Package ‘semisup’

October 17, 2020

Version 1.12.4

Title Semi-Supervised Mixture Model

Description Implements a parametric semi-supervised mixture model. The permutation test detects markers with main or interactive effects, without distinguishing them. Possible applications include genome-wide association analysis and differential expression analysis.

biocViews SNP, GenomicVariation, SomaticMutation, Genetics, Classification, Clustering, DNASeq, Microarray, MultipleComparison

Depends R (>= 3.0.0)

Imports VGAM

Suggests knitr, testthat, SummarizedExperiment

VignetteBuilder knitr

License GPL-3

LazyData true

RoxygenNote 7.0.0

URL `https://github.com/rauschenberger/semisup`

BugReports `https://github.com/rauschenberger/semisup/issues`

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

git_branch RELEASE_3_11

git_last_commit 659af71

git_last_commit_date 2020-05-08

Date/Publication 2020-10-16

Author Armin Rauschenberger [aut, cre]

Maintainer Armin Rauschenberger <`armin.rauschenberger@uni.lu`>

R topics documented:

`semisup-package` *Semi-supervised mixture model*

Description

This R package implements the semi-supervised mixture model. Use `mixtura` for model fitting, and `scrutor` for hypothesis testing.

Getting started

Please type the following commands:
```
utils::vignette("semisup")
?semisup::mixtura
?semisup::scrutor
```

More information

A Rauschenberger, RX Menezes, MA van de Wiel, NM van Schoor, and MA Jonker (2017). "Detecting SNPs with interactive effects on a quantitative trait", *Manuscript in preparation*.

`<a.rauschenberger@vumc.nl>`

`mixtura` *Model fitting*

Description

This function fits a semi-supervised mixture model. It simultaneously estimates two mixture components, and assigns the unlabelled observations to these.

Usage

```
mixtura(y, z, dist = "norm",
        phi = NULL, pi = NULL, gamma = NULL,
        test = NULL, iter = 100, kind = 0.05,
        debug = TRUE, ...)
```

Arguments

`y`	**observations:** numeric vector of length `n`
`z`	**class labels:** integer vector of length `n`, with entries `0`, `1` and `NA`
`dist`	distributional assumption**:** character `"norm"` (Gaussian), `"nbinom"` (negative bionomial), or `"zinb"` (zero-inflated negative binomial)
`phi`	dispersion parameters**:** numeric vector of length `q`, or `NULL`
`pi`	zero-inflation parameter(s)**:** numeric vector of length `q`, or `NULL`
`gamma`	offset**:** numeric vector of length `n`, or `NULL`
`test`	resampling procedure**:** character `"perm"` (permutation) or `"boot"` (parametric bootstrap), or `NULL`

`iter`	(maximum) number of resampling iterations : positive integer, or `NULL`
`kind`	resampling accuracy: numeric between `0` and `1`, or `NULL`; all p-values above `kind` are approximate
`debug`	verification of arguments: `TRUE` or `FALSE`
`...`	settings `EM` algorithm: `starts`, `it.em` and `epsilon` (see `arguments`)

Details

By default, `phi` and `pi` are estimated by the maximum likelihood method, and `gamma` is replaced by a vector of ones.

Value

This function fits and compares a one-component (`H0`) and a two-component (`H1`) mixture model.

`posterior`	probability of belonging to class 1: numeric vector of length `n`
`converge`	path of the log-likelihood: numeric vector with maximum length `it.em`
`estim0`	parameter estimates under `H0`: data frame
`estim1`	parameter estimates under `H1`: data frame
`loglik0`	log-likelihood under `H0`: numeric
`loglik1`	log-likelihood under `H1`: numeric
`lrts`	likelihood-ratio test statistic: positive numeric
`p.value`	`H0` versus `H1`: numeric between `0` and `1`, or `NULL`

Reference

A Rauschenberger, RX Menezes, MA van de Wiel, NM van Schoor, and MA Jonker (2017). "Detecting SNPs with interactive effects on a quantitative trait", *Manuscript in preparation*.

See Also

Use `scrutor` for hypothesis testing. All other functions are `internal`.

Examples

```
# data simulation
n <- 100
z <- rep(0:1,each=n/2)
y <- rnorm(n=n,mean=2,sd=1)
z[(n/4):n] <- NA

# model fitting
mixtura(y,z,dist="norm",test="perm")
```

scrutor — *Hypothesis testing*

Description

This function tests whether the unlabelled observations come from a mixture of two distributions.

Usage

```
scrutor(Y, Z, dist = "norm",
        phi = NULL, pi = NULL, gamma = NULL,
        test = "perm", iter = NULL, kind = NULL,
        debug = TRUE, ...)
```

Arguments

`Y`	**observations:** numeric vector of length `n`, or numeric matrix with `n` rows (samples) and `q` columns (variables)
`Z`	**class labels:** numeric vector of length `n`, or numeric matrix with `n` rows (samples) and `p` columns (variables), with entries `0` and `NA`
`dist`	distributional assumption**:** character `"norm"` (Gaussian), `"nbinom"` (negative bionomial), or `"zinb"` (zero-inflated negative binomial)
`phi`	dispersion parameter(s)**:** numeric vector of length `q`, or `NULL` (`norm`: none, `nbinom`: MLE)
`pi`	zero-inflation parameter(s)**:** numeric vector of length `q`, or `NULL` (`norm`: none,`nbinom`: MLE)
`gamma`	offset**:** numeric vector of length `n`, or `NULL`
`test`	resampling procedure**:** character `"perm"` (permutation) or `"boot"` (parametric bootstrap), or `NULL`
`iter`	(maximum) number of resampling iterations **:** positive integer, or `NULL`
`kind`	resampling accuracy**:** numeric between `0` and `1`, or `NULL`**;** all p-values above `kind` are approximate
`debug`	verification of arguments**:** `TRUE` or `FALSE`
`...`	settings `EM` algorithm**:** `starts`, `it.em` and `epsilon` (see `arguments`)

Details

By default, `phi` and `pi` are estimated by the maximum likelihood method, and `gamma` is replaced by a vector of ones.

Value

This function tests a one-component (`H0`) against a two-component mixture model (`H1`).

`y`	index observations
`z`	index class labels
`lrts`	test statistic
`p.value`	p-value

Reference

A Rauschenberger, RX Menezes, MA van de Wiel, NM van Schoor, and MA Jonker (2017). "Detecting SNPs with interactive effects on a quantitative trait", *Manuscript in preparation*.

See Also

Use `mixtura` for model fitting. All other functions are `internal`.

Examples

```
# data simulation
n <- 100
z <- rep(0:1,each=n/2)
y <- rnorm(n=n,mean=2*z,sd=1)
z[(n/4):n] <- NA

# hypothesis testing
scrutor(y,z,dist="norm")
```

Index