

# Package ‘semisup’

October 12, 2022

**Version** 1.21.0

**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** master

**git\_last\_commit** 6de5e1b

**git\_last\_commit\_date** 2022-04-26

**Date/Publication** 2022-10-12

**Author** Armin Rauschenberger [aut, cre]

**Maintainer** Armin Rauschenberger <[armin.rauschenberger@uni.lu](mailto:armin.rauschenberger@uni.lu)>

## R topics documented:

semisup-package . . . . .	2
mixtura . . . . .	2
scrutor . . . . .	4

---

semisup-package	<i>Semi-supervised mixture model</i>
-----------------	--------------------------------------

---

### 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 (2020). "Semi-supervised mixture test for detecting markers associated with a quantitative trait", *Manuscript in preparation*.

<a.rauschenberger@vumc.nl>

---

mixtura	<i>Model fitting</i>
---------	----------------------

---

### 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	<b>observations:</b> numeric vector of length n
z	<b>class labels:</b> integer vector of length n, with entries 0, 1 and NA
dist	<b>distributional assumption:</b> character "norm" (Gaussian), "nbinom" (negative binomial), or "zinb" (zero-inflated negative binomial)
phi	<b>dispersion parameters:</b> numeric vector of length q, or NULL
pi	<b>zero-inflation parameter(s):</b> numeric vector of length q, or NULL
gamma	<b>offset:</b> numeric vector of length n, or NULL
test	<b>resampling procedure:</b> character "perm" (permutation) or "boot" (parametric bootstrap), or NULL
iter	<b>(maximum) number of resampling iterations :</b> positive integer, or NULL
kind	<b>resampling accuracy:</b> numeric between 0 and 1, or NULL; all p-values above kind are approximate
debug	<b>verification of arguments:</b> TRUE or FALSE
...	<b>settings EM algorithm:</b> starts, it.em and epsilon (see <a href="#">arguments</a> )

**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 ( $H_0$ ) and a two-component ( $H_1$ ) mixture model.

posterior	<b>probability of belonging to class 1:</b> numeric vector of length n
converge	<b>path of the log-likelihood:</b> numeric vector with maximum length it.em
estim0	<b>parameter estimates under <math>H_0</math>:</b> data frame
estim1	<b>parameter estimates under <math>H_1</math>:</b> data frame
loglik0	<b>log-likelihood under <math>H_0</math>:</b> numeric
loglik1	<b>log-likelihood under <math>H_1</math>:</b> numeric
lrts	<b>likelihood-ratio test statistic:</b> positive numeric
p.value	<b><math>H_0</math> versus <math>H_1</math>:</b> numeric between 0 and 1, or NULL

**Reference**

A Rauschenberger, RX Menezes, MA van de Wiel, NM van Schoor, and MA Jonker (2020). "Semi-supervised mixture test for detecting markers associated with 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	<b>observations:</b> numeric vector of length n, or numeric matrix with n rows (samples) and q columns (variables)
Z	<b>class labels:</b> numeric vector of length n, or numeric matrix with n rows (samples) and p columns (variables), with entries 0 and NA
dist	<b>distributional assumption:</b> character "norm" (Gaussian), "nbinom" (negative binomial), or "zinb" (zero-inflated negative binomial)
phi	<b>dispersion parameter(s):</b> numeric vector of length q, or NULL (norm: none, nbinom: MLE)
pi	<b>zero-inflation parameter(s):</b> numeric vector of length q, or NULL (norm: none, nbinom: MLE)
gamma	<b>offset:</b> numeric vector of length n, or NULL
test	<b>resampling procedure:</b> character "perm" (permutation) or "boot" (parametric bootstrap), or NULL
iter	<b>(maximum) number of resampling iterations :</b> positive integer, or NULL
kind	<b>resampling accuracy:</b> numeric between 0 and 1, or NULL; all p-values above kind are approximate
debug	<b>verification of arguments:</b> TRUE or FALSE
...	<b>settings EM algorithm:</b> starts, it.em and epsilon (see <a href="#">arguments</a> )

## 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 ( $H_0$ ) against a two-component mixture model ( $H_1$ ).

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 (2020). "Semi-supervised mixture test for detecting markers associated with 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

\* **documentation**

semisup-package, [2](#)

\* **methods**

mixtura, [2](#)

scrutor, [4](#)

arguments, [3](#), [4](#)

internal, [3](#), [5](#)

mixtura, [2](#), [2](#), [5](#)

scrutor, [2](#), [3](#), [4](#)

semisup (semisup-package), [2](#)

semisup-package, [2](#)