Package ‘mbmixture’

November 28, 2024

Version 0.6

Date 2024-11-27

Title Microbiome Mixture Analysis

Description Evaluate whether a microbiome sample is a mixture of two samples, by fitting a model for the number of read counts as a function of single nucleotide polymorphism (SNP) allele and the genotypes of two potential source samples.
Lobo et al. (2021) <doi:10.1093/g3journal/jkab308>.

Author Karl W Broman [aut, cre] (<`https://orcid.org/0000-0002-4914-6671`>)

Maintainer Karl W Broman <`broman@wisc.edu`>

Depends R (>= 3.1.0)

Imports stats, parallel, numDeriv

Suggests knitr, rmarkdown, testthat, devtools, roxygen2

License MIT + file LICENSE

URL `https://github.com/kbroman/mbmixture`

BugReports `https://github.com/kbroman/mbmixture/issues`

VignetteBuilder knitr

LazyData true

Encoding UTF-8

ByteCompile true

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-11-27 23:20:02 UTC

Contents

`bootstrapNull` *Bootstrap to assess significance*

Description

Perform a parametric bootstrap to assess whether there is significant evidence that a sample is a mixture.

Usage

```
bootstrapNull(
  tab,
  n_rep = 1000,
  interval = c(0, 1),
  tol = 0.000001,
  check_boundary = TRUE,
  cores = 1,
  return_raw = TRUE
)
```

Arguments

`tab` Dataset of read counts as 3d array of size 3x3x2, genotype in first sample x genotype in second sample x allele in read.

`n_rep` Number of bootstrap replicates

`interval` Interval to which each parameter should be constrained

`tol` Tolerance for convergence

`check_boundary` If TRUE, explicitly check the boundaries of `interval`.

`cores` Number of CPU cores to use, for parallel calculations. (If `0`, use `parallel::detectCores()`.) Alternatively, this can be links to a set of cluster sockets, as produced by `parallel::makeCluster()`.

`return_raw` If TRUE, return the raw results. If FALSE, just return the p-value. Unlink `bootstrapSE()`, here the default is TRUE.

Value

If `return_raw=FALSE`, a single numeric value (the p-value).If `return_raw=TRUE`, a vector of length `n_rep` with the LRT statistics from each bootstrap replicate.

See Also

`bootstrapSE()`

Examples

```
data(mbmixdata)
# just 100 bootstrap replicates, as an illustration
bootstrapNull(mbmixdata, n_rep=100)
```

`bootstrapSE` *Bootstrap to get standard errors*

Description

Perform a parametric bootstrap to get estimated standard errors.

Usage

```
bootstrapSE(
  tab,
  n_rep = 1000,
  interval = c(0, 1),
  tol = 0.000001,
  check_boundary = FALSE,
  cores = 1,
  return_raw = FALSE
)
```

Arguments

`tab` Dataset of read counts as 3d array of size 3x3x2, genotype in first sample x genotype in second sample x allele in read.

`n_rep` Number of bootstrap replicates

`interval` Interval to which each parameter should be constrained

`tol` Tolerance for convergence

`check_boundary` If TRUE, explicitly check the boundaries of `interval`.

`cores` Number of CPU cores to use, for parallel calculations. (If `0`, use `parallel::detectCores()`.) Alternatively, this can be links to a set of cluster sockets, as produced by `parallel::makeCluster()`.

`return_raw` If TRUE, return the raw results. If FALSE, just return the estimated standard errors.

Value

If `return_raw=FALSE`, a vector of two standard errors. If `return_raw=TRUE`, an matrix of size `n_rep` x 2 with the detailed bootstrap results.

See Also

`bootstrapNull()`

Examples

```
data(mbmixdata)
# just 100 bootstrap replicates, as an illustration
bootstrapSE(mbmixdata, n_rep=100)
```

mbmixdata *Example dataset for mbmixture package*

Description

Example dataset for mbmixture package.

Usage

```
data(mbmixdata)
```

Format

Dataset of read counts as 3d array of size 3x3x2, genotype in first sample x genotype in second sample x allele in read.

Examples

```
data(mbmixdata)
mle_pe(mbmixdata)
```

`mbmix_loglik` *log likelihood function for microbiome mixture*

Description

Calculate log likelihood function for microbiome sample mixture model at particular values of `p` and `e`.

Usage

```
mbmix_loglik(tab, p, e = 0)
```

Arguments

`tab`	Dataset of read counts as 3d array of size 3x3x2, genotype in first sample x genotype in second sample x allele in read.
`p`	Contaminant probability (proportion of mixture coming from the second sample).
`e`	Sequencing error rate.

Value

The log likelihood evaluated at `p` and `e`.

Examples

```
data(mbmixdata)
mbmix_loglik(mbmixdata, p=0.74, e=0.002)
```

`mle_e` *MLE of e for fixed p*

Description

Calculate the MLE of the sequencing error rate e for a fixed value of the contaminant probability p.

Usage

```
mle_e(
  tab,
  p = 0.05,
  interval = c(0, 1),
  tol = 0.000001,
  check_boundary = FALSE
)
```

Arguments

`tab` Dataset of read counts as 3d array of size 3x3x2, genotype in first sample x genotype in second sample x allele in read.

`p` Assumed value for the contaminant probability

`interval` Interval to which each parameter should be constrained

`tol` Tolerance for convergence

`check_boundary` If TRUE, explicitly check the boundaries of `interval`.

Value

A single numeric value, the MLE of `e`, with the log likelihood as an attribute.

Examples

```
data(mbmixdata)
mle_e(mbmixdata, p=0.74)
```

`mle_p` *MLE of p for fixed e*

Description

Calculate the MLE of the contaminant probability p for a fixed value of the sequencing error rate e.

Usage

```
mle_p(
  tab,
  e = 0.002,
  interval = c(0, 1),
  tol = 0.000001,
  check_boundary = FALSE
)
```

Arguments

`tab` Dataset of read counts as 3d array of size 3x3x2, genotype in first sample x genotype in second sample x allele in read.

`e` Assumed value for the sequencing error rate

`interval` Interval to which each parameter should be constrained

`tol` Tolerance for convergence

`check_boundary` If TRUE, explicitly check the boundaries of `interval`.

Value

A single numeric value, the MLE of p, with the log likelihood as an attribute.

Examples

```
data(mbmixdata)
mle_p(mbmixdata, e=0.002)
```

---

mle_pe *Find MLEs for microbiome mixture*

Description

Find joint MLEs of p and e for microbiome mixture model

Usage

```
mle_pe(
  tab,
  interval = c(0, 1),
  tol = 0.000001,
  check_boundary = FALSE,
  SE = FALSE
)
```

Arguments

tab	Dataset of read counts as 3d array of size 3x3x2, genotype in first sample x genotype in second sample x allele in read.
interval	Interval to which each parameter should be constrained
tol	Tolerance for convergence
check_boundary	If TRUE, explicitly check the boundaries of interval.
SE	If TRUE, get estimated standard errors.

Value

A vector containing the estimates of p and e along with the evaluated log likelihood and likelihood ratio test statistics for the hypotheses p=0 and p=1.

Examples

```
data(mbmixdata)
mle_pe(mbmixdata)
```

Index