

Package ‘fingerPro’

February 20, 2018

Type Package

Title Sediment Source Fingerprinting

Version 1.0

Date 2018-01-31

Description Quantifies the provenance of the sediments in a catchment or study area. Based on a comprehensive characterization of the sediment sources and the end sediment mixtures a mixing model algorithm is applied to the sediment mixtures in order to estimate the relative contribution of each potential source. The package includes several statistical methods such as Kruskal-Wallis test, discriminant function analysis ('DFA'), principal component plot ('PCA') to select the optimal subset of tracer properties. The variability within each sediment source is also considered to estimate the statistical distribution of the sources contribution.

License GPL (>= 2)

URL <https://github.com/eead-csic-eesa>

LazyData true

Imports Rcpp (>= 0.11.3), klaR (>= 0.6-12), ggplot2 (>= 2.2.1), GGally (>= 1.3.2), MASS (>= 7.3.45), Rcmdr (>= 2.4-1), plyr (>= 1.8.4), reshape (>= 0.8.7), rgl (>= 0.99.9), grid (>= 3.1.1), gridExtra (>= 2.3), scales (>= 0.5.0), car (>= 2.1.6)

LinkingTo Rcpp, RcppGSL

RoxygenNote 6.0.1

NeedsCompilation yes

Author Ivan Lizaga [aut, cre],
Borja Latorre [aut],
Leticia Gaspar [aut],
Ana Navas [aut],
Vince Q Vu [ctb]

Maintainer Ivan Lizaga <lizaga.ivan10@gmail.com>

Repository CRAN

Date/Publication 2018-02-20 11:30:30 UTC

R topics documented:

| | |
|-----------------------------------|-----------|
| fingerPro-package | 2 |
| boxPlot | 3 |
| catchment | 4 |
| correlationPlot | 4 |
| DFATest | 5 |
| ggbiplot | 5 |
| inputSample | 6 |
| inputSource | 7 |
| KWTest | 7 |
| LDAPlot | 8 |
| PCAPlot | 8 |
| plotResults | 9 |
| rangeTest | 9 |
| unmix | 10 |
| unmixing_corrected_gof1 | 10 |
| writeResults | 11 |
| Index | 12 |

| | |
|-------------------|---------------------------------------|
| fingerPro-package | <i>Sediment Source Fingerprinting</i> |
|-------------------|---------------------------------------|

Description

Soil erosion is one of the biggest challenges for food production and reservoirs siltation around the world. Information on sediment, nutrients and pollutant transport is required for effective control strategies. Source estimates are difficult to obtain using traditional monitoring techniques, but sediment source fingerprinting, has been proved to be a valuable tool. Sediment source fingerprinting offers the potential to assess sediment provenance as a basis to develop management plans and prevent erosion. The procedure focuses on developing methods that enable the apportionment of sediment sources to be identified from a composite sample of sediment mixture material. We developed an R-package as a tool to quantify the provenance of the sediments in a catchment. A mixing model algorithm is applied to the sediment mixture samples in order to estimate the relative contribution of each potential source. The package consists of a set of functions used to: i) characterise and pre-process the data, select the optimum subset of tracers; ii) unmix sediment samples and quantify the apportionment of each source; iii) assess the effect of the source variability; and iv) visualize and export the results.

Author(s)

Ivan Lizaga, Borja Latorre, Leticia Gaspar, Ana Navas

Maintainer: Ivan Lizaga <ilizaga@eead.csic.es // lizaga.ivan10@gmail.com>

See Also

<https://github.com/eead-csic-eesa>

Examples

```

#Load the dataset called "catchment"
data <- catchment
LDAPlot(data, P3D=FALSE)
#
data <- rangeTest(data)
data <- KWTest(data)
#Too many tracer properties were removed
LDAPlot(data, P3D=FALSE)
#2D and 3D LDAPlots suggest that two of the sources have to be combined

#reload the original dataset "catchment"
data <- catchment
# Combine sources PI1 and PI based on the previous LDAPlot
data$Land_Use[data$Land_Use == 'PI1'] <- 'PI'
data <- rangeTest(data)
data <- KWTest(data)
LDAPlot(data, P3D=FALSE)
#Now the optimum tracer properties selected discriminate well, so proceed with the unmix function
result <- unmix(data, samples = 100, iter =100)
#Display the results
plotResults(result, y_high = 4.5, n = 1)
writeResults(result)

```

boxPlot

Box and whiskers plot

Description

The boxplot compactly shows the distribution of a continuous variable. It displays five summary statistics (the median, two hinges and two whiskers), and all "outlying" points individually.

Usage

```
boxPlot(data, columns = 1:ncol(data) - 2, ncol = 3)
```

Arguments

| | |
|---------|---|
| data | Data frame containing source and mixtures data |
| columns | Numeric vector containing the index of the columns in the chart (the first column refers to the first variable) |
| ncol | Number of charts per row |

Examples

```

#Load the dataset called "catchment"
data <- catchment
boxPlot(data,columns = 1:4, ncol = 2)
#The function display the boxplot of the selected tracer properties

```

| | |
|-----------|--|
| catchment | <i>Land use and fingerprinting properties in a Mediterranean catchment</i> |
|-----------|--|

Description

A dataset containing the different tracer properties of the different land uses in a Mediterranean catchment and one mixture sample located at the output of the catchment. The variables are as follows:

Usage

catchment

Format

A data frame with 22 rows and 25 variables:

id reference number id of each sample analysed

Land_Use grouping variable, in this study refers to the different land uses in the catchment

Cs.137, K.40, Bi.214, Ra.226, Th.232, U.238, Ba, Nb, Zr, Sr, Rb, Pb, Zn, Fe, Mn, Cr, V, Ti, Ca, K, Al, Si, Mg
value of the tracer property for each sample

| | |
|-----------------|---------------------------------|
| correlationPlot | <i>Correlation matrix chart</i> |
|-----------------|---------------------------------|

Description

The function displays a correlation matrix of each of the properties divided by the different sources to help the user in the decision.

Usage

```
correlationPlot(data, columns = c(1:ncol(data) - 1), mixtures = F)
```

Arguments

| | |
|----------|--|
| data | Data frame containing source and mixtures data |
| columns | Numeric vector containing the index of the columns in the chart (the first column refers to the grouping variable) |
| mixtures | Boolean to include or exclude the mixture samples in the chart |

Examples

```
#Load the dataset called "catchment"
data <- catchment
correlationPlot(data, columns = 1:4)
#The function display a correlation plot with the selected tracer properties
```

| | |
|---------|--|
| DFATest | <i>Discriminant function analysis test</i> |
|---------|--|

Description

Performs a stepwise forward variable selection using the Wilk's Lambda criterion.

Usage

```
DFATest(data, niveau = 0.1)
```

Arguments

| | |
|--------|---|
| data | Data frame containing source and mixtures |
| niveau | level for the approximate F-test decision |

Value

Data frame only containing the variables that pass the DFA test

Examples

```
#Load the dataset called "catchment"
data <- catchment
#Some of the tracer properties are collinear
data <- DFATest(data)
#The test discard those tracer properties that were unable to discriminate between selected sources
```

| | |
|----------|--|
| ggbiplot | <i>Biplot for Principal Components using ggplot2</i> |
|----------|--|

Description

Biplot for Principal Components using ggplot2

Usage

```
ggbiplot(pcoobj, choices = 1:2, scale = 1, pc.biplot = TRUE,
  obs.scale = 1 - scale, var.scale = scale, groups = NULL,
  ellipse = FALSE, ellipse.prob = 0.68, labels = NULL, labels.size = 3,
  alpha = 1, var.axes = TRUE, circle = FALSE, circle.prob = 0.69,
  varname.size = 3, varname.adjust = 1.5, varname.abbrev = FALSE, ...)
```

Arguments

| | |
|----------------|--|
| pcobj | an object returned by pcomp() or princomp() |
| choices | which PCs to plot |
| scale | covariance biplot (scale = 1), form biplot (scale = 0). When scale = 1, the inner product between the variables approximates the covariance and the distance between the points approximates the Mahalanobis distance. |
| pc.biplot | for compatibility with biplot.princomp() |
| obs.scale | scale factor to apply to observations |
| var.scale | scale factor to apply to variables |
| groups | optional factor variable indicating the groups that the observations belong to. If provided the points will be colored according to groups |
| ellipse | draw a normal data ellipse for each group? |
| ellipse.prob | size of the ellipse in Normal probability |
| circle.prob | size of the circle in Normal probability |
| ... | etc |
| labels | optional vector of labels for the observations |
| labels.size | size of the text used for the labels |
| alpha | alpha transparency value for the points (0 = transparent, 1 = opaque) |
| var.axes | draw arrows for the variables? |
| circle | draw a correlation circle? (only applies when pcomp was called with scale = TRUE and when var.scale = 1) |
| varname.size | size of the text for variable names |
| varname.adjust | adjustment factor the placement of the variable names, >= 1 means farther from the arrow |
| varname.abbrev | whether or not to abbreviate the variable names |

Value

a ggplot2 plot

| | |
|-------------|--------------------------------|
| inputSample | <i>Input sediment mixtures</i> |
|-------------|--------------------------------|

Description

The function select and extract the sediment mixtures of the dataset.

Usage

```
inputSample(data)
```

Arguments

| | |
|------|--|
| data | Data frame containing source and mixtures data |
|------|--|

| | |
|-------------|-------------------------------|
| inputSource | <i>Input sediment sources</i> |
|-------------|-------------------------------|

Description

The function select and extract the source samples of the dataset.

Usage

```
inputSource(data)
```

Arguments

| | |
|------|--|
| data | Data frame containing source and mixtures data |
|------|--|

| | |
|--------|-------------------------------------|
| KWTest | <i>Kruskal-Wallis rank sum test</i> |
|--------|-------------------------------------|

Description

This function excludes from the original data frame the properties which do not show significant differences between sources.

Usage

```
KWTest(data, pvalue = 0.05)
```

Arguments

| | |
|--------|---|
| data | Data frame containing source and mixtures |
| pvalue | p-value threshold |

Value

Data frame only containing the variables that pass the Kruskal-Wallis test

Examples

```
#Load the dataset called "catchment"  
data <- catchment  
#Some of the tracer properties are collinear  
data <- KWTest(data)  
#The test discard those tracer properties that were unable to discriminate between selected sources
```

LDAPlot

Linear discriminat analysis chart

Description

The function performs a linear discriminant analysis and displays the data in the relevant dimensions.

Usage

```
LDAPlot(data, P3D = FALSE)
```

Arguments

| | |
|------|--|
| data | Data frame containing source and mixtures data |
| P3D | Boolean to switch between 2 to 3 dimensional chart |

PCAPlot

Principal component analysis chart

Description

The function performs a principal components analysis on the given data matrix and displays a biplot using `vqv.ggbiplot` package of the results for each different source to help the user in the decision.

Usage

```
PCAPlot(data, components = c(1:2))
```

Arguments

| | |
|------------|--|
| data | Data frame containing source and mixtures data |
| components | Numeric vector containing the index of the two principal components in the chart |

| | |
|-------------|---|
| plotResults | <i>Displays the results in the screen</i> |
|-------------|---|

Description

The function performs a density chart of the relative contribution of the potential sediment sources for each sediment mixture in the dataset.

Usage

```
plotResults(data, y_high = 10, n = 1)
```

Arguments

| | |
|--------|--|
| data | Data frame containing the relative contribution of the potential sediment sources for each sediment mixture in the dataset |
| y_high | Number of the vertical height of the y-axis |
| n | Number of charts per row |

| | |
|-----------|-------------------|
| rangeTest | <i>Range test</i> |
|-----------|-------------------|

Description

Function that excludes the properties of the sediment mixture/s outside the minimum and maximum values in the sediment sources.

Usage

```
rangeTest(data)
```

Arguments

| | |
|------|---|
| data | Data frame containing source and mixtures |
|------|---|

Value

Data frame containing sediment sources and mixtures

| | |
|-------|--------------------------------|
| unmix | <i>Unmix sediment mixtures</i> |
|-------|--------------------------------|

Description

Asses the relative contribution of the potential sediment sources for each sediment mixture in the dataset.

Usage

```
unmix(data, samples = 100L, iter = 100L, seed = 123456L)
```

Arguments

| | |
|---------|--|
| data | Data frame containing sediment source and mixtures |
| samples | Number of samples in each hypercube dimension |
| iter | Iterations in the source variability analysis |
| seed | Seed for the random number generator |

Value

Data frame containing the relative contribution of the sediment sources for each sediment mixture and iterations

| | |
|-------------------------|---------------------|
| unmixing_corrected_gof1 | <i>Mixing model</i> |
|-------------------------|---------------------|

Description

Mixing model

Usage

```
unmixing_corrected_gof1(sources, samples, trials = 100L, iter = 100L,
  seed = 69512L)
```

Arguments

| | |
|---------|---|
| sources | Data frame containing sediment sources data |
| samples | Data frame containing sediment mixtures data |
| trials | Number of samples in each hypercube dimension |
| iter | Iterations in the source variability analysis |
| seed | Seed for the random number generator |

Value

Data frame containing the relative contribution of the sediment sources for each sediment mixture and iterations

writeResults *Save the results*

Description

The function saves the results in the workspace for all the sediment mixture samples and for each sediment mixture sample separately

Usage

`writeResults(data)`

Arguments

`data` Data frame containing the relative contribution of the potential sediment sources for each sediment mixture in the dataset

Index

*Topic **Sediment source
fingerprinting package**

fingerPro-package, 2

*Topic **datasets**

catchment, 4

boxPlot, 3

catchment, 4

correlationPlot, 4

DFATest, 5

fingerPro (fingerPro-package), 2

fingerPro-package, 2

ggbiplot, 5

inputSample, 6

inputSource, 7

KWTest, 7

LDAPlot, 8

PCAPlot, 8

plotResults, 9

rangeTest, 9

unmix, 10

unmixing_corrected_gof1, 10

writeResults, 11