Package 'rpcss'

August 16, 2024

Title Core Collection Constitution by Principal Component Scoring Strategy

Version 0.0.0.9000

Description Generate a Core Collection with Principal Component Scoring Strategy (PCSS) (Noirot et al. 1996 <doi:10.2307/2527837> ; Noirot et al. 2003 <https://horizon.documentation.ird.fr/exl-doc/pleins_textes/divers21-03/010031886.pdf>) using qualitative and/or quantitative trait data.

License GPL-2 | GPL-3

Encoding UTF-8

Depends R (>= 3.5.0)

VignetteBuilder knitr

RoxygenNote 7.3.2

BuildManual TRUE

Imports FactoMineR, ggplot2, ggrepel, gslnls, mathjaxr, Rdpack

Suggests EvaluateCore, factoextra, knitr, rmarkdown, pander

RdMacros mathjaxr, Rdpack

URL https://github.com/aravind-j/rpcss

BugReports https://github.com/aravind-j/rpcss/issues

Contents

`biplot.pcss.core` *Generate Biplots from* `pcss.core` *Output*

Description

`biplot.pcss.core` generates biplots of scores of genotypes with or without vectors for traits from the output of `pcss.core`.

Usage

```
## S3 method for class 'pcss.core'
biplot(
  x,
  ndim = 3,
  highlight.core = c("size", "variance", "logistic", "none"),
  show.traits = c("all", "none", "quantitative", "qualitative"),
  qual.scale = 1,
  quant.scale = 1,
  point.alpha = 0.8,
  segment.alpha = 0.8
)
```

Arguments

`ndim`	The number of dimensions for which biplots have to plotted.
`highlight.core`	The core collection to be highlighted. Either `"size"`, `"variance"`, `"logistic"`, or `"none"`. See **Details**.
`show.traits`	Which kind of the traits to be shown in the biplot. Either `"all"`, `"none"`, `"quantitative"` or `"qualitative"`.
`qual.scale`	A scale factor to be applied to qualitative trait coordinates plotted in biplot.
`quant.scale`	A scale factor to be applied to quantitative trait coordinates plotted in biplot.
`point.alpha`	Alpha transparency value for biplot points.
`segment.alpha`	Alpha transparency value for biplot segments.
`An`	object of class `pcss.core`.

Details

Use `"size"` to highlight core collection according to the threshold `size` criterion or use `"variance"` to highlight core collection according to the variability threshold criterion or use `"logistic"` to highlight core collection generated according to inflection point of rate of progress of cumulative variability retained identified by logistic regression. Use `"none"` to not highlight any accessions.

Value

A list of biplots as `ggplot` objects.

See Also

pcss.core, plot.PCA, plot.MCA, plot.FAMD, fviz_pca, fviz_mca, fviz_famd

Examples

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Prepare example data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

library(EvaluateCore)

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")
data = cbind(Genotypes = rownames(cassava_EC), cassava_EC)
quant <- c("NMSR", "TTRN", "TFWSR", "TTRW", "TFWSS", "TTSW", "TTPW", "AVPW",
           "ARSR", "SRDM")
qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
          "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
          "PSTR")
rownames(data) <- NULL

# Convert qualitative data columns to factor
data[, qual] <- lapply(data[, qual], as.factor)


library(FactoMineR)
library(factoextra)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# With quantitative data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out1 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = NULL, eigen.threshold = NULL, size = 0.2,
                  var.threshold = 0.75)

# Plot biplot
biplot(out1, ndim = 3, highlight.core = "size", quant.scale = 3,
       point.alpha = 0.5)

# Plot biplot with FactoMineR
plot(out1$raw.out, choix=c("ind"), label  = "none", axes = c(1, 2))

plot(out1$raw.out, choix=c("ind"), label  = "none", axes = c(1, 3))

plot(out1$raw.out, choix=c("ind"), label  = "none", axes = c(2, 3))

# Plot biplot with factoextra
fviz_pca_biplot(out1$raw.out, geom.ind = "point", axes = c(1, 2))

fviz_pca_biplot(out1$raw.out, geom.ind = "point", axes = c(1, 3))

fviz_pca_biplot(out1$raw.out, geom.ind = "point", axes = c(2, 3))
```

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out2 <- pcss.core(data = data, names = "Genotypes", quantitative = NULL,
                  qualitative = qual, eigen.threshold = NULL,
                  size = 0.2, var.threshold = 0.75)

# Plot biplot
biplot(out2, ndim = 3, highlight.core = "size", qual.scale = 1,
       point.alpha = 0.5)

# Plot biplot with FactoMineR
plot(out2$raw.out, choix=c("ind"), label  = "none", axes = c(1, 2))

plot(out2$raw.out, choix=c("ind"), label  = "none", axes = c(1, 3))

plot(out2$raw.out, choix=c("ind"), label  = "none", axes = c(2, 3))

# Plot biplot with factoextra
fviz_mca_biplot(out2$raw.out, geom.ind = "point", axes = c(1, 2))

fviz_mca_biplot(out2$raw.out, geom.ind = "point", axes = c(1, 3))

fviz_mca_biplot(out2$raw.out, geom.ind = "point", axes = c(2, 3))

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (quantitative and qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out3 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = qual, eigen.threshold = NULL)

# Plot biplot
biplot(out3, ndim = 3, highlight.core = "size",
       quant.scale = 3, qual.scale = 1,
       point.alpha = 0.5)

# Plot biplot with FactoMineR
plot(out3$raw.out, choix=c("ind"), label  = "none", axes = c(1, 2))

plot(out3$raw.out, choix=c("ind"), label  = "none", axes = c(1, 3))

plot(out3$raw.out, choix=c("ind"), label  = "none", axes = c(2, 3))

# Plot biplot with factoextra

# Fix rownames
row.names(out3$raw.out$quali.var$coord) <-
  unlist(lapply(seq_along(data[, qual]),
                function(i) paste(qual[i],
                                  levels(data[, qual[i]]), sep = "_")))

fviz_famd_ind(out3$raw.out, geom = "point", axes = c(1, 2))

fviz_famd_ind(out3$raw.out, geom = "point", axes = c(1, 3))
```

```
fviz_famd_ind(out3$raw.out, geom = "point", axes = c(2, 3))
```

`contrib.pcss.core` *Plot Contribution or Loadings of Traits for each Dimension/Factor from* `pcss.core` *Output*

Description

`screeplot.pcss.core` generates bar plots of contributions or loadings ("right singular vectors") of traits for each dimension/factor from the output of `pcss.core`.

Usage

```
contrib.pcss.core(
  x,
  ndim = NULL,
  plot.loadings = FALSE,
  use.sign = TRUE,
  sort.value = TRUE
)
```

Arguments

`ndim`	The number of dimensions for which contribution or loadings of traits are to be plotted.
`plot.loadings`	If `TRUE`, the loadings or "right singular vectors" are plotted instead of contributions. Default is `FALSE`.
`use.sign`	If `TRUE`, contributions of variables are given the sign of their corresponding coordinates. Default is `TRUE`.
`sort.value`	If `TRUE`, the bars are sorted according to their value.
`An`	object of class `pcss.core`.

Value

The contributions/loadings bar plot as a `ggplot` object.

See Also

`pcss.core`

Examples

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Prepare example data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

library(EvaluateCore)

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")
data = cbind(Genotypes = rownames(cassava_EC), cassava_EC)
quant <- c("NMSR", "TTRN", "TFWSR", "TTRW", "TFWSS", "TTSW", "TTPW", "AVPW",
           "ARSR", "SRDM")
qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
          "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
          "PSTR")
rownames(data) <- NULL

# Convert qualitative data columns to factor
data[, qual] <- lapply(data[, qual], as.factor)


library(FactoMineR)
library(factoextra)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# With quantitative data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out1 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = NULL, eigen.threshold = NULL, size = 0.2,
                  var.threshold = 0.75)

# Plot contributions of genotypes - with sign - sorted
contrib.pcss.core(x = out1, ndim = 5)

# Plot contributions of genotypes - without sign - sorted
contrib.pcss.core(x = out1, ndim = 5, use.sign = FALSE)

# Plot loadings/coordinates of genotypes - with sign - sorted
contrib.pcss.core(x = out1, ndim = 5, plot.loadings = TRUE)

# Plot contributions of genotypes - with sign - unsorted
contrib.pcss.core(x = out1, ndim = 5, sort.value = FALSE)

# Plot biplot with factoextra
fviz_contrib(out1$raw.out, choice = "var", axes = 1)
fviz_contrib(out1$raw.out, choice = "var", axes = 2)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out2 <- pcss.core(data = data, names = "Genotypes", quantitative = NULL,
                  qualitative = qual, eigen.threshold = NULL,
```

```
                  size = 0.2, var.threshold = 0.75)

# Plot contributions of genotypes - with sign - sorted
contrib.pcss.core(x = out2, ndim = 5)

# Plot contributions of genotypes - without sign - sorted
contrib.pcss.core(x = out2, ndim = 5, use.sign = FALSE)

# Plot loadings/coordinates of genotypes - with sign - sorted
contrib.pcss.core(x = out2, ndim = 5, plot.loadings = TRUE)

# Plot contributions of genotypes - with sign - unsorted
contrib.pcss.core(x = out2, ndim = 5, sort.value = FALSE)

# Plot biplot with factoextra
fviz_contrib(out2$raw.out, choice = "var", axes = 1)
fviz_contrib(out2$raw.out, choice = "var", axes = 2)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (quantitative and qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out3 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = qual, eigen.threshold = NULL)

# Plot contributions of genotypes - sorted
contrib.pcss.core(x = out3, ndim = 5)

# Plot contributions of genotypes - without sign - sorted
contrib.pcss.core(x = out3, ndim = 5, use.sign = FALSE)

# Plot loadings/coordinates of genotypes - sorted
contrib.pcss.core(x = out3, ndim = 5, plot.loadings = TRUE)

# Plot contributions of genotypes - with sign - unsorted
contrib.pcss.core(x = out3, ndim = 5, sort.value = FALSE)

# Plot biplot with factoextra
# fviz_contrib(out3$raw.out, choice = "quanti.var", axes = 1)
# fviz_contrib(out3$raw.out, choice = "quali.var", axes = 1)
# fviz_contrib(out3$raw.out, choice = "quanti.var", axes = 2)
# fviz_contrib(out3$raw.out, choice = "quali.var", axes = 2)
```

`coreplot.pcss.core` *Plot the cumulative variability retained by individuals/genotypes from* `pcss.core` *Output*

Description

`coreplot.pcss.core` generates plots of cumulative variability retained by individuals/genotypes from `pcss.core` Output. The size of core collection and the corresponding cumulative variance retained are highlighted according to the criterion used.

Usage

```
coreplot.pcss.core(x, criterion = c("size", "variance", "logistic"))
```

Arguments

x	An object of class pcss.core.
criterion	The core collection generation criterion. Either "size", "variance", or "logistic". See **Details**.

Details

Use "size" to highlight core collection according to the threshold size criterion or use "variance" to highlight core collection according to the variability threshold criterion or use "logistic" to highlight core collection generated according to inflection point of rate of progress of cumulative variability retained identified by logistic regression.

Value

A plot of cumulative variability retained by individuals/genotypes as a ggplot object. In case of criterion = "logistic", a list with plots of cumulative variability retained by individuals/genotypes and rate of progress of cumulative contribution to variability. The size and variability retained by core collection are highlighted in each plot.

See Also

pcss.core

Examples

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Prepare example data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

library(EvaluateCore)

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")
data = cbind(Genotypes = rownames(cassava_EC), cassava_EC)
quant <- c("NMSR", "TTRN", "TFWSR", "TTRW", "TFWSS", "TTSW", "TTPW", "AVPW",
           "ARSR", "SRDM")
qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
          "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
          "PSTR")
rownames(data) <- NULL

# Convert qualitative data columns to factor
data[, qual] <- lapply(data[, qual], as.factor)


library(FactoMineR)
library(factoextra)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# With quantitative data
```

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out1 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = NULL, eigen.threshold = NULL, size = 0.2,
                  var.threshold = 0.75)

# For core set constituted by size criterion
coreplot.pcss.core(x = out1, criterion = "size")

# For core set constituted by variance criterion
coreplot.pcss.core(x = out1, criterion = "variance")

# For core set constituted by logistic regression criterion
coreplot.pcss.core(x = out1, criterion = "logistic")

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out2 <- pcss.core(data = data, names = "Genotypes", quantitative = NULL,
                  qualitative = qual, eigen.threshold = NULL,
                  size = 0.2, var.threshold = 0.75)

# For core set constituted by size criterion
coreplot.pcss.core(x = out2, criterion = "size")

# For core set constituted by variance criterion
coreplot.pcss.core(x = out2, criterion = "variance")

# For core set constituted by logistic regression criterion
coreplot.pcss.core(x = out2, criterion = "logistic")

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (quantitative and qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out3 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = qual, eigen.threshold = NULL)

# For core set constituted by size criterion
coreplot.pcss.core(x = out3, criterion = "size")

# For core set constituted by variance criterion
coreplot.pcss.core(x = out3, criterion = "variance")

# For core set constituted by logistic regression criterion
coreplot.pcss.core(x = out3, criterion = "logistic")
```

Description

Generate a Core Collection with Principal Component Scoring Strategy (PCSS) (Hamon and Noirot 1990; Noirot et al. 1996; Noirot et al. 2003) using qualitative and/or quantitative trait data.

Usage

```
pcss.core(
  data,
  names,
  quantitative,
  qualitative,
  eigen.threshold = NULL,
  size = 0.2,
  var.threshold = 0.75
)
```

Arguments

`data`	The data as a data frame object. The data frame should possess one row per individual and columns with the individual names and multiple trait/character data.
`names`	Name of column with the individual/genotype names as a character string.
`quantitative`	Name of columns with the quantitative traits as a character vector.
`qualitative`	Name of columns with the qualitative traits as a character vector.
`eigen.threshold`	The lower limit of the eigen value of factors to be included in the estimation. The default value is the average of all the eigen values.
`size`	The desired core set size proportion.
`var.threshold`	The desired proportion of total variability to be

Details

A core collection is constituted from an entire collection of N genotypes using quantitative data of J traits using Principal Component Scoring Strategy (PCSS) (Hamon and Noirot 1990; Noirot et al. 1996; Noirot et al. 2003) as follows:

1. Principal Component Analysis (PCA) is performed on the standardized genotype $\times$ trait data. This takes care of multicollinearity between the traits to generate J standardized and independent variables or factors or principal component.
2. Considering only a subset of factors K, the Generalized Sum of Squares (GSS) of N individuals in K factorial spaces is computed as $N \times K$.

   K can be the number of factors for which the eigen value λ is greater than a threshold value such as 1 (Kaiser-Guttman criterion) or the average of all the eigen values.
3. The contribution of the ith genotype to GSS (P_i) or total variability is calculated as below.

$$P_i = \sum_{j=1}^{K} x_{ij}^2$$

   Where x_{ij} is the component score or coordinate of the ith genotype on the jth principal component.

4. For each genotype, its relative contribution to GSS or total variabilityis computed as below.

$$CR_i = \frac{P_i}{N \times K}$$

5. The genotypes are sorted in descending order of magnitude of their contribution to GSS and then the cumulative contribution of successive genotypes to GSS is computed.
6. The core collection can then be selected by three different methods.
   (a) Selection of fixed proportion or percentage or number of the top accessions.
   (b) Selection of the top accessions that contribute up to a fixed percentage of the GSS.
   (c) Fitting a logistic regression model of the following form to the cumulative contribution of successive genotypes to GSS (Balakrishnan et al. 2000).

$$\frac{y}{A-y} = e^{a+bn}$$

The above equation can be reparameterized as below.

$$\log_e\left(\frac{y}{A-y}\right) = a + bn$$

Where, a and b are the intercept and regression coefficient, respectively; y is the cumulative contribution of successive genotypes to GSS; n is the rank of the genotype when sorted according to the contribution to GSS and A is the asymptote of the curve ($A = 100$).
The rate of increase in the successive contribution of genotypes to GSS can be computed as

$$\frac{\mathrm{d}y}{\mathrm{d}x} = by(A-y)$$

to find the point of inflection where the rate of increase starts declining.
The number of accessions included till the peak or infection point are selected to constitute the core collection.

Similarly for qualitative traits, standardized and independent variables or factors can be obtained by Correspondence Analysis (CA) on complete disjunctive table of genotype $\times$ trait data or to be specific Multiple Correspondence Analysis (MCA). In `rpcss`, this has also been extended for data sets having both quantitative and qualitative traits by implementing Factor Analysis for Mixed Data (FAMD) for obtaining standardized and independent variables or factors.

In `rpcss`, PCA, MCA and FAMD are implemented via the `FactoMineR` package. (Le et al. 2008; Husson et al. 2017).

Value

A list of class `pcss.core` with the following components.

`details`	The details of the core set generaton process.
`raw.out`	The original output of `PCA`, `CA` and `FAMD` functions of `FactoMineR`
`eigen`	A data frame with eigen values and their partial and cumulative contribution to percentage of variance.
`eigen.threshold`	The threshold eigen value used.
`rotation`	A matrix of rotation values or loadings.

scores	A matrix of scores from PCA, CA or FAMD.
variability.ret	A data frame of individuals/genotypes ordered by variability retained.
cores.info	A data frame of core set size and percentage variability retained according to the method used.

See Also

PCA, CA and FAMD

Examples

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Prepare example data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

library(EvaluateCore)

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")
data = cbind(Genotypes = rownames(cassava_EC), cassava_EC)
quant <- c("NMSR", "TTRN", "TFWSR", "TTRW", "TFWSS", "TTSW", "TTPW", "AVPW",
           "ARSR", "SRDM")
qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
          "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
          "PSTR")
rownames(data) <- NULL

# Convert qualitative data columns to factor
data[, qual] <- lapply(data[, qual], as.factor)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (quantitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out1 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = NULL, eigen.threshold = NULL, size = 0.2,
                  var.threshold = 0.75)

out1

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out2 <- pcss.core(data = data, names = "Genotypes", quantitative = NULL,
                  qualitative = qual, eigen.threshold = NULL,
                  size = 0.2, var.threshold = 0.75)

out2

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (quantitative and qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
out3 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = qual, eigen.threshold = NULL)

out3
```

`print.pcss.core` *Prints summary of* `pcss.core` *object*

Description

`pcss.core` prints to console the summary of an object of class `pcss.core` including the dimensionality reduction method used, the basic details including parameters and the information on the core sets that can be constituted.

Usage

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

Arguments

`x`	An object of class `pcss.core`.
`...`	Unused

See Also

`pcss.core`

`screeplot.pcss.core` *Plot Eigen values as a Scree Plot from* `pcss.core` *Output*

Description

`screeplot.pcss.core` generates a scree plot of eigen values from the output of `pcss.core`.

Usage

```
## S3 method for class 'pcss.core'
screeplot(x, ndim = NULL, show.values = TRUE)
```

Arguments

`ndim`	The number of eigen values to be plotted in the scree plot.
`show.values`	If `TRUE`, the eigen values are shown in the plot as annotation labels. Default is `TRUE`.
`An`	object of class `pcss.core`.

Value

The scree plot as a ggplot object.

See Also

pcss.core, fviz_screeplot

Examples

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Prepare example data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

library(EvaluateCore)

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")
data = cbind(Genotypes = rownames(cassava_EC), cassava_EC)
quant <- c("NMSR", "TTRN", "TFWSR", "TTRW", "TFWSS", "TTSW", "TTPW", "AVPW",
           "ARSR", "SRDM")
qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
          "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
          "PSTR")
rownames(data) <- NULL

# Convert qualitative data columns to factor
data[, qual] <- lapply(data[, qual], as.factor)


library(FactoMineR)
library(factoextra)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# With quantitative data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out1 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = NULL, eigen.threshold = NULL, size = 0.2,
                  var.threshold = 0.75)

# Plot scree plot
screeplot.pcss.core(x = out1)

# Plot biplot with factoextra
fviz_screeplot(out1$raw.out)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out2 <- pcss.core(data = data, names = "Genotypes", quantitative = NULL,
                  qualitative = qual, eigen.threshold = NULL,
                  size = 0.2, var.threshold = 0.75)
```

```
# Plot scree plot
screeplot.pcss.core(x = out2)

# Plot biplot with factoextra
fviz_screeplot(out2$raw.out)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (quantitative and qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out3 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = qual, eigen.threshold = NULL)

# Plot scree plot
screeplot.pcss.core(x = out3)

# Plot biplot with factoextra
fviz_screeplot(out3$raw.out)
```

`subset.pcss.core` *Fetch the names of individuals/genotypes in the core set generated from* `pcss.core` *Output*

Description

`subset.pcss.core` returns names of individuals/genotypes in the core collection from `pcss.core` Output.

Usage

```
## S3 method for class 'pcss.core'
subset(x, criterion = c("size", "variance", "logistic"))
```

Arguments

`x`	An object of class `pcss.core`.
`criterion`	The core collection generation criterion. Either `"size"`, `"variance"`, or `"logistic"`. See **Details**.

Details

Use `"size"` to return names of individuals/genotypes in the core collection according to the threshold `size` criterion or use `"variance"` to return names according to the variability threshold criterion or use `"logistic"` to return names according to inflection point of rate of progress of cumulative variability retained identified by logistic regression.

Value

The names of individuals/genotypes in the core collection as a character vector.

See Also

pcss.core

Examples

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Prepare example data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

library(EvaluateCore)

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")
data = cbind(Genotypes = rownames(cassava_EC), cassava_EC)
quant <- c("NMSR", "TTRN", "TFWSR", "TTRW", "TFWSS", "TTSW", "TTPW", "AVPW",
           "ARSR", "SRDM")
qual <- c("CUAL", "LNGS", "PTLC", "DSTA", "LFRT", "LBTEF", "CBTR", "NMLB",
          "ANGB", "CUAL9M", "LVC9M", "TNPR9M", "PL9M", "STRP", "STRC",
          "PSTR")
rownames(data) <- NULL

# Convert qualitative data columns to factor
data[, qual] <- lapply(data[, qual], as.factor)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# With quantitative data
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out1 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = NULL, eigen.threshold = NULL, size = 0.2,
                  var.threshold = 0.75)

# Core sets
out1$cores.info

# Fetch genotype names of core set by size criterion
subset.pcss.core(x = out1, criterion = "size")

# Fetch genotype names of core set by variance criterion
subset.pcss.core(x = out1, criterion = "variance")

# Fetch genotype names of core set by logistic regression criterion
subset.pcss.core(x = out1, criterion = "logistic")

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out2 <- pcss.core(data = data, names = "Genotypes", quantitative = NULL,
                  qualitative = qual, eigen.threshold = NULL,
                  size = 0.2, var.threshold = 0.75)

# Core sets
out2$cores.info
```

```
# Fetch genotype names of core set by size criterion
subset.pcss.core(x = out2, criterion = "size")

# Fetch genotype names of core set by variance criterion
subset.pcss.core(x = out2, criterion = "variance")

# Fetch genotype names of core set by logistic regression criterion
subset.pcss.core(x = out2, criterion = "logistic")
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Get core sets with PCSS (quantitative and qualitative data)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

out3 <- pcss.core(data = data, names = "Genotypes",
                  quantitative = quant,
                  qualitative = qual, eigen.threshold = NULL)

# Core sets
out3$cores.info

# Fetch genotype names of core set by size criterion
subset.pcss.core(x = out3, criterion = "size")

# Fetch genotype names of core set by variance criterion
subset.pcss.core(x = out3, criterion = "variance")

# Fetch genotype names of core set by logistic regression criterion
subset.pcss.core(x = out3, criterion = "logistic")
```

Index