Package 'metapod'

May 10, 2024

Version 1.12.0

Date 2023-12-22

Title Meta-Analyses on P-Values of Differential Analyses

Imports Rcpp

Suggests testthat, knitr, BiocStyle, rmarkdown

LinkingTo Rcpp

biocViews MultipleComparison, DifferentialPeakCalling

Description Implements a variety of methods for combining p-values in differential analyses of genome-scale datasets. Functions can combine p-values across different tests in the same analysis (e.g., genomic windows in ChIP-seq, exons in RNA-seq) or for corresponding tests across separate analyses (e.g., replicated comparisons, effect of different treatment conditions). Support is provided for handling log-transformed input p-values, missing values and weighting where appropriate.

License GPL-3

NeedsCompilation yes

SystemRequirements C++11

VignetteBuilder knitr

RoxygenNote 7.1.1

git_url https://git.bioconductor.org/packages/metapod

git_branch RELEASE_3_19

git_last_commit 4f07cb9

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-09

Author Aaron Lun [aut, cre]

Maintainer Aaron Lun <infinite.monkeys.with.keyboards@gmail.com>

Contents

averageGroupedStats . . . 2
averageParallelStats . . . 3
combineGroupedPValues . . . 4
combineParallelPValues . . . 6
countGroupedDirection . . . 7
countParallelDirection . . . 9
groupedBerger . . . 11
groupedFisher . . . 12
groupedHolmMin . . . 14
groupedPearson . . . 16
groupedSimes . . . 17
groupedStouffer . . . 19
groupedWilkinson . . . 21
parallelBerger . . . 23
parallelFisher . . . 24
parallelHolmMin . . . 26
parallelPearson . . . 28
parallelSimes . . . 29
parallelStouffer . . . 31
parallelWilkinson . . . 33
summarizeGroupedDirection . . . 35
summarizeParallelDirection . . . 36

Index **39**

`averageGroupedStats` *Average grouped statistics*

Description

Average grouped statistics with consideration of weights and protection from missing values.

Usage

```
averageGroupedStats(values, grouping, weights = NULL)
```

Arguments

`values`	A numeric vector containing statistics for individual tests.
`grouping`	A vector or factor of length equal to `p.values`, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of `p.values` belonging to that group. This assumes that `p.values` is ordered such that all entries in the same group are adjacent to each other.
`weights`	A numeric vector of length equal to `p.values`, containing a positive weight for each test. Alternatively `NULL`, in which case equal weights are assigned to all tests.

Details

If `weights` is supplied, a weighted average is computed for each group. If `values` contains missing values, they are ignored and will not contribute to the average.

Value

A named numeric vector of (weighted) averages, of length equal to the number of unique levels in `grouping`.

Author(s)

Aaron Lun

See Also

`averageParallelStats`, for the parallel counterpart.

Examples

```
grouping <- sample(LETTERS, 20, replace=TRUE)
averageGroupedStats(1:20, grouping)
averageGroupedStats(1:20, grouping, weights=runif(20))
```

`averageParallelStats` *Average parallel statistics*

Description

Average parallel statistics with consideration of weights and protection from missing values.

Usage

```
averageParallelStats(values, weights = NULL)
```

Arguments

`values`	A list of numeric vectors of the same length, containing statistics from parallel tests.
`weights`	A numeric vector of positive weights, with one value per vector in `....` Each weight is applied to all entries of itscorresponding vector, i.e., all p-values in that vector receive the same weight. Alternatively, a list of numeric vectors of weights with the same structure as `p.values`. Each p-value is then assigned the weight in the corresponding entry of `weights`. Alternatively `NULL`, in which case all p-values are assigned equal weight.

Details

If `weights` is supplied, a weighted average is computed for each parallel set of tests. If `values` contains missing values, they are ignored and will not contribute to the average.

Value

A numeric vector of (weighted) averages, of length equal to the lengths of the vectors in `values`.

Author(s)

Aaron Lun

See Also

`averageParallelStats`, for the parallel counterpart.

Examples

```
averageParallelStats(list(1:10, 2:11))
averageParallelStats(list(1:10, 2:11), weights=c(2:1))
averageParallelStats(list(1:10, rep(NA, 10)))
```

---

combineGroupedPValues *Combine grouped p-values*

Description

Combine p-values from grouped hypothesis tests using a variety of meta-analysis methods. Each group of p-values is defined as those assigned to the same level of the grouping factor.

Usage

```
combineGroupedPValues(
  p.values,
  grouping,
  method = c("simes", "holm-min", "berger", "fisher", "pearson", "wilkinson",
    "stouffer"),
  weights = NULL,
  log.p = FALSE,
  min.n = 1,
  min.prop = 0.5
)
```

Arguments

p.values	A numeric vector containing p-values for individual tests.
grouping	A vector or factor of length equal to p.values, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of p.values belonging to that group. This assumes that p.values is ordered such that all entries in the same group are adjacent to each other.
method	String specifying the method to use to combine p-values.
weights	A numeric vector of length equal to p.values, containing a positive weight for each test. Alternatively NULL, in which case equal weights are assigned to all tests.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.
min.n	Integer scalar specifying the minimum number of individual nulls to reject when testing the joint null.
min.prop	Numeric scalar in [0, 1], specifying the minimum proportion of individual nulls to reject when testing the joint null.

Details

min.prop and min.n only have an effect for method="wilkinson" and "holm-min".

weights only has an effect for method="simes", "holm-min" and "stouffer".

Value

A list containing:

- p.value, a named numeric vector of length equal to the number of unique levels in grouping. This contains the combined p-value for each group, log-transformed if log.p=TRUE. Each entry is named according to the group.
- representative, a named integer scalar specifying the index of representative test for each group. Each index refers to an entry of p.values and is named according to its group.
- influential, a logical vector of length equal to p.values. Entries are TRUE for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

Examples

```
p <- runif(10000)
g <- sample(100, 10000, replace=TRUE)

fish <- combineGroupedPValues(p, g, method="fisher")
hist(fish$p.value)

z <- combineGroupedPValues(p, g, method="stouffer", weights=rexp(10000))
```

```
hist(z$p.value)

simes <- combineGroupedPValues(p, g, method="simes")
hist(simes$p.value)

berger <- combineGroupedPValues(p, g, method="berger")
hist(berger$p.value)
```

`combineParallelPValues` *Combine parallel p-values*

Description

Combine p-values from parallel hypothesis tests using a variety of meta-analysis methods. Each group of p-values is defined from the corresponding entries across all vectors. The function processes all vectors "in parallel" - hence the name.

Usage

```
combineParallelPValues(
  p.values,
  method = c("simes", "holm-min", "berger", "fisher", "pearson", "wilkinson",
    "stouffer"),
  weights = NULL,
  log.p = FALSE,
  min.n = 1,
  min.prop = 0.5
)
```

Arguments

`p.values`	A list of numeric vectors of the same length, containing the p-values to be combined.
`method`	String specifying the method to use to combine p-values.
`weights`	A numeric vector of positive weights, with one value per vector in `...`. Each weight is applied to all entries of itscorresponding vector, i.e., all p-values in that vector receive the same weight. Alternatively, a list of numeric vectors of weights with the same structure as `p.values`. Each p-value is then assigned the weight in the corresponding entry of `weights`. Alternatively `NULL`, in which case all p-values are assigned equal weight.
`log.p`	Logical scalar indicating whether the p-values in `p.values` are log-transformed.
`min.n`	Integer scalar specifying the minimum number of individual nulls to reject when testing the joint null.
`min.prop`	Numeric scalar in [0, 1], specifying the minimum proportion of individual nulls to reject when testing the joint null.

Details

`min.prop` and `min.n` only have an effect for `method="wilkinson"` and `"holm-min"`.

`weights` only has an effect for `method="simes"`, `"holm-min"` and `"stouffer"`.

Value

A list containing:

- `p.value`, a numeric vector of length equal to the length of each vector in `p.values`. This contains the Simes p-value for each group, log-transformed if `log.p=TRUE`.
- `representative`, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of `p.values` containing the representative test.
- `influential`, a list of logical vectors mirroring the structure of `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final combined p-value.

Author(s)

Aaron Lun

Examples

```
p1 <- runif(10000)
p2 <- runif(10000)
p3 <- runif(10000)

fish <- combineParallelPValues(list(p1, p2, p3), method="fisher")
hist(fish$p.value)

z <- combineParallelPValues(list(p1, p2, p3), method="stouffer", weights=1:3)
hist(z$p.value)

simes <- combineParallelPValues(list(p1, p2, p3), method="simes")
hist(simes$p.value)

berger <- combineParallelPValues(list(p1, p2, p3), method="berger")
hist(berger$p.value)
```

---

countGroupedDirection *Count directions of grouped tests*

Description

Count the number of grouped tests that are significant and changing in each direction. Each group of tests is defined according to a grouping factor.

Usage

```
countGroupedDirection(
  p.values,
  grouping,
  effects,
  p.threshold = 0.05,
  effect.threshold = 0,
  method = c("BH", "holm"),
  log.p = FALSE
)
```

Arguments

p.values	A numeric vector containing p-values for individual tests.
grouping	A vector or factor of length equal to p.values, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of p.values belonging to that group. This assumes that p.values is ordered such that all entries in the same group are adjacent to each other.
effects	A numeric vector of length equal to p.values, containing the effect size for each test.
p.threshold	Numeric scalar defining the adjusted p-value threshold at which test is significant.
effect.threshold	Numeric scalar defining the threshold at which an effect is “"up"” or “"down"”.
method	String specifying how the multiple testing correction within each group is to be performed.
log.p	Logical scalar indicating whether the p.values are log-transformed.

Details

We apply a multiple testing correction within each group and define all significant tests as those with adjusted p-values below p.threshold. We then count the number of tests with effects greater than effect.threshold (i.e., going up) or less than the threshold (i.e., going down). This allows us to quantify the direction of change across the group into two counts.

The output of this function is simpler to interpret than the summarize*Direction functions. However, p.threshold has no clear relationship to the significance threshold applied to the combined p-values. Groups with many low p-values will have milder multiplicity corrections, resulting in more non-zero counts for both the up and down tests; this often complicates matters by introducing mixed directions of effect, even when the most significant changes in the group are in one direction.

Note that, if log.p=TRUE, the function will automatically handle the log-transformation of p.threshold.

Value

A list with the "up" and "down" entries. Both are integer vectors of length equal to the number of groups, containing the number of significant tests changing each each direction in each group.

Author(s)

Aaron Lun

See Also

summarizeGroupedDirection, for another way to summarize the overall direction into a single string.

countParallelDirection, for the equivalent function operating on parallel tests.

Examples

```
p <- rbeta(100, 0.5, 1)
eff <- rnorm(100)
g <- sample(20, 100, replace=TRUE)

(dirs <- countGroupedDirection(p, g, eff))
```

---

countParallelDirection

Count directions of parallel tests

Description

Count the number of parallel tests that are significant and changing in each direction. Each group of tests is defined as corresponding entries across vectors.

Usage

```
countParallelDirection(
  p.values,
  effects,
  p.threshold = 0.05,
  effect.threshold = 0,
  method = c("BH", "holm"),
  log.p = FALSE
)
```

Arguments

p.values	A list of numeric vectors of the same length, containing the p-values to be combined.
effects	A list of numeric vectors of same structure as p.values, containing the effect sizes for each set of tests.
p.threshold	Numeric scalar defining the adjusted p-value threshold at which test is significant.

`effect.threshold`
: Numeric scalar defining the threshold at which an effect is “"up"” or “"down"”.

`method`
: String specifying how the multiple testing correction within each group is to be performed.

`log.p`
: Logical scalar indicating whether the `p.values` are log-transformed.

Details

We apply a multiple testing correction within each group and define all significant tests as those with adjusted p-values below `p.threshold`. We then count the number of tests with effects greater than `effect.threshold` (i.e., going up) or less than the threshold (i.e., going down). This allows us to quantify the direction of change across the group into two counts.

The output of this function is simpler to interpret than the `summarize*Direction` functions. However, `p.threshold` has no clear relationship to the significance threshold applied to the combined p-values. Groups with many low p-values will have milder multiplicity corrections, resulting in more non-zero counts for both the `up` and `down` tests; this often complicates matters by introducing mixed directions of effect, even when the most significant changes in the group are in one direction.

Note that, if `log.p=TRUE`, the function will automatically handle the log-transformation of `p.threshold`.

Value

A list with the `"up"` and `"down"` entries. Both are integer vectors of length equal to the number of groups, containing the number of significant tests changing each each direction in each group.

Author(s)

Aaron Lun

See Also

`summarizeParallelDirection`, for another way to summarize the overall direction into a single string.

`countGroupedDirection`, for the equivalent function based on a grouping factor.

Examples

```
p1 <- rbeta(100, 0.5, 1)
eff1 <- rnorm(100)
p2 <- rbeta(100, 0.5, 1)
eff2 <- rnorm(100)

(dirs <- countParallelDirection(list(p1, p2), list(eff1, eff2)))
```

groupedBerger *Combine grouped p-values with Berger's IUT*

Description

Combine p-values from grouped tests with Berger's intersection-union test (IUT). Groups are defined according to unique levels of a grouping factor.

Usage

```
groupedBerger(p.values, grouping, log.p = FALSE)
```

Arguments

p.values	A numeric vector containing p-values for individual tests.
grouping	A vector or factor of length equal to p.values, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of p.values belonging to that group. This assumes that p.values is ordered such that all entries in the same group are adjacent to each other.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.

Details

The joint null hypothesis for each group is that *any* of the individual null hypotheses are true. Berger's IUT will only reject the joint null if all of the individual nulls are rejected. This method is applicable under arbitrary dependency structures. No weights are considered.

The representative test for each group is defined as the test with the largest p-value, as this is ultimately used as the IUT p-value. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

Value

A list containing:

- p.value, a named numeric vector of length equal to the number of unique levels in grouping. This contains the IUT p-value for each group, log-transformed if log.p=TRUE. Each entry is named according to the group.
- representative, a named integer scalar specifying the representative test for each group. Each index refers to an entry of p.values and is named according to its group.
- influential, a logical vector of length equal to p.values. Entries are TRUE for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

References

Berger RL and Hsu JC (1996). Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Statist. Sci.* 11, 283-319.

See Also

parallelBerger, for a version that operates on parallel vectors of p-values.

Examples

```
p1 <- rbeta(100, 0.8, 1)
g <- sample(10, length(p1), replace=TRUE)

# Standard application:
out <- groupedBerger(p1, g)
str(out)

# With log p-values.
out <- groupedBerger(log(p1), g, log.p=TRUE)
str(out)
```

---

groupedFisher *Combine grouped p-values with Fisher's method*

Description

Combine p-values from grouped tests with Fisher's method. Groups are defined according to unique levels of a grouping factor.

Usage

```
groupedFisher(p.values, grouping, log.p = FALSE)
```

Arguments

p.values	A numeric vector containing p-values for individual tests.
grouping	A vector or factor of length equal to p.values, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of p.values belonging to that group. This assumes that p.values is ordered such that all entries in the same group are adjacent to each other.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.

Details

The joint null hypothesis for each group is that all of the individual null hypotheses are true. Fisher's method combines information from all individual nulls to determine if the joint null should be rejected. Compared to Stouffer's and Pearson's methods, Fisher's method provides more sensitivity to the smallest individual p-value. This method is only applicable to independent tests and no weights are considered.

The representative test for each group is defined as the test with the lowest p-value, as this has the greatest effect on the combined p-value. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

Value

A list containing:

- `p.value`, a named numeric vector of length equal to the number of unique levels in `grouping`. This contains the Fisher combined p-value for each group, log-transformed if `log.p=TRUE`. Each entry is named according to the group.
- `representative`, a named integer scalar specifying the representative test for each group. Each index refers to an entry of `p.values` and is named according to its group.
- `influential`, a logical vector of length equal to `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

References

Fisher RA (1925). *Statistical Methods for Research Workers*. Oliver and Boyd (Edinburgh).

See Also

`parallelFisher`, for a version that operates on parallel vectors of p-values.

`groupedStouffer` and `groupedPearson`, for different approaches to testing a joint null of independent hypotheses.

Examples

```
p1 <- rbeta(100, 0.8, 1)
g <- sample(10, length(p1), replace=TRUE)

# Standard application:
out <- groupedFisher(p1, g)
str(out)

# With log p-values.
out <- groupedFisher(log(p1), g, log.p=TRUE)
str(out)
```

`groupedHolmMin` *Combine grouped p-values with the minimum Holm approach*

Description

Combine p-values from grouped tests with the minimum Holm approach. Groups are defined according to unique levels of a grouping factor.

Usage

```
groupedHolmMin(
  p.values,
  grouping,
  weights = NULL,
  log.p = FALSE,
  min.n = 1,
  min.prop = 0.5
)
```

Arguments

`p.values`	A numeric vector containing p-values for individual tests.
`grouping`	A vector or factor of length equal to `p.values`, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of `p.values` belonging to that group. This assumes that `p.values` is ordered such that all entries in the same group are adjacent to each other.
`weights`	A numeric vector of length equal to `p.values`, containing a positive weight for each test. Alternatively `NULL`, in which case equal weights are assigned to all tests.
`log.p`	Logical scalar indicating whether the p-values in `p.values` are log-transformed.
`min.n`	Integer scalar specifying the minimum number of individual nulls to reject when testing the joint null.
`min.prop`	Numeric scalar in [0, 1], specifying the minimum proportion of individual nulls to reject when testing the joint null.

Details

Here, the joint null hypothesis for each group is that fewer than N individual null hypotheses are false. The joint null is only rejected if N or more individual nulls are rejected; hence the "minimum" in the function name.

N is defined as the larger of `min.n` and the product of `min.prop` with the number of tests in the group (rounded up). This allows users to scale rejection of the joint null with the size of the group, while avoiding a too-low N when the group is small. Note that N is always capped at the total size of the group.

To compute the combined p-value, we apply the Holm-Bonferroni correction to all p-values in the set and take the N-th smallest value. This effectively recapitulates the step-down procedure where we reject individual nulls until we reach the N-th test. This method works correctly in the presence of dependencies between p-values.

If non-equal weights are provided, they are used to effectively downscale the p-values. This aims to redistribute the error rate across the individual tests, i.e., tests with lower weights are given fewer opportunities to drive acceptance of the joint null.

The representative test for each group is defined as that with the N-th smallest p-value, as this is directly used as the combined p-value. The influential tests for each group are defined as those with p-values no greater than the representative test's p-value. This is based on the fact that increasing them (e.g., by setting them to unity) would result in a larger combined p-value.

Value

A list containing:

- `p.value`, a named numeric vector of length equal to the number of unique levels in `grouping`. This contains the minimum Holm p-value for each group, log-transformed if `log.p=TRUE`. Each entry is named according to the group.
- `representative`, a named integer scalar specifying the representative test for each group. Each index refers to an entry of `p.values` and is named according to its group.
- `influential`, a logical vector of length equal to `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

References

Holm S (1979). A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* 6, 65-70.

See Also

`parallelHolmMin`, for a version that operates on parallel vectors of p-values.

`groupedWilkinson`, for a more relaxed version of this test when hypotheses are independent.

Examples

```
p1 <- rbeta(100, 0.8, 1)
g <- sample(10, length(p1), replace=TRUE)

# Standard application:
out <- groupedHolmMin(p1, g)
str(out)

# With weights:
out <- groupedHolmMin(p1, g, weights=rexp(length(p1)))
str(out)
```

```
# With log p-values.
out <- groupedHolmMin(log(p1), g, log.p=TRUE)
str(out)
```

`groupedPearson` *Combine grouped p-values with Pearson's method*

Description

Combine p-values from grouped tests with Pearson's method. Groups are defined according to unique levels of a grouping factor.

Usage

```
groupedPearson(p.values, grouping, log.p = FALSE)
```

Arguments

`p.values`	A numeric vector containing p-values for individual tests.
`grouping`	A vector or factor of length equal to `p.values`, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of `p.values` belonging to that group. This assumes that `p.values` is ordered such that all entries in the same group are adjacent to each other.
`log.p`	Logical scalar indicating whether the p-values in `p.values` are log-transformed.

Details

Here, the joint null hypothesis for each group is that all of the individual null hypotheses are true. Pearson's method combines information from all individual nulls to determine if the joint null should be rejected. Compared to Stouffer's and Pearson's methods, Pearson's method is more sensitive to the largest individual p-value. This method is only applicable to independent tests and no weights are considered.

The representative test for each group is defined as the test with the largest p-value, as this has the greatest effect on the combined p-value. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

Value

A list containing:

- `p.value`, a named numeric vector of length equal to the number of unique levels in `grouping`. This contains the Pearson combined p-value for each group, log-transformed if `log.p=TRUE`. Each entry is named according to the group.

- `representative`, a named integer scalar specifying the representative test for each group. Each index refers to an entry of `p.values` and is named according to its group.
- `influential`, a logical vector of length equal to `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

References

Pearson K (1934). On a new method of deternining "goodness of fit." *Biometrika* 26, 425-442.

See Also

`parallelPearson`, for a version that operates on parallel vectors of p-values.

`groupedFisher` and `groupedStouffer`, for different approaches to testing a joint null of independent hypotheses.

Examples

```
p1 <- rbeta(100, 0.8, 1)
g <- sample(10, length(p1), replace=TRUE)

# Standard application:
out <- groupedPearson(p1, g)
str(out)

# With log p-values.
out <- groupedPearson(log(p1), g, log.p=TRUE)
str(out)
```

---

groupedSimes *Combine grouped p-values with Simes' method*

Description

Combine p-values from grouped tests with Simes' method. Groups are defined according to unique levels of a grouping factor.

Usage

```
groupedSimes(p.values, grouping, weights = NULL, log.p = FALSE)
```

Arguments

p.values	A numeric vector containing p-values for individual tests.
grouping	A vector or factor of length equal to p.values, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of p.values belonging to that group. This assumes that p.values is ordered such that all entries in the same group are adjacent to each other.
weights	A numeric vector of length equal to p.values, containing a positive weight for each test. Alternatively NULL, in which case equal weights are assigned to all tests.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.

Details

The joint null hypothesis for each group is that all of the individual null hypotheses are true. Simes' method will reject the joint null if any of the individual nulls are rejected, providing weak control of the family-wise error rate.

In theory, the method is only applicable to independent tests, but experience suggests that it is quite robust to dependencies. The calculation itself is very closely related to the Benjamini-Hochberg method for controlling the false discovery rate. One can actually obtain Simes' combined p-value by taking the smallest BH-adjusted p-value across a group.

If non-equal weights are provided, they are treated as relative frequency weights. That is, if one p-value is given a weight of 10 and another p-value is given a weight of 1, the former is considered to occur 10 times more frequently than the latter.

The representative test for each group is defined as the test with the p-value that is ultimately used as the combined p-value. Briefly, one can identify this test as that with the smallest BH-adjusted p-value if the monotonicity adjustment were omitted. The influential tests for each group are defined as those with p-values no greater than the representative test's p-value. This is based on the fact that increasing them (e.g., by setting them to unity) would result in a larger combined p-value.

Value

A list containing:

- p.value, a named numeric vector of length equal to the number of unique levels in grouping. This contains the combined Simes p-value for each group, log-transformed if log.p=TRUE. Each entry is named according to the group.
- representative, a named integer scalar specifying the index of representative test for each group. Each index refers to an entry of p.values and is named according to its group.
- influential, a logical vector of length equal to p.values. Entries are TRUE for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

References

Simes RJ (1986). An improved Bonferroni procedure for multiple tests of significance. *Biometrika* 73:751-754.

Sarkar SK and Chung CK (1997). The Simes method for multiple hypothesis testing with positively dependent test statistics. *J. Am. Stat. Assoc.* 92, 1601-1608.

Benjamini Y and Hochberg Y (1997). Multiple hypotheses testing with weights. *Scand. J. Stat.* 24, 407-418.

See Also

parallelSimes, for a version that operates on parallel vectors of p-values.

Examples

```
p1 <- rbeta(100, 0.8, 1)
g <- sample(10, length(p1), replace=TRUE)

# Standard application:
out <- groupedSimes(p1, g)
str(out)

# With weights:
out <- groupedSimes(p1, g, weights=rexp(length(p1)))
str(out)

# With log p-values.
out <- groupedSimes(log(p1), g, log.p=TRUE)
str(out)
```

---

groupedStouffer *Combine grouped p-values with Stouffer's Z-score method*

Description

Combine p-values from grouped tests with Stouffer's Z-score method. Groups are defined according to unique levels of a grouping factor.

Usage

```
groupedStouffer(p.values, grouping, weights = NULL, log.p = FALSE)
```

Arguments

p.values	A numeric vector containing p-values for individual tests.

grouping	A vector or factor of length equal to p.values, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of p.values belonging to that group. This assumes that p.values is ordered such that all entries in the same group are adjacent to each other.
weights	A numeric vector of length equal to p.values, containing a positive weight for each test. Alternatively NULL, in which case equal weights are assigned to all tests.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.

Details

The joint null hypothesis for each group is that all of the individual null hypotheses are true. Stouffer's method combines information from all individual nulls to determine if the joint null should be rejected. This serves as a compromise between Fisher's method (sensitive to the smallest p-value) and Pearson's method (sensitive to the largest p-value).

Stouffer's method is only applicable for independent tests. Weights are supported by scaling the contribution of each individual null to the summed Z-score. In this manner, more highly weighted tests will have a greater effect on the final combined p-value.

The representative test for each group is defined as the test with the most negative weighted Z-score, as this has the greatest effect on the combined p-value when the joint null is rejected. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

When a group contains both zero and unity p-values, we compare the sum of weights for all zero p-values and all unity p-values. If the former is larger, the combined p-value is set to zero; if the latter, to unity. If they are equal, we pretend that the two sets of p-values "cancel out" and contribute nothing to the summed Z-score. This is not entirely rigorous but provides reasonable output in the presence of such boundary values.

Value

A list containing:

- p.value, a named numeric vector of length equal to the number of unique levels in grouping. This contains the combined p-value for each group, log-transformed if log.p=TRUE. Each entry is named according to the group.
- representative, a named integer scalar specifying the representative test for each group. Each entry is named according to the group.
- influential, a logical vector of length equal to p.values. Entries are TRUE for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

References

Stouffer SA et al. (1949). *The American Soldier, Vol. 1 - Adjustment during Army Life*. Princeton University Press (Princeton).

Whitlock MC (2005). Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach. *J. Evol. Biol.* 18, 5:1368-73.

See Also

`parallelStouffer`, for a version that operates on parallel vectors of p-values.

`groupedFisher` and `groupedPearson`, for different approaches to testing a joint null of independent hypotheses.

Examples

```
p1 <- rbeta(100, 0.8, 1)
g <- sample(10, length(p1), replace=TRUE)

# Standard application:
out <- groupedStouffer(p1, g)
str(out)

# With weights:
out <- groupedStouffer(p1, g, weights=rexp(length(p1)))
str(out)

# With log p-values.
out <- groupedStouffer(log(p1), g, log.p=TRUE)
str(out)
```

`groupedWilkinson` *Combine grouped p-values with Wilkinson's method*

Description

Combine p-values from grouped tests with Wilkinson's method. Groups are defined according to unique levels of a grouping factor.

Usage

```
groupedWilkinson(p.values, grouping, log.p = FALSE, min.n = 1, min.prop = 0.5)
```

Arguments

`p.values`	A numeric vector containing p-values for individual tests.

grouping	A vector or factor of length equal to p.values, specifying the group to which each test is assigned. Alternatively, an rle object where each run corresponds to a group and specifies the entries of p.values belonging to that group. This assumes that p.values is ordered such that all entries in the same group are adjacent to each other.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.
min.n	Integer scalar specifying the minimum number of individual nulls to reject when testing the joint null.
min.prop	Numeric scalar in [0, 1], specifying the minimum proportion of individual nulls to reject when testing the joint null.

Details

Here, the joint null hypothesis for each group is all individual null hypotheses are false. Rejection of the joint null is heavily favored in situations where N or more individual nulls are rejected. This is achieved in Wilkinson's method by considering the N-th order statistic for uniformly distributed p-values. The individual tests are assumed to be independent, and all weights are ignored.

N is defined as the larger of min.n and the product of min.prop with the number of tests in the group (rounded up). This allows users to scale rejection of the joint null with the size of the group, while avoiding a too-low N when the group is small. Note that N is always capped at the total size of the group.

The representative test for each group is defined as that with the N-th smallest p-value, as this is used to compute the combined p-value. The influential tests for each group are defined as those with p-values no greater than the representative test's p-value. This is based on the fact that increasing them (e.g., by setting them to unity) would result in a larger combined p-value.

Value

A list containing:

- p.value, a named numeric vector of length equal to the number of unique levels in grouping. This contains the combined p-value for each group, log-transformed if log.p=TRUE. Each entry is named according to the group.
- representative, a named integer scalar specifying the representative test for each group. Each index refers to an entry of p.values and is named according to its group.
- influential, a logical vector of length equal to p.values. Entries are TRUE for any p-value that is deemed "influential" to the final combined p-value for its group.

Author(s)

Aaron Lun

References

Wilkinson B (1951). A statistical consideration in psychological research. *Psychol. Bull.* 48, 156-158.

See Also

parallelWilkinson, for a version that operates on parallel vectors of p-values.

groupedHolmMin, which has a more strict interpretation of *N*, amongst other things.

Examples

```
p1 <- rbeta(100, 0.8, 1)
g <- sample(10, length(p1), replace=TRUE)

# Standard application:
out <- groupedWilkinson(p1, g)
str(out)

# With log p-values.
out <- groupedWilkinson(log(p1), g, log.p=TRUE)
str(out)
```

---

parallelBerger *Combine parallel p-values with Berger's IUT*

Description

Combine p-values from parallel tests with Berger's intersection-union test (IUT). Each group of p-values is defined from the corresponding entries across all vectors.

Usage

```
parallelBerger(p.values, log.p = FALSE)
```

Arguments

p.values	A list of numeric vectors of the same length, containing the p-values to be combined.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.

Details

The joint null hypothesis for each group is that *any* of the individual null hypotheses are true. Berger's IUT will only reject the joint null if all of the individual nulls are rejected. This method is applicable under arbitrary dependency structures. No weights are considered.

The representative test for each group is defined as the test with the largest p-value, as this is ultimately used as the IUT p-value. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

Value

A list containing:

- `p.value`, a numeric vector of length equal to the length of each vector in `p.values`. This contains the IUT p-value for each group, log-transformed if `log.p=TRUE`.
- `representative`, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of `p.values` containing the representative test.
- `influential`, a list of logical vectors mirroring the structure of `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final combined p-value.

Author(s)

Aaron Lun

References

Berger RL and Hsu JC (1996). Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Statist. Sci.* 11, 283-319.

See Also

`groupedBerger`, for the version that combines p-values based on a grouping factor.

Examples

```
p1 <- rbeta(100, 0.8, 1)
p2 <- runif(100)
p3 <- rbeta(100, 0.5, 1)

# Standard application:
out <- parallelBerger(list(p1, p2, p3))
str(out)

# With log p-values.
out <- parallelBerger(list(log(p1), log(p2), log(p3)), log.p=TRUE)
str(out)
```

---

parallelFisher *Combine parallel p-values with Fisher's method*

Description

Combine p-values from parallel tests with Fisher's method. Each group of p-values is defined from the corresponding entries across all vectors.

Usage

```
parallelFisher(p.values, log.p = FALSE)
```

Arguments

p.values	A list of numeric vectors of the same length, containing the p-values to be combined.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.

Details

The joint null hypothesis for each group is that all of the individual null hypotheses are true. Fisher's method combines information from all individual nulls to determine if the joint null should be rejected. Compared to Stouffer's and Pearson's methods, Fisher's method provides more sensitivity to the smallest individual p-value. This method is only applicable to independent tests and no weights are considered.

The representative test for each group is defined as the test with the lowest p-value, as this has the greatest effect on the combined p-value. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

Value

A list containing:

- p.value, a numeric vector of length equal to the length of each vector in p.values. This contains the combined Fisher p-value for each group, log-transformed if log.p=TRUE.
- representative, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of p.values containing the representative test.
- influential, a list of logical vectors mirroring the structure of p.values. Entries are TRUE for any p-value that is deemed "influential" to the final combined p-value.

Author(s)

Aaron Lun

References

Fisher RA (1925). *Statistical Methods for Research Workers*. Oliver and Boyd (Edinburgh).

See Also

groupedFisher, for a version that combines p-values based on a grouping factor.

parallelStouffer and parallelPearson, for different approaches to testing a joint null of independent hypotheses.

Examples

```
p1 <- rbeta(100, 0.8, 1)
p2 <- runif(100)
p3 <- rbeta(100, 0.5, 1)

# Standard application:
```

```
out <- parallelFisher(list(p1, p2, p3))
str(out)

# With log p-values.
out <- parallelFisher(list(log(p1), log(p2), log(p3)), log.p=TRUE)
str(out)
```

`parallelHolmMin` *Combine parallel p-values with the minimum Holm approach*

Description

Combine p-values from parallel tests with the minimum Holm approach. Each group of p-values is defined from the corresponding entries across all vectors.

Usage

```
parallelHolmMin(
  p.values,
  weights = NULL,
  log.p = FALSE,
  min.n = 1,
  min.prop = 0.5
)
```

Arguments

`p.values`	A list of numeric vectors of the same length, containing the p-values to be combined.
`weights`	A numeric vector of positive weights, with one value per vector in `...`. Each weight is applied to all entries of itscorresponding vector, i.e., all p-values in that vector receive the same weight. Alternatively, a list of numeric vectors of weights with the same structure as `p.values`. Each p-value is then assigned the weight in the corresponding entry of `weights`. Alternatively `NULL`, in which case all p-values are assigned equal weight.
`log.p`	Logical scalar indicating whether the p-values in `p.values` are log-transformed.
`min.n`	Integer scalar specifying the minimum number of individual nulls to reject when testing the joint null.
`min.prop`	Numeric scalar in [0, 1], specifying the minimum proportion of individual nulls to reject when testing the joint null.

Details

Here, the joint null hypothesis for each group is that fewer than N individual null hypotheses are false. The joint null is only rejected if N or more individual nulls are rejected; hence the "minimum" in the function name.

N is defined as the larger of `min.n` and the product of `min.prop` with the number of tests in the group (rounded up). This allows users to scale rejection of the joint null with the size of the group, while avoiding a too-low N when the group is small. Note that N is always capped at the total size of the group.

To compute the combined p-value, we apply the Holm-Bonferroni correction to all p-values in the set and take the N-th smallest value. This effectively recapitulates the step-down procedure where we reject individual nulls until we reach the N-th test. This method works correctly in the presence of dependencies between p-values.

If non-equal weights are provided, they are used to effectively downscale the p-values. This aims to redistribute the error rate across the individual tests, i.e., tests with lower weights are given fewer opportunities to drive acceptance of the joint null.

The representative test for each group is defined as that with the N-th smallest p-value, as this is directly used as the combined p-value. The influential tests for each group are defined as those with p-values no greater than the representative test's p-value. This is based on the fact that increasing them (e.g., by setting them to unity) would result in a larger combined p-value.

Value

A list containing:

- `p.value`, a numeric vector of length equal to the length of each vector in `p.values`. This contains the combined p-value for each group, log-transformed if `log.p=TRUE`.
- `representative`, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of `p.values` containing the representative test.
- `influential`, a list of logical vectors mirroring the structure of `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final per-group p-value.

Author(s)

Aaron Lun

References

Holm S (1979). A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* 6, 65-70.

See Also

`groupedHolmMin`, for a version that combines p-values based on a grouping factor.

`parallelWilkinson`, for a more relaxed version of this test when hypotheses are independent.

Examples

```
p1 <- rbeta(100, 0.8, 1)
p2 <- runif(100)
p3 <- rbeta(100, 0.5, 1)

# Standard application:
out <- parallelHolmMin(list(p1, p2, p3))
str(out)

# With vector-level weights:
out <- parallelHolmMin(list(p1, p2, p3), weights=c(10, 20, 30))
str(out)

# With log p-values.
out <- parallelHolmMin(list(log(p1), log(p2), log(p3)), log.p=TRUE)
str(out)
```

`parallelPearson` *Combine parallel p-values with Pearson's method*

Description

Combine p-values from parallel tests with Pearson's method. Each group of p-values is defined from the corresponding entries across all vectors.

Usage

```
parallelPearson(p.values, log.p = FALSE)
```

Arguments

`p.values`	A list of numeric vectors of the same length, containing the p-values to be combined.
`log.p`	Logical scalar indicating whether the p-values in `p.values` are log-transformed.

Details

Here, the joint null hypothesis for each group is that all of the individual null hypotheses are true. Pearson's method combines information from all individual nulls to determine if the joint null should be rejected. Compared to Stouffer's and Pearson's methods, Pearson's method is more sensitive to the largest individual p-value. This method is only applicable to independent tests and no weights are considered.

The representative test for each group is defined as the test with the largest p-value, as this has the greatest effect on the combined p-value. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

Value

A list containing:

- `p.value`, a numeric vector of length equal to the length of each vector in `p.values`. This contains the combined Pearson p-value for each group, log-transformed if `log.p=TRUE`.
- `representative`, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of `p.values` containing the representative test.
- `influential`, a list of logical vectors mirroring the structure of `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final combined p-value.

Author(s)

Aaron Lun

References

Pearson K (1934). On a new method of deternining "goodness of fit." *Biometrika* 26, 425-442.

See Also

`groupedPearson`, for a version that combines p-values based on a grouping factor.

`parallelFisher` and `parallelStouffer`, for different approaches to testing a joint null of independent hypotheses.

Examples

```
p1 <- rbeta(100, 0.8, 1)
p2 <- runif(100)
p3 <- rbeta(100, 0.5, 1)

# Standard application:
out <- parallelPearson(list(p1, p2, p3))
str(out)

# With log p-values.
out <- parallelPearson(list(log(p1), log(p2), log(p3)), log.p=TRUE)
str(out)
```

---

parallelSimes *Combine parallel p-values with Simes' method*

Description

Combine p-values from parallel tests with Simes' method. Each group of p-values is defined from the corresponding entries across all vectors.

Usage

```
parallelSimes(p.values, weights = NULL, log.p = FALSE)
```

Arguments

`p.values`	A list of numeric vectors of the same length, containing the p-values to be combined.
`weights`	A numeric vector of positive weights, with one value per vector in `...`. Each weight is applied to all entries of itscorresponding vector, i.e., all p-values in that vector receive the same weight. Alternatively, a list of numeric vectors of weights with the same structure as `p.values`. Each p-value is then assigned the weight in the corresponding entry of `weights`. Alternatively `NULL`, in which case all p-values are assigned equal weight.
`log.p`	Logical scalar indicating whether the p-values in `p.values` are log-transformed.

Details

The joint null hypothesis for each group is that all of the individual null hypotheses are true. Simes' method will reject the joint null if any of the individual nulls are rejected, providing weak control of the family-wise error rate.

In theory, the method is only applicable to independent tests, but experience suggests that it is quite robust to dependencies. The calculation itself is very closely related to the Benjamini-Hochberg method for controlling the false discovery rate. One can actually obtain Simes' combined p-value by taking the smallest BH-adjusted p-value across a group.

If non-equal weights are provided, they are treated as relative frequency weights. That is, if one p-value is given a weight of 10 and another p-value is given a weight of 1, the former is considered to occur 10 times more frequently than the latter.

The representative test for each group is defined as the test with the p-value that is ultimately used as the combined p-value. Briefly, one can identify this test as that with the smallest BH-adjusted p-value if the monotonicity adjustment were omitted. The influential tests for each group are defined as those with p-values no greater than the representative test's p-value. This is based on the fact that increasing them (e.g., by setting them to unity) would result in a larger combined p-value.

Value

A list containing:

- `p.value`, a numeric vector of length equal to the length of each vector in `p.values`. This contains the combined Simes p-value for each group, log-transformed if `log.p=TRUE`.
- `representative`, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of `p.values` containing the representative test.
- `influential`, a list of logical vectors mirroring the structure of `p.values`. Entries are `TRUE` for any p-value that is deemed "influential" to the final combined p-value.

Author(s)

Aaron Lun

References

Simes RJ (1986). An improved Bonferroni procedure for multiple tests of significance. *Biometrika* 73:751-754.

Sarkar SK and Chung CK (1997). The Simes method for multiple hypothesis testing with positively dependent test statistics. *J. Am. Stat. Assoc.* 92, 1601-1608.

Benjamini Y and Hochberg Y (1997). Multiple hypotheses testing with weights. *Scand. J. Stat.* 24, 407-418.

See Also

groupedSimes, for a version that combines p-values based on a grouping factor.

Examples

```
p1 <- rbeta(100, 0.8, 1)
p2 <- runif(100)
p3 <- rbeta(100, 0.5, 1)

# Standard application:
out <- parallelSimes(list(p1, p2, p3))
str(out)

# With vector-level weights:
out <- parallelSimes(list(p1, p2, p3), weights=c(10, 20, 30))
str(out)

# With log p-values.
out <- parallelSimes(list(log(p1), log(p2), log(p3)), log.p=TRUE)
str(out)
```

---

parallelStouffer *Combine parallel p-values with Stouffer's Z-score*

Description

Combine p-values from parallel tests with Stouffer's Z-score method. Each group of p-values is defined from the corresponding entries across all vectors.

Usage

```
parallelStouffer(p.values, weights = NULL, log.p = FALSE)
```

Arguments

p.values	A list of numeric vectors of the same length, containing the p-values to be combined.
weights	A numeric vector of positive weights, with one value per vector in .... Each weight is applied to all entries of itscorresponding vector, i.e., all p-values in that vector receive the same weight. Alternatively, a list of numeric vectors of weights with the same structure as p.values. Each p-value is then assigned the weight in the corresponding entry of weights. Alternatively NULL, in which case all p-values are assigned equal weight.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.

Details

The joint null hypothesis for each group is that all of the individual null hypotheses are true. Stouffer's method combines information from all individual nulls to determine if the joint null should be rejected. This serves as a compromise between Fisher's method (sensitive to the smallest p-value) and Pearson's method (sensitive to the largest p-value).

Stouffer's method is only applicable for independent tests. Weights are supported by scaling the contribution of each individual null to the summed Z-score. In this manner, more highly weighted tests will have a greater effect on the final combined p-value.

The representative test for each group is defined as the test with the most negative weighted Z-score, as this has the greatest effect on the combined p-value when the joint null is rejected. All tests for each group are considered to be influential as increasing any of them (e.g., to unity) would result in a larger combined p-value.

When a group contains both zero and unity p-values, we compare the sum of weights for all zero p-values and all unity p-values. If the former is larger, the combined p-value is set to zero; if the latter, to unity. If they are equal, we pretend that the two sets of p-values "cancel out" and contribute nothing to the summed Z-score. This is not entirely rigorous but provides reasonable output in the presence of such boundary values.

Value

A list containing:

- p.value, a numeric vector of length equal to the length of each vector in p.values. This contains the combined Stouffer p-value for each group, log-transformed if log.p=TRUE.
- representative, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of p.values containing the representative test.
- influential, a list of logical vectors mirroring the structure of p.values. Entries are TRUE for any p-value that is deemed "influential" to the final combined p-value.

Author(s)

Aaron Lun

References

Stouffer SA et al. (1949). *The American Soldier, Vol. 1 - Adjustment during Army Life*. Princeton University Press (Princeton).

Whitlock MC (2005). Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach. *J. Evol. Biol.* 18, 5:1368-73.

See Also

`groupedStouffer`, for a version that combines p-values based on a grouping factor.

`parallelFisher` and `parallelPearson`, for different approaches to testing a joint null of independent hypotheses.

Examples

```
p1 <- rbeta(100, 0.8, 1)
p2 <- runif(100)
p3 <- rbeta(100, 0.5, 1)

# Standard application:
out <- parallelStouffer(list(p1, p2, p3))
str(out)

# With weights:
out <- parallelStouffer(list(p1, p2, p3), weights=5:7)
str(out)

# With log p-values.
out <- parallelStouffer(list(log(p1), log(p2), log(p3)), log.p=TRUE)
str(out)
```

---

parallelWilkinson *Combine p-values with Wilkinson's method*

Description

Combine p-values from parallel tests with Wilkinson's method. Each group of p-values is defined from the corresponding entries across all vectors. The function processes all vectors "in parallel" - hence the name.

Usage

```
parallelWilkinson(p.values, log.p = FALSE, min.n = 1, min.prop = 0.5)
```

Arguments

p.values	A list of numeric vectors of the same length, containing the p-values to be combined.
log.p	Logical scalar indicating whether the p-values in p.values are log-transformed.
min.n	Integer scalar specifying the minimum number of individual nulls to reject when testing the joint null.
min.prop	Numeric scalar in [0, 1], specifying the minimum proportion of individual nulls to reject when testing the joint null.

Details

Here, the joint null hypothesis for each group is all individual null hypotheses are false. Rejection of the joint null is heavily favored in situations where N or more individual nulls are rejected. This is achieved in Wilkinson's method by considering the N-th order statistic for uniformly distributed p-values. The individual tests are assumed to be independent, and all weights are ignored.

N is defined as the larger of min.n and the product of min.prop with the number of tests in the group (rounded up). This allows users to scale rejection of the joint null with the size of the group, while avoiding a too-low N when the group is small. Note that N is always capped at the total size of the group.

The representative test for each group is defined as that with the N-th smallest p-value, as this is used to compute the combined p-value. The influential tests for each group are defined as those with p-values no greater than the representative test's p-value. This is based on the fact that increasing them (e.g., by setting them to unity) would result in a larger combined p-value.

Value

A list containing:

- p.value, a numeric vector of length equal to the length of each vector in p.values. This contains the combined p-value for each group, log-transformed if log.p=TRUE.
- representative, an integer scalar specifying the representative test in each group. Specifically, this refers to the index of the *vector* of p.values containing the representative test.
- influential, a list of logical vectors mirroring the structure of p.values. Entries are TRUE for any p-value that is deemed "influential" to the final per-group p-value.

Author(s)

Aaron Lun

References

Wilkinson B (1951). A statistical consideration in psychological research. *Psychol. Bull.* 48, 156-158.

See Also

groupedWilkinson, for a version that combines p-values based on a grouping factor.

parallelHolmMin, which has a more strict interpretation of *N*, amongst other things.

Examples

```
p1 <- rbeta(100, 0.8, 1)
p2 <- runif(100)
p3 <- rbeta(100, 0.5, 1)

# Standard application:
out <- parallelWilkinson(list(p1, p2, p3))
str(out)

# With log p-values.
out <- parallelWilkinson(list(log(p1), log(p2), log(p3)), log.p=TRUE)
str(out)
```

`summarizeGroupedDirection` *Summarize overall direction of grouped tests*

Description

Summarize the overall direction of grouped tests in a meta-analysis, based on the influential tests defined by one of the `grouped*` functions.

Usage

```
summarizeGroupedDirection(effects, grouping, influential, threshold = 0)
```

Arguments

`effects`	A numeric vector containing the effect size for each test.
`grouping`	A vector of factor of length equal to `effects`, specifying the assigned group for each tests. Alternatively, an rle object where each run corresponds to a group and specifies the entries of `effects` belonging to that group. This assumes that `effects` is ordered such that all entries in the same group are adjacent to each other.
`influential`	A logical vector of length equal to `effects`, indicating whether each test is influential in its assigned group.
`threshold`	Numeric scalar defining the threshold at which an effect is “"up"” or “"down"”.

Details

We focus on the direction of effect for the influential tests that actually contribute to a group's final p-value. For example, if we did our meta-analysis using `parallelSimes`,we are not particularly concerned about the direction of tests with large p-values. Thus, we just ignore them when summarizing the group's direction in this function. Otherwise, we would unnecessarily obtain a `mixed` direction of effect if a test with a large p-value had a weakly opposing effect.

Of course, the interpretation of "influential" really depends on the choice of meta-analysis strategy. It is also possible that this function reports a single direction when the group really is mixed, e.g., if the tests with the lowest p-values are changing in one direction but tests with weaker but still interesting effects are changing in the other direction. The extent to which this is of interest is left to the discretion of the user.

Value

A character vector of length equal to the number of groups. Each entry can be:

- `"up"`, if all influential tests have effects above `threshold`.
- `"down"`, if all influential tests have effects below `threshold`.
- `"none"`, if all influential tests have effects equal to `threshold`.
- `"mixed"`, if there are influential tests with effects above and below `threshold`.

Author(s)

Aaron Lun

See Also

`groupedSimes` and related `grouped*` functions, to obtain `influential`.

`summarizeParallelDirection`, for the equivalent function based on parallel tests.

`countGroupedDirection`, to count the number of effects in each direction.

Examples

```
p <- rbeta(100, 0.5, 1)
eff <- rnorm(100)
g <- sample(20, 100, replace=TRUE)

out <- groupedSimes(p, g)
(dir <- summarizeGroupedDirection(eff, g, out$influential))
```

---

summarizeParallelDirection

Summarize overall direction of parallel tests

Description

Summarize the overall direction of parallel tests in a meta-analysis, based on the influential tests defined by one of the `parallel*` functions.

Usage

```
summarizeParallelDirection(effects, influential, threshold = 0)
```

Arguments

effects	A list of numeric vectors of the same length, containing the effect sizes for each set of tests. Each group of tests is defined as corresponding entries across vectors.
influential	A list of logical vectors with the same structure as effects, specifying the tests that are influential to the final per-group p-value.
threshold	Numeric scalar defining the threshold at which an effect is “"up"” or “"down"”.

Details

We focus on the direction of effect for the influential tests that actually contribute to a group's final p-value. For example, if we did our meta-analysis using parallelSimes,we are not particularly concerned about the direction of tests with large p-values. Thus, we just ignore them when summarizing the group's direction in this function. Otherwise, we would unnecessarily obtain a mixed direction of effect if a test with a large p-value had a weakly opposing effect.

Of course, the interpretation of “influential” really depends on the choice of meta-analysis strategy. It is also possible that this function reports a single direction when the group really is mixed, e.g., if the tests with the lowest p-values are changing in one direction but tests with weaker but still interesting effects are changing in the other direction. The extent to which this is of interest is left to the discretion of the user.

Value

A character vector of length equal to the number of groups. Each entry can be:

- "up", if all influential tests have effects above threshold.
- "down", if all influential tests have effects below threshold.
- "none", if all influential tests have effects equal to threshold.
- "mixed", if there are influential tests with effects above and below threshold.

Author(s)

Aaron Lun

See Also

parallelSimes and related parallel* functions, to obtain influential.

summarizeGroupedDirection, for the equivalent function based on a grouping factor.

countParallelDirection, to count the number of effects in each direction.

Examples

```
p1 <- rbeta(100, 0.5, 1)
eff1 <- rnorm(100)
p2 <- rbeta(100, 0.5, 1)
eff2 <- rnorm(100)
```

```
out <- parallelSimes(list(p1, p2))
(dir <- summarizeParallelDirection(list(eff1, eff2), out$influential))
```

Index

averageGroupedStats, 2
averageParallelStats, *3*, 3, *4*

combineGroupedPValues, 4
combineParallelPValues, 6
countGroupedDirection, 7, *10*, *36*
countParallelDirection, *9*, 9, *37*

groupedBerger, 11, *24*
groupedFisher, 12, *17*, *21*, *25*
groupedHolmMin, 14, *23*, *27*
groupedPearson, *13*, 16, *21*, *29*
groupedSimes, 17, *31*, *36*
groupedStouffer, *13*, *17*, 19, *33*
groupedWilkinson, *15*, 21, *34*

parallelBerger, *12*, 23
parallelFisher, *13*, 24, *29*, *33*
parallelHolmMin, *15*, 26, *34*
parallelPearson, *17*, *25*, 28, *33*
parallelSimes, *19*, 29, *35*, *37*
parallelStouffer, *21*, *25*, *29*, 31
parallelWilkinson, *23*, *27*, 33

rle, *2*, *5*, *8*, *11, 12*, *14*, *16*, *18*, *20*, *22*, *35*

summarizeGroupedDirection, *9*, 35, *37*
summarizeParallelDirection, *10*, *36*, 36