corrected <- eem_inner_filter_effect(eems, absorbance = absorbance, pathlength = 1)
```

```
op <- par(mfrow = c(2, 1))
plot(eems, which = 1)
plot(eems_corrected, which = 1)
par(op)
```

`eem_names` *The names of an eem or eemlist objects*

Description

The names of an eem or eemlist objects

Usage

```
eem_names(eem)
```

Arguments

`eem`	An object of class `eemlist`.

Value

An object of class `eemlist`.

A character vector containing the names of the EEMs.

Examples

```
file <- system.file("extdata/cary/", package = "eemR")
eem <- eem_read(file, recursive = TRUE, import_function = "cary")

eem_names(eem)
```

`eem_names<-` *Set the sample names of an eem or eemlist objects*

Description

Set the sample names of an eem or eemlist objects

Usage

```
eem_names(x) <- value
```

Arguments

`x`	An object of class `eem` or `eemlist`.
`value`	A character vector with new sample names. Must be equal in length to the number of samples in the `eem` or `eemlist`.

Value

An eem or eemlist.

Examples

```
folder <- system.file("extdata/cary/scans_day_1", package = "eemR")
eems <- eem_read(folder, import_function = "cary")

eem_names(eems)
eem_names(eems) <- c("a", "b", "c", "d")
eem_names(eems)
```

---

eem_peaks *Extract fluorescence peaks*

Description

Extract fluorescence peaks

Usage

```
eem_peaks(eem, ex, em, verbose = TRUE)
```

Arguments

eem	An object of class eemlist.
ex	A numeric vector with excitation wavelengths.
em	A numeric vector with emission wavelengths.
verbose	Logical determining if additional messages should be printed.

Value

An object of class eemlist.

A data frame containing excitation and emission peak values. See details for more information.

Interpolation

Different excitation and emission wavelengths are often used to measure EEMs. Hence, it is possible to have mismatches between measured wavelengths and wavelengths used to calculate specific metrics. In these circumstances, EEMs are interpolated using the interp2 function from the parcma library. A message warning the user will be prompted if data interpolation is performed.

See Also

interp2

Examples

```
file <- system.file("extdata/cary/scans_day_1/", "sample1.csv", package = "eemR")
eem <- eem_read(file, import_function = "cary")

eem_peaks(eem, ex = c(250, 350), em = c(300, 400))
```

`eem_raman_normalisation` *Fluorescence Intensity Calibration Using the Raman Scatter Peak of Water*

Description

Normalize fluorescence intensities to the standard scale of Raman Units (R.U).

Usage

```
eem_raman_normalisation(eem, blank = NA)
```

Arguments

`eem`	An object of class `eemlist`.
`blank`	An object of class `eemlist`.

Details

The function will first try to use the provided `blank`. If the blank is omitted, the function will then try to extract the blank from the `eemlist` object. This is done by looking for sample names containing one of these complete or partial strings (ignoring case):

1. nano
2. miliq
3. milliq
4. mq
5. blank

Note that if `blank` is omitted, the function will group the `eemlist` based on file location and will assumes that there is a blank sample in each folder. In that context, the blank will be used on each sample in the same folder. If more than one blank is found they will be averaged (a message will be printed if this appends).

Consider the following example where there are two folders that could represent scans performed on two different days 'scans_day_1' and 'scans_day_2'.

scans_day_1

nano.csv

sample1.csv
sample2.csv
sample3.csv
scans_day_2
blank.csv
s1.csv
s2.csv
s3.csv

In each folder there are three samples and one blank files. In that context, ‘eem_remove_blank()‘ will use the blank ‘nano.csv‘ from ‘sample1.csv‘, ‘sample2.csv‘ and ‘sample3.csv‘. The same strategy will be used for files in folder ‘scans_day_2‘ but with blank named ‘blank.csv‘.

Note that the blanks eem are not returned by the function.

The normalization procedure consists in dividing all fluorescence intensities by the area (integral) of the Raman peak. The peak is located at excitation of 350 nm. (ex = 370) between 371 nm. and 428 nm in emission (371 <= em <= 428). Note that the data is interpolated to make sure that fluorescence at em 350 exist.

Value

An object of class `eemlist`.

An object of class `eem` containing:

- sample The file name of the eem.
- x A matrix with fluorescence values.
- em Emission vector of wavelengths.
- ex Excitation vector of wavelengths.

References

Lawaetz, A. J., & Stedmon, C. A. (2009). Fluorescence Intensity Calibration Using the Raman Scatter Peak of Water. Applied Spectroscopy, 63(8), 936-940.

doi:10.1366/000370209788964548

Murphy, K. R., Stedmon, C. a., Graeber, D., & Bro, R. (2013). Fluorescence spectroscopy and multi-way techniques. PARAFAC. Analytical Methods, 5(23), 6557.

http://xlink.rsc.org/?DOI=c3ay41160e

Examples

```
# Open the fluorescence eem
file <- system.file("extdata/cary/scans_day_1", "sample1.csv", package = "eemR")
eem <- eem_read(file, import_function = "cary")

plot(eem)

# Open the blank eem
```

```
file <- system.file("extdata/cary/scans_day_1", "nano.csv", package = "eemR")
blank <- eem_read(file, import_function = "cary")

# Do the normalisation
eem <- eem_raman_normalisation(eem, blank)

plot(eem)
```

`eem_read` *Read excitation-emission fluorescence matrix (eem)*

Description

Read excitation-emission fluorescence matrix (eem)

Usage

```
eem_read(file, recursive = FALSE, import_function)
```

Arguments

`file`	File name or folder containing fluorescence file(s).
`recursive`	logical. Should the listing recurse into directories?
`import_function`	Either a character or a user-defined function to import a single eem. If a character, it should be one of "cary", "aqualog", "shimadzu", "fluoromax4". See `browseVignettes("eemR")` to learn how to create your own import function.

Details

At the moment, Cary Eclipse, Aqualog and Shimadzu EEMs are supported.

`eemR` will automatically try to determine from which spectrofluorometer the files originate and load the data accordingly. Note that EEMs are reshaped so X[1, 1] represents the fluorescence intensity at X[min(ex), min(em)].

Value

If `file` is a single filename:

An object of class `eem` containing:

- sample The file name of the eem.
- x A matrix with fluorescence values.
- em Emission vector of wavelengths.
- ex Excitation vector of wavelengths.

If `file` is a folder, the function returns an object of class `eemlist` which is simply a list of `eem`.

Examples

```
file <- system.file("extdata/cary/scans_day_1/", package = "eemR")
eems <- eem_read(file, recursive = TRUE, import_function = "cary")
```

`eem_remove_blank` *Blank correction*

Description

This function is used to remove blank from eems which can help to reduce the effect of scatter bands.

Usage

```
eem_remove_blank(eem, blank = NA)
```

Arguments

`eem`	An object of class `eemlist`.
`blank`	An object of class `eemlist`.

Details

The function will first try to use the provided `blank`. If the blank is omitted, the function will then try to extract the blank from the `eemlist` object. This is done by looking for sample names containing one of these complete or partial strings (ignoring case):

1. nano
2. miliq
3. milliq
4. mq
5. blank

Note that if `blank` is omitted, the function will group the `eemlist` based on file location and will assumes that there is a blank sample in each folder. In that context, the blank will be used on each sample in the same folder. If more than one blank is found they will be averaged (a message will be printed if this appends).

Consider the following example where there are two folders that could represent scans performed on two different days 'scans_day_1' and 'scans_day_2'.

scans_day_1

nano.csv
sample1.csv
sample2.csv
sample3.csv

scans_day_2
blank.csv
s1.csv
s2.csv
s3.csv

In each folder there are three samples and one blank files. In that context, ‘eem_remove_blank()‘ will use the blank ‘nano.csv‘ from ‘sample1.csv‘, ‘sample2.csv‘ and ‘sample3.csv‘. The same strategy will be used for files in folder ‘scans_day_2‘ but with blank named ‘blank.csv‘.

Note that the blanks eem are not returned by the function.

Note that blank correction should be performed before Raman normalization (`eem_raman_normalisation()`). An error will occur if trying to perform blank correction after Raman normalization.

Value

An object of class `eemlist`.

References

Murphy, K. R., Stedmon, C. a., Graeber, D., & Bro, R. (2013). Fluorescence spectroscopy and multi-way techniques. PARAFAC. Analytical Methods, 5(23), 6557. http://doi.org/10.1039/c3ay41160e

http://xlink.rsc.org/?DOI=c3ay41160e

Examples

```
## Example 1

# Open the fluorescence eem
file <- system.file("extdata/cary/scans_day_1", "sample1.csv", package = "eemR")
eem <- eem_read(file, import_function = "cary")

plot(eem)

# Open the blank eem
file <- system.file("extdata/cary/scans_day_1", "nano.csv", package = "eemR")
blank <- eem_read(file, import_function = "cary")

plot(blank)

# Remove the blank
eem <- eem_remove_blank(eem, blank)

plot(eem)

## Example 2

# Open the fluorescence eem
folder <- system.file("extdata/cary/scans_day_1", package = "eemR")
eems <- eem_read(folder, import_function = "cary")
```

```
plot(eems, which = 3)

# Open the blank eem
file <- system.file("extdata/cary/scans_day_1", "nano.csv", package = "eemR")
blank <- eem_read(file, import_function = "cary")

plot(blank)

# Remove the blank
eems <- eem_remove_blank(eems, blank)

plot(eems, which = 3)

# Automatic correction
folder <- system.file("extdata/cary/", package = "eemR")

# Look at the folder structure
list.files(folder, "*.csv", recursive = TRUE)

eems <- eem_read(folder, recursive = TRUE, import_function = "cary")
res <- eem_remove_blank(eems)
```

`eem_remove_scattering` *Remove Raman and Rayleigh scattering*

Description

Remove Raman and Rayleigh scattering

Usage

```
eem_remove_scattering(eem, type, order = 1, width = 10)
```

Arguments

`eem`	An object of class `eemlist`.
`type`	A string, either "raman" or "rayleigh".
`order`	A integer number, either 1 (first order) or 2 (second order).
`width`	Slit width in nm for the cut. Default is 10 nm.

Value

An object of class `eemlist`.

References

Lakowicz, J. R. (2006). Principles of Fluorescence Spectroscopy. Boston, MA: Springer US.#'

doi:10.1007/9780387463124

Murphy, K. R., Stedmon, C. a., Graeber, D., & Bro, R. (2013). Fluorescence spectroscopy and multi-way techniques. PARAFAC. Analytical Methods, 5(23), 6557. https://doi.org/10.1039/c3ay41160e#'

https://pubs.rsc.org/en/content/articlelanding/2013/AY/c3ay41160e

Examples

```
# Open the fluorescence eem
file <- system.file("extdata/cary/scans_day_1", "sample1.csv", package = "eemR")
eem <- eem_read(file, import_function = "cary")

plot(eem)

# Remove the scattering
eem <- eem_remove_scattering(eem = eem, type = "raman", order = 1, width = 10)
eem <- eem_remove_scattering(eem = eem, type = "rayleigh", order = 1, width = 10)

plot(eem)
```

---

eem_set_wavelengths *Set Excitation and/or Emission wavelengths*

Description

This function allows to manually specify either excitation or emission vector of wavelengths in EEMs. This function is mostly used with spectrophotometers such as Shimadzu that do not include excitation wavelengths in fluorescence files.

Usage

```
eem_set_wavelengths(eem, ex, em)
```

Arguments

eem	An object of class eemlist.
ex	A numeric vector of excitation wavelengths.
em	A numeric vector of emission wavelengths.

Value

An object of class eemlist.

Examples

```
folder <- system.file("extdata/shimadzu", package = "eemR")

eem <- eem_read(folder, import_function = "shimadzu")
eem <- eem_set_wavelengths(eem, ex = seq(230, 450, by = 5))

plot(eem)
```

`plot.eemlist` *Surface plot of eem*

Description

Surface plot of eem

Usage

```
## S3 method for class 'eemlist'
plot(x, which = 1, interactive = FALSE, show_peaks = FALSE, ...)
```

Arguments

`x`	An object of class `eemlist`.
`which`	An integer representing the index of eem to be plotted.
`interactive`	If `TRUE` a Shiny app will start to visualize EEMS.
`show_peaks`	Boolean indicating if Cobble's peaks should be displayed on the surface plot. Default is FALSE.
`...`	Extra arguments for `image.plot`.

Examples

```
folder <- system.file("extdata/cary/scans_day_1/", package = "eemR")
eem <- eem_read(folder, import_function = "cary")

plot(eem, which = 3)
```

`print.eemlist` *Display summary of an eemlist object*

Description

Display summary of an eemlist object

Usage

```
## S3 method for class 'eemlist'
print(x, ...)
```

Arguments

`x`	An object of class `eemlist`.
`...`	Extra arguments.

Value

A data frame containing summarized information on EEMs.

sample Character. Sample name of the EEM,

ex_min Numerical. Minimum excitation wavelength

ex_max Numerical. Maximum excitation wavelength

em_min Numerical. Minimum emission wavelength

em_max Numerical. Maximum emission wavelength

is_blank_corrected Logical. TRUE if the sample has been blank corrected.

is_scatter_corrected Logical. TRUE if scattering bands have been removed from the sample.

is_ife_corrected Logical. TRUE if the sample has been corrected for inner-filter effect.

is_raman_normalized Logical. TRUE if the sample has been Raman normalized.

manufacturer Character. The name of the manufacturer.

Examples

```
folder <- system.file("extdata/cary", package = "eemR")
eem <- eem_read(folder, recursive = TRUE, import_function = "cary")

print(eem)
```

summary.eemlist *Display summary of an eemlist object*

Description

Display summary of an eemlist object

Usage

```
## S3 method for class 'eemlist'
summary(object, ...)
```

Arguments

object	An object of class eemlist.
...	Extra arguments.

Value

A data frame containing summarized information on EEMs.

sample Character. Sample name of the EEM,

ex_min Numerical. Minimum excitation wavelength

ex_max Numerical. Maximum excitation wavelength

em_min Numerical. Minimum emission wavelength

em_max Numerical. Maximum emission wavelength

is_blank_corrected Logical. TRUE if the sample has been blank corrected.

is_scatter_corrected Logical. TRUE if scattering bands have been removed from the sample.

is_ife_corrected Logical. TRUE if the sample has been corrected for inner-filter effect.

is_raman_normalized Logical. TRUE if the sample has been Raman normalized.

manufacturer Character. The name of the manufacturer.

Examples

```
folder <- system.file("extdata/cary", package = "eemR")
eem <- eem_read(folder, recursive = TRUE, import_function = "cary")

summary(eem)
```

Index