Package ‘mMPA’

July 22, 2025

Type Package

Title Implementation of Marker-Assisted Mini-Pooling with Algorithm

Version 1.2.0

Author
``Tao Liu, PhD <`tliu@stat.brown.edu`> [aut, cre]'' ``Yizhen Xu, ScM <`yizhen_xu@brown.edu`> [aut]''

Maintainer ``Tao Liu, PhD'' <`tliu@stat.brown.edu`>

Description To determine the number of quantitative assays needed for a sample of data using pooled testing methods, which include mini-pooling (MP), MP with algorithm (MPA), and marker-assisted MPA (mMPA). To estimate the number of assays needed, the package also provides a tool to conduct Monte Carlo (MC) to simulate different orders in which the sample would be collected to form pools. Using MC avoids the dependence of the estimated number of assays on any specific ordering of the samples to form pools.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0.9000

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-10-12 10:50:02 UTC

Contents

`minipool` *Number of Assays Needed using Mini-Pooling*

Description

Function `minipool(...)` calculates the number of assays required, when using mini-pooling, for pools that are formed following the order that individual samples appear in the data.

Usage

```
minipool(v, K = 5, vf_cut = 1000, lod = 0, msg = T)
```

Arguments

`v`	A vector of non-negative numerical assay results.
`K`	Pool size; default is `K = 5`.
`vf_cut`	Cutoff value for defining positive cases; default is `vf_cut = 1000`.
`lod`	A vector of lower limits of detection or a scalar if the limits are the same; default is `lod = 0`.
`msg`	Message generated during calculation; default is `TRUE`.

Details

Suppose that N samples are collected for pooled testing. The first `K` samples are combined to form a pool, the next `K` samples are combined to form the second pool, and so on. If the number of samples for the last pool is less than `K`, these remaining samples are not used to form a pool (i.e. not included in the calculation). Therefore, a total of `N%/%K` pools are formed. The function calculates the number of assays needed for each of these pools. For mini-pooling, if a pool is negative, no further tests are needed and all samples in the pool are concluded as being negative; so the total number of assays required is one. Otherwise if the pool is tested positive, all individual samples in the pool are tested and the total number of assays required is `(K + 1)`.

Value

A vectorof length `N%/%K` for the numbers of assays needed for all pools that are formed.

References

Dorfman R. The detection of defective members of large populations. The Annals of Mathematical Statistics. 1943;14(4):436-440.

Liu T, Hogan JW, Daniels, MJ, Coetzer M, Xu Y, Bove G, et al. Improved HIV-1 Viral Load Monitoring Capacity Using Pooled Testing with Marker-Assisted Deconvolution. Journal of AIDS. 2017;75(5): 580-587.

See Also

mpa, mmpa, pooling_mc

Examples

```
K=5; n = 50;
n.pool  = n/K; n.pool
#  [1] 10
set.seed(100)
pvl = rgamma(n, shape = 2.8, scale = 150)
minipool(pvl)
#  A total of 10 pools are formed.
#  The numbers of assays required by these pools are:
#  [1] 6 6 6 6 6 6 6 6 6 6
```

`mmpa` *Number of Assays Required using Marker-Assisted Mini-Pooling with Algorithm (mMPA)*

Description

Function `mmpa(...)` calculates the number of assays required, when using mMPA, for pools that are formed following the order of individual samples in the data.

Usage

```
mmpa(v, s, K = 5, vf_cut = 1000, lod = 0, msg = T)
```

Arguments

`v`	A vector of non-negative numerical assay results.
`s`	A vector of risk scores; `s` must have the same length as `v`. The risk score `s` needs to be positively associated with `v`; othewise an error message will be generated.
`K`	Pool size; default is `K = 5`.
`vf_cut`	Cutoff value for defining positive cases; default is `vf_cut = 1000`.
`lod`	A vector of lower limits of detection or a scalar if the limits are the same; default is `lod = 0`.
`msg`	Message generated during calculation; default is `TRUE`.

Details

For a given sample (v_i, s_i), i = 1, ..., N, the first `K` samples are combined to form a pool, the next `K` samples are combined to form the second pool, and so on. If the number of samples for the last pool is less than `K`, these remaining samples are not used to form a pool (i.e. not included in the calculation) . Therefore, a total of `N%/%K` pools are formed. The function calculates the number of assays needed for each of these pools.

Value

A vectorof length `N%/%K` for the numbers of assays needed for all pools that are formed.

References

Liu T, Hogan JW, Daniels, MJ, Coetzer M, Xu Y, Bove G, et al. Improved HIV-1 Viral Load Monitoring Capacity Using Pooled Testing with Marker-Assisted Deconvolution. Journal of AIDS. 2017;75(5): 580-587.

Bilder CR, Tebbs JM, Chen P. Informative retesting. Journal of the American Statistical Association. 2010;105(491):942-955.

May S, Gamst A, Haubrich R, Benson C, Smith DM. Pooled nucleic acid testing to identify antiretroviral treatment failure during HIV infection. Journal of Acquired Immune Deficiency Syndromes. 2010;53(2):194-201.

See Also

minipool, mpa, pooling_mc

Examples

```
K=5; n = 50;
n.pool  = n/K; n.pool
#   [1] 10
set.seed(100)
pvl = rgamma(n, shape = 2.8, scale = 150)
riskscore = (rank(pvl)/n) * 0.5 + runif(n) * 0.5
mmpa(v = pvl, s = riskscore)
#   A total of 10 pools are formed.
#   The numbers of assays required by these pools are:
#   [1] 3 3 4 4 2 3 3 4 3 3
```

`mpa` *Number of Assays Needed using Mini-Pooling with Algorithm (MPA)*

Description

Function `mpa(...)` calculates the number of assays required, when using MPA, for pools that are formed following the order of individual samples in the data.

Usage

```
mpa(v, K = 5, vf_cut = 1000, lod = 0, msg = T)
```

Arguments

`v`	A vector of non-negative numerical assay results.
`K`	Pool size; default is `K = 5`.
`vf_cut`	Cutoff value for defining positive cases; default is `vf_cut = 1000`.
`lod`	A vector of lower limits of detection or a scalar if the limits are the same; default is `lod = 0`.
`msg`	Message generated during calculation; default is `TRUE`.

Details

For a given sample v_i, i = 1, ..., N, the first `K` samples v_1, ..., v_5 are combined to form a pool, the next `K` samples v_6, ..., v_10 are combined to form the second pool, and so on. If the number of samples for the last pool is less than `K`, these remaining samples are not used to form a pool (i.e. not included in the calculation) . Therefore, a total of `N%/%K` pools are formed. The function calculates the number of assays needed for each of these pools. See May et al (2010).

Value

A vectorof length `N%/%K` for the numbers of assays needed for all pools that are formed.

References

May, S., Gamst, A., Haubrich, R., Benson, C., & Smith, D. M. (2010). Pooled nucleic acid testing to identify antiretroviral treatment failure during HIV infection. Journal of acquired immune deficiency syndromes (1999), 53(2), 194.

Liu T, Hogan JW, Daniels, MJ, Coetzer M, Xu Y, Bove G, et al. Improved HIV-1 Viral Load Monitoring Capacity Using Pooled Testing with Marker-Assisted Deconvolution. Journal of AIDS. 2017;75(5): 580-587.

See Also

minipool, mmpa, pooling_mc

Examples

```
K=5; n = 50;
n.pool  = n/K; n.pool
#   [1] 10
set.seed(100)
pvl = rgamma(n, shape = 2.8, scale = 150)
mpa(v = pvl)
#  A total of 10 pools are formed.
#   The numbers of assays required by these pools are:
#   [1] 3 3 4 4 2 5 4 4 4 4
```

`pooling_mc` *Monte Carlo Simulation for Estimating the Number of Assays Required when Using Pooled Testing*

Description

This function uses Monte Carlo (MC) to simulate different orders in which the samples would be collected to form pools. Unlike the function `minipool`, `mpa`, and `mmpa` that calculate the number of assays needed for pools that are formed following the exact order of the samples that are listed in the data, this function `pooling_mc` permutes the data many (`perm_num`) times so as to estimate the average number of assays required (ATR) per individual. Using MC avoids the dependence on any specific ordering of forming pools.

Usage

```
pooling_mc(v, s = NULL, K = 5, vf_cut = 1000, lod = 0,
  method = "mmpa", perm_num = 100, msg = F)
```

Arguments

`v`	A vector of non-negative numerical assay results.
`s`	A vector of risk scores; `s` must be provided if `method = "mmpa"` and have the same length as `v`. The risk score `s` needs to be positively associated with `v`; othewise an error message will be generated.
`K`	Pool size; default is `K = 5`.
`vf_cut`	Cutoff value for defining positive cases; default is `vf_cut = 1000`.
`lod`	A vector of lower limits of detection or a scalar if the limits are the same; default is `lod = 0`.
`method`	Method that is used for pooled testing; must be one of `minipool`, `mpa`, and `mmpa`. By default, `method = "mmpa"`.
`perm_num`	The number of permutation to be used for the calculation; default is `100`.
`msg`	Message generated during calculation; default is `FALSE`.

Value

The outcome is a matrix of dimension `num_pool` by `perm_num`. The row number is the number of pools (`num_pool`) from each permutation of the data, which is determined by the sample size `N` and pool size `K`; `num_pool = N%/%K`. The column number is the number of permutations (`num_pool`).

References

Liu T, Hogan JW, Daniels, MJ, Coetzer M, Xu Y, Bove G, et al. Improved HIV-1 Viral Load Monitoring Capacity Using Pooled Testing with Marker-Assisted Deconvolution. Journal of AIDS. 2017;75(5): 580-587.

Bilder CR, Tebbs JM, Chen P. Informative retesting. Journal of the American Statistical Association. 2010;105(491):942-955.

May S, Gamst A, Haubrich R, Benson C, Smith DM. Pooled nucleic acid testing to identify antiretroviral treatment failure during HIV infection. Journal of Acquired Immune Deficiency Syndromes. 2010;53(2):194-201.

Dorfman R. The detection of defective members of large populations. The Annals of Mathematical Statistics. 1943;14(4):436-440.

See Also

minipool, mpa, mmpa

Examples

```
### sample size = 300
n = 300;
set.seed(100)
pvl = rgamma(n, shape = 2.8, scale = 150)
summary(pvl)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#   53      225      392     424    564    1373
riskscore = (rank(pvl)/n) * 0.5 + runif(n) * 0.5
cor(pvl, riskscore, method = "spearman")
#   [1] 0.69
### Pool size K is set to 5
K=5;
### so, the number of pools = 60
n.pool  = n/K; n.pool
#   [1] 60
foo = pooling_mc(pvl, riskscore, perm_num = 100)
### Average number of assays needed per pool for each of the 100
### permutations of the data
apply(foo, 2, mean)
#   [1] 3.43 3.33 3.35 3.47 3.37 3.33 3.37 3.27 3.43 3.28 3.32 3.35 3.35 3.37
#   [15] 3.38 3.37 3.30 3.43 3.28 3.38 3.42 3.35 3.35 3.48 3.30 3.47 3.40 3.35
#  [29] 3.25 3.30 3.38 3.43 3.25 3.45 3.35 3.33 3.42 3.38 3.40 3.33 3.32 3.38
#   [43] 3.33 3.37 3.37 3.33 3.35 3.38 3.38 3.30 3.30 3.33 3.37 3.32 3.30 3.40
#   [57] 3.37 3.42 3.30 3.37 3.38 3.32 3.45 3.38 3.37 3.50 3.33 3.40 3.28 3.37
#   [71] 3.23 3.33 3.23 3.42 3.32 3.32 3.45 3.35 3.32 3.32 3.33 3.33 3.30 3.38
#   [85] 3.37 3.33 3.33 3.20 3.37 3.33 3.30 3.40 3.40 3.32 3.33 3.37 3.40 3.38
#   [99] 3.30 3.33
### Estimated average number of assays needed per pool
mean(foo)
#   3.35
### Estimated average number of assays needed per individual
mean(foo)/K
#   [1] 0.67
```

Index