Package 'spectratrait'

February 5, 2021

Title A simple add-on package to aid in the fitting of leaf-level spectra-trait PLSR models

Version 0.9.9

Maintainer Shawn P. Serbin <sserbin@bnl.gov>

Description Provides functions to conduct standardized leaf-level spectra-trait PLSR model fitting including uncertainty analysis that follow DOI: https://doi.org/10.1111/nph.16123

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports httr (>= 1.4.2),
readr (>= 1.3.1),
pls (>= 2.7-2),
dplyr (>= 1.0.1),
reshape2 (>= 1.4.4),
here (>= 0.1),
plotrix (>= 3.7-8),
ggplot2 (>= 3.3.2),
gridExtra (>= 2.3),
scales (>= 1.1.1)

Suggests devtools (>= 2.3.1),
remotes (>= 2.2.0),
RCurl (>= 1.98-1.2)

Depends R (>= 2.10)

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

create_data_split 2
ely_plsr_data 3
f.coef.valid 3
f.plot.coef 4
f.plot.spec 4
find_optimal_components 5

get_ecosis_data 6
pls_permutation 7
source_GitHubData 7
testForPackage 8
VIP 8
VIPjh 8

Index **9**

`create_data_split` *Create a calibration (training) / validation data split for PLSR model fitting and testing*

Description

Create a calibration (training) / validation data split for PLSR model fitting and testing

Usage

```
create_data_split(
  dataset = NULL,
  approach = NULL,
  split_seed = 123456789,
  prop = 0.8,
  group_variables = NULL
)
```

Arguments

`dataset`	input full PLSR dataset to split into cal/val datasets
`approach`	approach to splitting the dataset. Options: base or dplyr
`split_seed`	random seed to use for splitting data
`prop`	the proportion of data to preserve for calibration (e.g. 0.8) and validation (0.2). This sets the calibration proportion
`group_variables`	Use factor variables to conduct a stratfied sampling for cal/val

Value

output_list A list containing the calibration dataset (cal_data) and validation dataset (val_data)

Author(s)

Julien Lamour, Jeremiah Anderson, Shawn P. Serbin

ely_plsr_data — *Ely et al (2019) example leaf-level PLSR dataset. DOI: https://doi.org/10.1093/jxb/erz061*

Description

Ely et al (2019) example leaf-level PLSR dataset. DOI: https://doi.org/10.1093/jxb/erz061

Usage

```
ely_plsr_data
```

Format

An object of class `data.frame` with 178 rows and 1908 columns.

f.coef.valid — *f.coef.valid*

Description

f.coef.valid

Usage

```
f.coef.valid(plsr.out, data_plsr, ncomp, inVar)
```

Arguments

`plsr.out`	plsr model obtained with jaccknife = TRUE
`data_plsr`	data used for the plsr model with Spectra the matrix of spectra
`ncomp`	number of selection components
`inVar`	Name of the PLSR model response variable

Value

B returns the intercept and the coefficients of the jackknife or bootstrap validation

Author(s)

Julien Lamour

f.plot.coef *f.plot.coef*

Description

f.plot.coef

Usage

```
f.plot.coef(
  Z,
  wv,
  xlim = NULL,
  position = "topright",
  type = "Coefficient",
  plot_label = NULL
)
```

Arguments

Z	Coefficient matrix with each row corresponding to the coefficients and wavelength in columns
wv	vector of wavelengths
xlim	vector to change the default xlim of the plots (ex xlim = c(500, 2400))
position	Position of the legend (see base function legend for help)
type	Name of the y axis and of the legend
plot_label	optional plot label to include with the figure

Author(s)

Julien Lamour

f.plot.spec *f.plot.spec*

Description

f.plot.spec

Usage

```
f.plot.spec(
  Z,
  wv,
  xlim = NULL,
  position = "topright",
  type = "Reflectance",
  plot_label = NULL
)
```

Arguments

Z	Spectra matrix with each row corresponding to a spectra and wavelength in columns
wv	vector of wavelengths corresponding to the column of the spectra matrix Z
xlim	vector to change the default xlim of the plots (ex xlim = c(500, 2400))
position	Position of the legend (see base function legend for help)
type	Name of the y axis and of the legend. E.g. Reflectance, Transmittance
plot_label	optional plot label to include with the figure

Author(s)

Julien Lamour, Shawn P. Serbin

---

find_optimal_components

Apply different methods to determing the optimal number of PLSR model components

---

Description

Apply different methods to determing the optimal number of PLSR model components

Usage

```
find_optimal_components(
  dataset = NULL,
  method = "pls",
  maxComps = 20,
  iterations = 20,
  seg = 100,
  prop = 0.7,
  random_seed = 123456789
)
```

Arguments

dataset	input full PLSR dataset. Usually just the calibration dataset
method	Which approach to use to find optimal components. Options: pls, firstPlateau, firstMin
maxComps	maximum number of components to consider
iterations	how many different permutations to run
seg	For the built-in pls method, how many different data segments to select from the input dataset
prop	proportion of data to preserve for each permutation
random_seed	random seed to use for splitting data

Value

nComps the optimal number of PLSR components

Author(s)

Julien Lamour, Jeremiah Anderson, Shawn P. Serbin

`get_ecosis_data` *Function to pull data from EcoSIS using the EcoSIS API*

Description

Function to pull data from EcoSIS using the EcoSIS API

Usage

```
get_ecosis_data(ecosis_id = NULL)
```

Arguments

`ecosis_id`	the alphanumeric EcoSIS API dataset ID

Value

EcoSIS spectral dataset object

Author(s)

Shawn P. Serbin, Alexey Shiklomanov

Examples

```
## Not run:
ecosis_id <- "960dbb0c-144e-4563-8117-9e23d14f4aa9"
dat_raw <- get_ecosis_data(ecosis_id = ecosis_id)
head(dat_raw)
names(dat_raw)[1:40]

## End(Not run)
```

`pls_permutation` *Run a PLSR model permutation analysis. Can be used to determine the optimal number of components or conduct a boostrap uncertainty analysis*

Description

See Serbin et al. (2019). DOI: https://doi.org/10.1111/nph.16123

Usage

```
pls_permutation(
  dataset = NULL,
  maxComps = 20,
  iterations = 20,
  seg = 100,
  prop = 0.7
)
```

Arguments

`dataset`	input full PLSR dataset. Usually just the calibration dataset
`maxComps`	maximum number of components to use for each PLSR fit
`iterations`	how many different permutations to run
`seg`	currently unused - should be removed from this function call
`prop`	proportion of data to preserve for each permutation

Author(s)

Julien Lamour, Shawn P. Serbin

`source_GitHubData` *Function to source text data from GitHub*

Description

Function to source text data from GitHub

Usage

```
source_GitHubData(url, sep = ",", header = TRUE)
```

Arguments

`url`	http/https URL to the github dataset
`sep`	dataset file delimiter
`header`	TRUE/FALSE does the file have a column header?

Author(s)

gist.github.com/christophergandrud/4466237

`testForPackage` *Function to check for installed package not presently used*

Description

Function to check for installed package not presently used

Usage

```
testForPackage(pkg)
```

`VIP` *VIP returns all VIP values for all variables and all number of components, as a ncomp x nvars matrix.*

Description

VIP returns all VIP values for all variables and all number of components, as a ncomp x nvars matrix.

Usage

```
VIP(object)
```

`VIPjh` *VIPjh returns the VIP of variable j with h components*

Description

VIPjh returns the VIP of variable j with h components

Usage

```
VIPjh(object, j, h)
```

Index

∗ **datasets**
ely_plsr_data, 3

create_data_split, 2

ely_plsr_data, 3

f.coef.valid, 3
f.plot.coef, 4
f.plot.spec, 4
find_optimal_components, 5

get_ecosis_data, 6

pls_permutation, 7

source_GitHubData, 7

testForPackage, 8

VIP, 8
VIPjh, 8