Package 'visa'

April 20, 2021

Type Package

Title Vegetation Imaging Spectroscopy Analyzer

Version 0.1.0

Description Provides easy-to-use tools for data analysis and visualization for hyperspectral remote sensing (also known as imaging spectroscopy), with a particular focus on vegetation hyperspectral data analysis. It consists of a set of functions, ranging from the organization of hyperspectral data in the proper data structure for spectral feature selection, calculation of vegetation index, multivariate analysis, as well as to the visualization of spectra and results of analysis in the 'ggplot2' style.

License GPL-3

LazyData true

Encoding UTF-8

URL https://github.com/kang-yu/visa

BugReports https://github.com/kang-yu/visa/issues

Depends R (>= 3.1.0)

Imports ggplot2, ggpmisc, methods, Matrix, reshape2, RColorBrewer

Suggests devtools, testthat, flux, knitr, rmarkdown, stringi

RoxygenNote 7.1.1

VignetteBuilder knitr

Collate 'visa.R' 'Spectra-class.R' 'cm.nsr.R' 'cm.sr.R' 'data-specdb.R' 'data-specdf.R' 'ggplot-method.R' 'ggplot.lmfit.R' 'ggspectra.R' 'ndvi2.R' 'spectra.R' 'sr.R' 'wavelength.R'

NeedsCompilation no

Author Kang Yu [aut, cre]

Maintainer Kang Yu <kang.yu@outlook.com>

Repository CRAN

Date/Publication 2021-04-20 07:20:02 UTC

R topics documented:

`as.spectra.data.frame` *Create a SpectraDataFrame*

Description

Constructor `as.spectra.data.frame` function creates a SpectraDataFrame object, which is equivalent to the use of as.specdf.

Usage

```
as.spectra.data.frame(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = character(0),
  data = data.frame(0),
  ...
)
```

Arguments

`spectra`	A matrix
`wavelength`	A numeric vector
`w.unit`	A character string
`data`	A data.frame
`...`	Other options for similar format of variables

Value

sdf	Returns a SpectraDataFrame.

Examples

```
sdf <- as.spectra.data.frame(matrix(1:10, 1), 1:10, "nm", data.frame(a = 1, b =2))
str(sdf)
```

---

cm.nsr *Selecting the best 2-Band combinations for Normalized Simple Ratio (NSR)*

Description

This function develops a optimization algorithm based on correlation analysis between spectral matrix 'spectra' and the vegetation variable of interest x, which determines the best spectral band combinations of the full spectrum that are most predictive for 'x'.

Usage

```
cm.nsr(S, x, w = wavelength(S), w.unit = NULL, cm.plot = FALSE)
```

Arguments

S	A matrix of spectral data, a row is a spectrum across all spectral bands.
x	A vector.
w	A vector of wavelength.
w.unit	Character string, default = NULL,
cm.plot	A logic value for whether plotting the coefficient matrix or not, default FALSE.

Details

This function runs a calculation of

$$NDVI = (\lambda_i - \lambda_j)/(\lambda_i + \lambda_j)$$

using all the possible pairs/combinations of any two bands (i,j) within the full spectrum range thoroughly. A correlation analysis is then performed between the x and all possible NDVIs, and it calculates the correlation coefficients (r) which indicates the predictive performance of each NDVI and its corresponding two-band combination. The output is the wavelength (nm) indicating the best two bands that produce the highest value of r.

Value

cm	Returns a correlation coefficients matrix.

See Also

cor

Examples

```
library(visa)
data(NSpec.DF)
x <- NSpec.DF$N # nitrogen
S <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 5)] # resampled to 5 nm steps
cm <- cm.nsr(S, x, cm.plot = TRUE)
```

---

cm.sr *Selecting the best 2-Band combinations for Simple Ratio (SR)*

Description

This function develops a optimization algorithm based on correlation analysis between spectral matrix 'spectra' and the vegetation variable of interest x, which determines the best spectral band combinations of the full spectrum that are most predictive for 'x'.

Usage

```
cm.sr(S, x, w = wavelength(S), w.unit = NULL, cm.plot = FALSE)
```

Arguments

S	A matrix of spectral data, a row is a spectrum across all spectral bands.
x	A vector.
w	A vector of wavelength.
w.unit	Character string, default = NULL,
cm.plot	A logic value for whether plotting the coefficient matrix or not, default FALSE.

Details

This function runs a calculation of

$$NDVI = \lambda_i / \lambda_j$$

using all the possible pairs/combinations of any two bands (i,j) within the full spectrum range thoroughly. A correlation analysis is then performed between the x and all possible NDVIs, and it calculates the correlation coefficients (r) which indicates the predictive performance of each NDVI and its corresponding two-band combination. The output is the wavelength (nm) indicating the best two bands that produce the highest value of r.

Value

cm	Returns a correlation coefficients matrix.

See Also

cm.nsr

Examples

```
library(visa)
data(NSpec.DF)
x <- NSpec.DF$N # nitrogen
S <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 10)] # resampled to 10 nm steps
cm <- cm.sr(S, x, cm.plot = FALSE)
```

---

ggplot *Create a new ggplot plot with a geom_line() layer from spectra data*

Description

`ggplot()` initializes a ggplot object. It can be used to declare the input spectra object for a graphic and to optionally specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.

Usage

```
## S3 method for class 'spectra'
ggplot(
  data,
  mapping = NULL,
  ...,
  wl = NULL,
  w.unit = "nm",
  environment = parent.frame()
)

## S3 method for class 'cm'
ggplot(
  data,
  mapping = NULL,
  ...,
  show.stat = TRUE,
  environment = parent.frame()
)
```

Arguments

`data`	Default spectra database to use for plot. If not a spectra database, the methods used will be those defined in package `ggplot2`. See `ggplot`. If not specified, must be supplied in each layer added to the plot.

mapping	Default list of aesthetic mappings to use for plot. If not specified, in the case of spectra objects, a default mapping will be used.
...	Other arguments passed on to methods. Not currently used.
wl	numeric The wavelength vector.
w.unit	character The wavelength unit of the spectra.
environment	If an variable defined in the aesthetic mapping is not found in the data, ggplot will look for it in this environment. It defaults to using the environment in which ggplot() is called.
show.stat	A logic value. whether show the best R^2 and bands.

Details

ggplot() is typically used to construct a plot incrementally, using the + operator to add layers to the existing ggplot object. This is advantageous in that the code is explicit about which layers are added and the order in which they are added. For complex graphics with multiple layers, initialization with ggplot is recommended.

Value

cm_plot	Returns a ggplot object of correlation-matrix.

Note

Current implementation does not merge default mapping with user supplied mapping. If user supplies a mapping, it is used as is. To add to the default mapping, aes() can be used by itself to compose the ggplot.

See Also

?ggpmisc::ggplot()

Examples

```
library(visa)
library(ggplot2)
ggplot.spectra(NSpec.DF)

library(visa)
data(NSpec.DF)
x <- NSpec.DF$N # nitrogen
S <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 5)] # resampled to 10 nm steps
cm <- cm.sr(S, x, cm.plot = FALSE)
ggplot.cm(cm)
```

`ggplot-method` *Plot functions*

Description

This functions plots model fit using ggplot.

Usage

```
## S3 method for class 'lmfit'
ggplot(x, y, ..., environment = parent.frame())
```

Arguments

`x, y`	Two vectors
`...`	Other arguments passed on to methods. Not currently used.
`environment`	If an variable defined in the aesthetic mapping is not found in the data, ggplot will look for it in this environment. It defaults to using the environment in which `ggplot()` is called.

Details

Visualization of linear fit (y = ax + b), using scatter plots and with regression line, as well as added details of regression equation and R^2.

Value

`p`	Returns a ggplot object.

Examples

```
library(visa)
x <- 1:10
y <- 2:11+0.5
ggplot.lmfit(x, y)
```

ndvi2 *Calculate and plot a 2-band NDVI.*

Description

This function calculates a 2-band NDVI using the nsr function.

Usage

```
ndvi2(s, b1, b2)
```

Arguments

s	Spectral data in the format of visa's Spectra object, spectra.data.frame or spectra.matrix.
b1	A integer number which defines the wavelength of the 1st spectral band.
b2	A integer number which defines the wavelength of the 2nd spectral band.

Details

Calculate a NDVI with two specific bands of choice. The new NDVI follows the the standard formula

$$NDVI = (\lambda_i + \lambda_j)/(\lambda_i - \lambda_j)$$

. Bands i and j correspond to the b1 and b2 input arguments, respectively. Wavelength indexes are determined based on the first argument 's'.

Value

ndvi	The returned values are the new NDVI.

Examples

```
library(visa)
s <- NSpec.DF$spectra
ndvi2(s, 780, 680)
```

NSpec.DB *Example data in the Spectra/SpectraDatabase format.*

Description

A S4 data structure containing the plant spectra and nitorgen (N) content. Spectra is organized as a matrix and is stored as a slot, named 'spectra'. The corresponding N content is stored in the slot 'data', which is a data.frame to be used for storing vegetation traits, such as here the plant N content.

Usage

```
NSpec.DB
```

Format

A Spectra object with 19 rows and 4 slots (spectra, wavelength, w.unit, data).

spectra A matrix of plant spectral data

wavelength A vector of wavelength for the 'spectra' data

w.unit A character string of wavelength unit (default "nm")

data A data.frame of vegetation traits, here plant nitrogen content ...currently not used

Examples

```
library(visa)
data(NSpec.DB)
str(NSpec.DB)
```

NSpec.DF *Example data in the SpectraDataFrame format*

Description

A dataset containing the plant Nitrogen content and spectra. The Spectra matrix is stored as a variable (in a column) of a data.frame.

Usage

```
NSpec.DF
```

Format

A data frame with 19 rows and 2 variables:

N Plant nitrogen content

spectra A variable of Matrix of plant spectra ...

See Also

data.frame and NSpec.DB

Examples

```
library(visa)
data(NSpec.DF)
str(NSpec.DF)
```

---

spectra *Access the spectra data of 'SpectraDatabase'.*

Description

Functions to access slot data of the Class Spectra.

Usage

```
spectra(object, ...)

## S4 method for signature 'Spectra'
spectra(object, ...)

## S4 method for signature 'data.frame'
spectra(object, ...)

## S4 method for signature 'matrix'
spectra(object, ...)
```

Arguments

object	A Spectra object, spectra.data.frame, or spectra.matrix.
...	Other options.

Details

Construct generic functions for the Spectra object, spectra.data.frame, and spectra.matrix.

Examples

```
# For the S4 class 'Spectra'
library(visa)
data(NSpec.DB)
spectra_matrix <- spectra(NSpec.DB)
# For the spectra data.frame
data(NSpec.DF)
spectra_matrix <- spectra(NSpec.DF)
```

`Spectra-class` *Create a Spectra or SpectraDatabase*

Description

Constructor `as.spectra` creates a Spectra object.

Constructor `as.spectra.database` creates a SpectraDatabase object.

Usage

```
as.spectra(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = "nm",
  data = data.frame(),
  ...
)

as.spectra.database(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = "nm",
  data = data.frame(),
  ...
)
```

Arguments

`spectra`	A matrix
`wavelength`	A numeric vector
`w.unit`	A character string
`data`	A data.frame
`...`	Other parameters

Slots

`spectra` A matrix

`wavelength` A numeric vector

`w.unit` A character string

`data` A data.frame

Examples

```
s <- as.spectra(matrix(1:100, 4), 1:25, "nm", data.frame(x = letters[1:4]))
str(s)

s <- as.spectra.database(matrix(1:100, 4), 1:25, "nm", data.frame(x = letters[1:4]))
str(s)
```

SpectraDatabase-class *Class 'SpectraDatabase'*

Description

SpectraDatabase is an extended 'Spectra' class, with associated vegetation data ('data') in a data.frame.

Slots

`spectra` A matrix

`wavelength` A numeric vector

`w.unit` A character string

`data` A data.frame of vegetation data corresponding to the spectra

---

SpectraDataFrame-class *Class 'SpectraDataFrame'*

Description

SpectraDataFrame is an extended 'Spectra' class, with associated vegetation data ('data') in a data.frame.

Slots

`spectra` A matrix

`wavelength` A numeric vector

`w.unit` A character string

`data` A data.frame of vegetation data corresponding to the spectra

SpectraMatrix-class *Class 'SpectraMatrix'*

Description

SpectraMatrix is a extended 'Spectra' class.

Constructor `as.spectra.matrix` creates a SpectraMatrix object.

Usage

```
as.spectra.matrix(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = character(0)
)
```

Arguments

`spectra`	A matrix
`wavelength`	A numeric vector
`w.unit`	A character string

Value

`sdf`	Returns a SpectraDataFrame.

Examples

```
smatrix <- as.spectra.matrix(matrix(1:10, 1), 1:10, "nm")
str(smatrix)
```

sr *Calculate Simple Ratio (SR).*

Description

Simple Ratio is the ratio of the spectra (mostly reflectance) between two bands in the format of

$$SR = \lambda_i / \lambda_j$$

It is a normalization of SR by doing NSR = (1-SR)/(1+SR), with the same two spectral bands.

Usage

```
sr(s, b1, b2)

nsr(s, b1, b2)

lm.sr(s, b1, b2, y)

lm.nsr(s, b1, b2, y)
```

Arguments

s	Spectral data in the format of visa's Spectra object, spectra.data.frame or spectra.matrix.
b1	A integer number which defines the wavelength of the 1st spectral band.
b2	A integer number which defines the wavelength of the 2nd spectral band.
y	A numeric variable to correlate with SR

Details

Simple ratio and NDVI looking indices are the two groups of mostly used spectral indices in remote sensing.

As it exactly reads in its name, it is a normalization of the SR and ranges in (0,1).

Value

sr	Returns a simple ratio index.
nsr	Returns a NSR index.
p	Returns a ggplot object.
p	Returns a ggplot object.

Examples

```
library(visa)
s <- NSpec.DF$spectra
sr1 <- sr(s, 480, 550)

s <- NSpec.DF$spectra
nsr1 <- nsr(s, 480, 550)

s <- NSpec.DF
y <- NSpec.DF$N
lm.sr(s,600,500,y)

s <- NSpec.DF
y <- NSpec.DF$N
lm.nsr(s,600,500,y)
```

wavelength *Access the wavelength of Spectra*

Description

Construct generic functions for the Spectra object, spectra.data.frame, and spectra.matrix.

Usage

```
wavelength(object, ...)

## S4 method for signature 'Spectra'
wavelength(object, ...)

## S4 method for signature 'data.frame'
wavelength(object, ...)

## S4 method for signature 'matrix'
wavelength(object, ...)
```

Arguments

object	A object of Spectra
...	Other options (... T/F with unit)

Details

A call to new returns a newly allocated object from the class identified by the first argument. This call in turn calls the method for the generic function ‘initialize‘. Construct a Spectra class by using the

Examples

```
library(visa)
# For S4 class Spectra
wavelength(NSpec.DB)
# For spectra data.frame format
wavelength(NSpec.DF)
```

Index