Package 'LEAVcore'

May 31, 2026

Title Constitution of Core Collections using Length of Encoded Attribute Values

Version 0.1.0

Description Construct core collections using the information measure 'Length of Encoded Attribute Values' (LEAV) using qualitative and/or quantitative trait data as described by Balakrishnan and Suresh (2001a) <https://indianjournals.com/article/ijpgr-14-1-006> and (2001b) <https://indianjournals.com/article/ijpgr-14-3-005>.

License GPL (>= 2)

Encoding UTF-8

BuildManual TRUE

Imports mathjaxr,
Rdpack,
dplyr,
stats,
stratification

Suggests EvaluateCore,
knitr,
rmarkdown,
SampleCore,
pander

RdMacros mathjaxr,
Rdpack

Copyright 2024-2026, ICAR-NBPGR

URL https://github.com/aravind-j/LEAVcore
https://aravind-j.github.io/LEAVcore/

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

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

Contents

`inflen.qual` *Compute Information Length for Qualitative Traits*

Description

The function `inflen.qual` computes the length of information code that can indicate the possession of a descriptor state of a qualitative trait (Wallace and Boulton 1968; Balakrishnan and Suresh 2001; Balakrishnan and Suresh 2001; Balakrishnan and Nair 2003).

Usage

```
inflen.qual(x, freq, adj = TRUE)
```

Arguments

`x`	Data of a qualitative trait for accessions in a collection as a vector of type factor.
`freq`	The target absolute frequencies of the descriptor states of the qualitative trait `x` in the subset of all accessions.
`adj`	logical. If `TRUE`, the proportion estimates are slightly biased to include zero frequency descriptor states in the computation (See **Details**). Default is `TRUE`.

Details

For each qualitative trait/descriptor d the probability of occurrence of a descriptor state m in the in a subset t is estimated as

$$p_{m,d,t} = \frac{n_{m,d,t} + 1}{n_{d,t} + M_d}$$

Where, $n_{m,d,t}$ is the number of accessions with m state of trait d in subset t, $n_{d,t}$ is the number of accessions with any known state of trait d in subset t,i.e. the number of accessions in subset t and M_d is the number of descriptor states of trait d.

This is a slightly biased estimate to include zero frequency descriptor states in the computation. The actual estimate is

$$p_{m,d,t} = \frac{n_{m,d,t}}{n_{d,t}}$$

Now the length of the information code that can optimally indicate the possession of descriptor state m of trait d in the subset t is computed as

$$c_{m,d,t} = -\ln p_{m,d,t}$$

Value

A data frame with 2 columns:

x The qualitative trait data

inflen Information length computed

References

Balakrishnan R, Nair NV (2003). "Strategies for developing core collections of sugarcane (*Saccharum officinarum* L.) germplasm-comparison of sampling from diversity groups constituted by three different methods." *Plant Genetic Resources Newsletter*, **134**, 33–41.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part II. Using an information measure for obtaining a core sample with pre-determined diversity levels for several descriptors simultaneously." *Indian Journal of Plant Genetic Resources*, **14**(1), 32–42.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part III. Obtaining diversity groups based on an information measure." *Indian Journal of Plant Genetic Resources*, **14**(3), 342–349.

Wallace CS, Boulton DM (1968). "An information measure for classification." *The Computer Journal*, **11**(2), 185–194.

See Also

inflen.quant

Examples

```
suppressPackageStartupMessages(library(EvaluateCore))

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")

# Data of 'Colour of unexpanded apical leaves' qualitative trait
CUAL <- as.factor(cassava_EC$CUAL)

# Get frequencies based on sample size
prop <-  prop.adj(CUAL, method = "sqrt")
size.prop <- 0.2
size.count <- ceiling(size.prop * length(CUAL))
CUALfreq <- round(prop * size.count)

# Compute information length
CUALinflen <- inflen.qual(x = CUAL, freq = CUALfreq, adj = TRUE)

head(CUALinflen)
```

`inflen.quant` *Compute Information Length for Quantitative Traits*

Description

The function `inflen.quant` computes the length of information code that can indicate the possession of a specific value by a quantitative trait (Wallace and Boulton 1968; Balakrishnan and Suresh 2001; Balakrishnan and Suresh 2001; Balakrishnan and Nair 2003).

Usage

```
inflen.quant(x, mean, sd, e = 1)
```

Arguments

`x`	Data of a quantitative trait for accessions in a collection as a numeric vector.
`mean`	The target mean.
`sd`	The target standard deviation
`e`	The least count of measurement for the quantitative trait (i.e. the accuracy of measurement).

Details

For each quantitative trait d, it is assumed that it is normally distributed within subset t with mean $\mu_{d,t}$ and the standard deviation $\sigma_{d,t}$ estimated as below.

$$\mu_{d,t} = \frac{\sum x_{d,s}}{n_{d,t}}$$

$$\sigma_{d,t} = \sqrt{\frac{\sum (x_{d,s} - \mu_{d,t})^2}{n_{d,t} - 1}}$$

From this, a distribution normalizing constant $g_{d,t}$ can be estimated as

$$g_{d,t} = \ln\left(\frac{\sigma_{d,t}}{K \cdot \varepsilon_d}\right)$$

Where $K = \frac{1}{\sqrt{2\Pi}}$, ε_d is the least count of measurement of the descriptor d. i.e. x is measured to an accuracy of $\pm\varepsilon_d$.

The probability of getting a measurement x from a distribution of mean μ and variance σ is approximately as follows.

$$K \cdot \frac{\varepsilon_d}{\sigma_{d,t}} \cdot e^{\frac{-(x_{d,s} - \mu_{d,t})^2}{2\sigma_{d,t}^2}}$$

Now the length of the information code that can optimally indicate the possession of a value x by the trait d is computed as follows:

$$c_{x,d,t} = g_{d,t} + \frac{(x_{d,s} - \mu_{d,t})^2}{2\sigma_{d,t}^2}$$

Value

A data frame with 2 columns:

x The quantitative trait data

inflen Information length computed

References

Balakrishnan R, Nair NV (2003). "Strategies for developing core collections of sugarcane (*Saccharum officinarum* L.) germplasm-comparison of sampling from diversity groups constituted by three different methods." *Plant Genetic Resources Newsletter*, **134**, 33–41.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part II. Using an information measure for obtaining a core sample with pre-determined diversity levels for several descriptors simultaneously." *Indian Journal of Plant Genetic Resources*, **14**(1), 32–42.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part III. Obtaining diversity groups based on an information measure." *Indian Journal of Plant Genetic Resources*, **14**(3), 342–349.

Wallace CS, Boulton DM (1968). "An information measure for classification." *The Computer Journal*, **11**(2), 185–194.

See Also

`inflen.qual`

Examples

```
suppressPackageStartupMessages(library(EvaluateCore))

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")

# Data of 'Average plant weight' quantitative trait
AVPW <- cassava_EC$AVPW

# Compute information length
AVPWinflen <- inflen.quant(x = AVPW, mean = 4, sd = 3.25, e = 1)

head(AVPWinflen)
```

LEAV *Compute the Length of Encoded Attribute Values*

Description

For accessions in a collection compute the Length of Encoded Attribute Values (LEAV) information measure from qualitative and quantitative trait data (Wallace and Boulton 1968; Balakrishnan and Suresh 2001; Balakrishnan and Suresh 2001; Balakrishnan and Nair 2003).

Usage

```
LEAV(
  data,
  names,
  quantitative = NULL,
  qualitative = NULL,
  freq,
  adj = TRUE,
  mean,
  sd,
  e
)
```

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 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.
freq	A named list with the target absolute frequencies of the descriptor states for each qualitative trait specified in qualitative. The list names should be same as qualitative.
adj	logical. If TRUE, the proportion estimates are slightly biased to include zero frequency descriptor states in the computation (See **Details**). Default is TRUE.
mean	A named numeric vector of target means for each quantitative trait specified in quantitative. The list names should be same as quantitative.
sd	A named numeric vector of target standard deviation for each quantitative trait specified in quantitative. The list names should be same as quantitative.
e	A named numeric vector of least count of measurement for each quantitative trait specified in quantitative. The list names should be same as quantitative.

Details

For each accession s in the collection, the message length F_s to optimally encode all the d traits/descriptors is computed as follows using the joint density distribution of the whole collection.

$$F_s = l_t + \sum_{i=1}^{p} c_{m_s,d_i,t} + \sum_{j=1}^{q} c_{x_s,d_j,t}$$

Here, the first expression l_t is the message length for the subset t to which an accession belongs when there are N accessions in the whole collection and n_t accessions in the subset t.

$$l_t = l_t = -\ln\left(\frac{N}{n_t}\right)$$

Similarly $\sum_{i=1}^{p} c_{m_s,d_i,t}$ is sum of the optimum message length for p qualitative traits, $\sum_{j=1}^{q} c_{x_s,d_j,t}$ is sum of the optimum optimum message length for q quantitative traits. See inflen.qual and inflen.quant for more details.

Value

A data frame with one row per accession in `data` and the following columns:

`names` Accession identifiers, as specified by the `names` argument.

`lt` The log-ratio message length term, $\log(N/n)$, where N is the total number of accessions in `data` and n is the sum of frequencies in `freq`.

`<qualitative traits>` One column per trait specified in `qualitative`, giving the information length $-\log(p_k)$ for the level k of that trait observed for each accession.

`<quantitative traits>` One column per trait specified in `quantitative`, giving the Gaussian information length $\log(\sigma/c\varepsilon) + (x-\mu)^2/2\sigma^2$ for each accession, where $c = 1/\sqrt{2\pi}$.

`LEAV` The total information length for each accession, equal to the row sum of `lt` and all trait information length columns.

References

Balakrishnan R, Nair NV (2003). "Strategies for developing core collections of sugarcane (*Saccharum officinarum* L.) germplasm-comparison of sampling from diversity groups constituted by three different methods." *Plant Genetic Resources Newsletter*, **134**, 33–41.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part II. Using an information measure for obtaining a core sample with pre-determined diversity levels for several descriptors simultaneously." *Indian Journal of Plant Genetic Resources*, **14**(1), 32–42.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part III. Obtaining diversity groups based on an information measure." *Indian Journal of Plant Genetic Resources*, **14**(3), 342–349.

Wallace CS, Boulton DM (1968). "An information measure for classification." *The Computer Journal*, **11**(2), 185–194.

See Also

`inflen.qual`, `inflen.quant`

Examples

```
suppressPackageStartupMessages(library(EvaluateCore))

# Get data from EvaluateCore
data("cassava_EC", package = "EvaluateCore")

cassava_EC <- cassava_EC[sample(1:nrow(cassava_EC), 500), ]

cassava_EC <- 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")
```

```
cassava_EC[, qual] <- lapply(cassava_EC[, qual], as.factor)

size <- 0.2

freq_list <- lapply(qual, function(x) {
  prop <-  prop.adj(cassava_EC[, x], method = "sqrt")
  size.count <- ceiling(size * length(x))
  round_preserve_sum(prop * size.count)
})
names(freq_list) <- qual

mean_vec <- sapply(cassava_EC[, quant],
                   function(x) {
                     floor(mean(x))
                   })
names(mean_vec) <- quant

sd_vec <- sapply(cassava_EC[quant],
             function(x) {
               round(sd(x), 1)
             })
names(sd_vec) <- quant

e_vec <- rep(1, length(quant))
names(e_vec) <- quant


# Compute LEAV
LEAV_cassava <- LEAV(data = cassava_EC, names = "genotypes",
                     quantitative = quant, qualitative = qual,
                     freq = freq_list, adj = TRUE,
                     mean = mean_vec, sd = sd_vec, e = e_vec)

head(LEAV_cassava)
```

LEAVcore_functions *Generate Core collections with Length of Encoded Attribute Values*

Description

Based on Length of Encoded Attribute Values (LEAV) (Wallace and Boulton 1968; Balakrishnan and Suresh 2001; Balakrishnan and Suresh 2001; Balakrishnan and Nair 2003) estimated from qualitative and/or quantitative trait data, core collections can be generated by the three following methods.

Method I Classification based on pre-determined diversity represented by LEAV estimates implemented in LEAVcore1.

Method II Purposive selection of accessions with highest rank of LEAV estimates implemented in LEAVcore2.

Method III Stratified sampling of accessions from diversity groups/strata computed from LEAV estimates partially implemented in LEAVcore3.

Usage

```
LEAVcore1(
  data,
  names,
  quantitative = NULL,
  qualitative = NULL,
  size,
  prop.adj = c("none", "log", "sqrt"),
  e,
  always.selected = NULL
)

LEAVcore2(
  data,
  names,
  quantitative = NULL,
  qualitative = NULL,
  size,
  prop.adj = c("none", "log", "sqrt"),
  e,
  always.selected = NULL
)

LEAVcore3(
  data,
  names,
  quantitative = NULL,
  qualitative = NULL,
  size,
  prop.adj = c("none", "log", "sqrt"),
  e,
  always.selected = NULL
)
```

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 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.
`size`	The desired core set size proportion.
`prop.adj`	The method for relative frequency transformation for qualitative traits. Either `"none"` for no transformation or `"log"` for log-frequency transformation or `"sqrt"` for square root-proportion transformation (see `prop.adj`).
`e`	A named numeric vector of least count of measurement for each quantitative trait specified in `quantitative`. The list names should be same as `quantitative`.
`always.selected`	Names of accessions to be always included in the core set as a character vector.

Details

Balakrishnan and Suresh (2001); Balakrishnan and Suresh (2001) describe three different methods of constructing core collections from estimates of Length of Encoded Attribute Values.

Method I: Classification based on pre-determined diversity represented by LEAV estimates: This is an objective classification scheme that assigns accessions to either a "core" or "non-core" group based on which group model they best fit.

The target frequency patterns for qualitative traits and distribution parameters for quantitative traits are determined first for the two groups: the Core and the Non-Core.

Target proportions for the core group are estimated from the base proportions of the qualitative trait levels. These may be subjected to transformations if required according to `"prop.adj"` argument to increase rare trait representation. Target counts are set by scaling these to the total count and capping them at the actual frequency available in the collection. Similarly for the non-core group, the target proportions are determined by subtracting the core model's frequencies from the total counts of each trait level in the entire collection.

The target distribution for the core group is modeled by applying a Gaussian kernel density function to the quantitative trait data, scaled to the core size. The non-core parameters are set to the actual mean and standard deviation of the entire collection.

Based on these target values, the message length (F) is estimated for each accession against both the models using `LEAV`. An accession is assigned to the core if $F_{core} \leq F_{non-core}$. If more accessions are selected than the target core size, the core is refined by ranking individuals by F_{core} values in ascending order and retaining only the top matches.

Method II: Purposive selection of accessions with highest rank of LEAV estimates: This is a directed selection method that captures the most unique and dispersed accessions to maximize diversity and reduce redundancy.

Here the LEAV index for every accession relative to the entire base collection is first estimated. Then the accessions are ranked in descending order of their LEAV estimates.Finally the core collection is constituted by selecting a pre-determined number of top-ranked accessions according to `"size"` argument.

Method III: Stratified sampling of accessions from diversity groups/strata computed from LEAV estimates: This is a two-step approach that first organizes the collection into optimized diversity groups based on LEAV estimates followed by a group-wise representative sampling.

Here also the LEAV index for every accession relative to the entire base collection is first estimated. These estimates are then divided into L strata using the Dalenius formula to minimize pooled variance (Dalenius and Hodges 1959).

The number of entries to be sampled from each stratum is then determined followed by stratified selection from each group to reach the final core size.

In `LEAVcore3`, only the stratification based on LEAV estimates is implemented. The downstream steps for allocation (allocate.basic, allocate.diversity, allocate.distance) and stratified selection (select.random, select.diversity, select.distance) are available from the sister package **SampleCore**.

Value

`LEAVcore1` returns a data frame with one row per accession in `data` and the following columns:

`names` Accession identifiers, as specified by the `names` argument.

`LEAV_core` The total LEAV score for each accession computed under the core group parameterisation (frequencies and moments estimated from the target core subset).

`LEAV_noncore` The total LEAV score for each accession computed under the non-core group parameterisation (frequencies and moments estimated from the remainder of the collection).

`always.selected` A logical vector indicating whether the accession was pre-specified in `always.selected`.

`core` A logical vector indicating whether the accession is selected into the core collection, either because `LEAV_core` $\leq$ `LEAV_noncore` (selected by the method) or because it appears in `always.selected`.

`LEAVcore2` returns a data frame with one row per accession in `data`, sorted in decreasing order of LEAV score, with the following columns:

`names` Accession identifiers, as specified by the `names` argument.

`lt` The log-ratio message length term $\log(N/n)$, where N is the total number of accessions in `data` and n is `size.count`.

`<trait columns>` One column per trait specified in `qualitative` and `quantitative`, giving the per-accession information length for that trait.

`LEAV` The total LEAV score for each accession, equal to the row sum of `lt` and all trait information length columns.

`always.selected` A logical vector indicating whether the accession was pre-specified in `always.selected`.

`core` A logical vector indicating whether the accession is selected into the core collection, either as one of the top `size.count` ranked accessions among non-`always.selected` accessions or because it appears in `always.selected`.

`LEAVcore3` returns a data frame with one row per accession in `data`, sorted in decreasing order of LEAV score, with the following columns:

`names` Accession identifiers, as specified by the `names` argument.

`lt` The log-ratio message length term $\log(N/n)$, where N is the total number of accessions in `data` and n is `size.count`.

`<trait columns>` One column per trait specified in `qualitative` and `quantitative`, giving the per-accession information length for that trait.

`LEAV` The total LEAV score for each accession, equal to the row sum of `lt` and all trait information length columns.

`LEAVStrata` An integer stratum identifier assigned by the Dalenius-Hodges cumulative root frequency method (Dalenius and Hodges 1959), indicating the stratum to which each accession belongs for proportional sampling. `NA` for accessions in `always.selected`, which are excluded from stratification.

`always.selected` A logical vector indicating whether the accession was pre-specified in `always.selected` and is therefore excluded from stratification.

References

Balakrishnan R, Nair NV (2003). "Strategies for developing core collections of sugarcane (*Saccharum officinarum* L.) germplasm-comparison of sampling from diversity groups constituted by three different methods." *Plant Genetic Resources Newsletter*, **134**, 33–41.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part II. Using an information measure for obtaining a core sample with pre-determined diversity levels for several descriptors simultaneously." *Indian Journal of Plant Genetic Resources*, **14**(1), 32–42.

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part III. Obtaining diversity groups based on an information measure." *Indian Journal of Plant Genetic Resources*, **14**(3), 342–349.

Dalenius T, Hodges JL (1959). "Minimum variance stratification." *Journal of the American Statistical Association*, **54**(285), 88–101.

Wallace CS, Boulton DM (1968). "An information measure for classification." *The Computer Journal*, **11**(2), 185–194.

See Also

inflen.qual, inflen.quant, LEAV, allocate.basic, allocate.distance, allocate.diversity, select.random, select.distance, select.diversity

Examples

```
suppressPackageStartupMessages(library(EvaluateCore))

# Get data from EvaluateCore
data("cassava_EC", package = "EvaluateCore")

cassava_EC <- 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")

cassava_EC[, qual] <- lapply(cassava_EC[, qual], as.factor)

e_vec <- rep(1, length(quant))
names(e_vec) <- quant

mand_accns <-
  c("TMe-2018", "TMe-801", "TMe-3191", "TMe-1830", "TMe-1790")

table(cassava_EC$genotypes %in% mand_accns)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Method I
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

LEAVcore1_out <-
  LEAVcore1(data = cassava_EC, names = "genotypes",
            quantitative = quant, qualitative = qual,
            size = 0.2, prop.adj = "log", e = e_vec,
            always.selected = mand_accns)

head(LEAVcore1_out)

# Selected accessions for core
core1 <- LEAVcore1_out[LEAVcore1_out$core == TRUE, "genotypes"]
```

```
core1

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Method II
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

LEAVcore2_out <-
  LEAVcore2(data = cassava_EC, names = "genotypes",
            quantitative = quant, qualitative = qual,
            size = 0.2, prop.adj = "log", e = e_vec,
            always.selected = mand_accns)

head(LEAVcore2_out)

# Selected accessions for core
core2 <- LEAVcore2_out[LEAVcore2_out$core == TRUE, "genotypes"]

core2

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Method III
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

LEAVcore3_out <-
  LEAVcore3(data = cassava_EC, names = "genotypes",
            quantitative = quant, qualitative = qual,
            size = 0.2, prop.adj = "log", e = e_vec,
            always.selected = mand_accns)

head(LEAVcore3_out)

# Strata/Group-wise counts
table(LEAVcore3_out$LEAVStrata)

# Sample accessions from strata to form core set using SampleCOre
suppressPackageStartupMessages(library(SampleCore))

# Append LEAV strata to original data
data <- merge.data.frame(cassava_EC,
                         LEAVcore3_out[, c("genotypes", "LEAVStrata",
                                           "always.selected")],
                         by = "genotypes")
data$LEAVStrata <- as.factor(data$LEAVStrata)

# Use log allocation
log_alloc <-
  allocate.basic(data = data[data$always.selected != TRUE, ],
                 names = "genotypes",
                 group = "LEAVStrata", method = "log",
                 size = 0.2)

# Use random selection
set.seed(123)
sel_random_out <-
  select.random(data = data[data$always.selected != TRUE, ],
                names = "genotypes",
                group = "LEAVStrata", alloc = log_alloc,
```

```
                # Already included in LEAVcore3_out
                always.selected = NULL)

# Append always selected accessions
core3 <-
  c(sel_random_out,
    list(always.selected =
         LEAVcore3_out[LEAVcore3_out$always.selected == TRUE,
                       "genotypes"]))
# Final core
core3
```

`prop.adj` *Relative Frequency Adjustments*

Description

Compute and transform relative frequencies for a qualitative trait in a germplasm collection by the following methods (Balakrishnan and Suresh 2001):

- Square root-proportion
- Log-frequency

Usage

```
prop.adj(x, method = c("none", "log", "sqrt"), size.count = NULL)
```

Arguments

`x`	Data of a qualitative trait for accessions in a collection as a vector of type factor.
`method`	The method for transformation. Either `"none"` for no transformation or `"log"` for log-frequency transformation or `"sqrt"` for square root-proportion transformation.
`size.count`	A positive integer specifying the target size of the core collection. The sum of frequencies allocated across levels of each qualitative trait will not exceed this value, and serves as the upper bound for iterative proportion clamping when `size.count` is supplied. If `NULL`, no clamping is performed and the adjusted proportions are returned as-is.

Details

If p_i is the relative frequency of the ith descriptive state for a qualitative trait in a collection, then the square root-proportion transformed relative q_i is computed as

$$q_i = \frac{\sqrt{p_i}}{\sum_{i=1}^{s} \sqrt{p_i}}$$

Where s is the number of possible descriptor states for the qualitative trait in the collection.

Similarly, the log-frequency transformed relative q_i is computed as

$$q_i = \frac{\log(F_i + k)}{\sum_{i=1}^{s} \log(F_i + k)}$$

Where F_i is the absolute frequency of the ith descriptive state for a qualitative trait in a collection. It is incremented by a constant $k = 0.000001$ prior to log transformation. This ensures that singleton descriptor states (where $F_i = 1$) yield a small but non-zero proportion rather than being assigned a zero proportion due to $\log(1) = 0$, which would otherwise exclude all accessions of that descriptor state from core selection irrespective of `size.count`.

When `size.count` is supplied, the transformed proportions q_i are subject to iterative clamping to ensure that the implied frequency $q_i \times n$ for any descriptor state i does not exceed its actual count in the collection, where n is `size.count`. Excess proportion from clamped states is redistributed proportionally among unclamped states and the process repeats until no state exceeds its maximum allowable proportion F_i/n.

Value

The relative frequencies as a named numeric vector.

References

Balakrishnan R, Suresh KK (2001). "Strategies for developing core collections of safflower (*Carthamus tinctorius* L.) germplasm-part II. Using an information measure for obtaining a core sample with pre-determined diversity levels for several descriptors simultaneously." *Indian Journal of Plant Genetic Resources*, **14**(1), 32–42.

Examples

```
suppressPackageStartupMessages(library(EvaluateCore))

library(EvaluateCore)

# Get data from EvaluateCore

data("cassava_EC", package = "EvaluateCore")

# Data of 'Colour of unexpanded apical leaves' qualitative trait
CUAL <- as.factor(cassava_EC$CUAL)

# Raw relative frequencies
prop.adj(CUAL, method = "none")

# Square root-proportion transformed relative frequencies
prop.adj(CUAL, method = "sqrt")

# Square log-frequency transformed relative frequencies
prop.adj(CUAL, method = "log")
```

round_preserve_sum *Round Numeric Values While Preserving a Target Sum*

Description

Applies the Hamilton (largest remainder or Hare-Niemeyer or Vinton) rounding method (Balinski and Young 2001) to a numeric vector so that the rounded values sum to a specified target.

Usage

```
round_preserve_sum(x, target = round(sum(x)))
```

Arguments

x	A numeric vector to round.
target	A numeric scalar giving the desired sum of the rounded values. Defaults to round(sum(x)).

Details

Values are first rounded down using floor(), and the remaining deficit is allocated to elements with the largest fractional parts.

Value

An numeric vector of the same length as x, where the elements sum to target.

References

Balinski ML, Young HP (2001). *Fair Representation: Meeting the Ideal of One Man, One Vote*. Brookings Institution Press. ISBN 978-0-8157-0111-8.

Examples

```
round_preserve_sum(c(1.2, 2.7, 3.5))

round_preserve_sum(c(10.4, 10.4, 10.2), target = 32)
```

Index