Package ‘semisup’

April 26, 2017

Version 1.0.0

Title Detecting SNPs with interactive effects on a quantitative trait

Description This R package moves away from testing interaction terms, and move towards testing whether an individual SNP is involved in any interaction. This reduces the multiple testing burden to one test per SNP, and allows for interactions with unobserved factors. Analysing one SNP at a time, it splits the individuals into two groups, based on the number of minor alleles. If the quantitative trait differs in mean between the two groups, the SNP has a main effect. If the quantitative trait differs in distribution between some individuals in one group and all other individuals, it possibly has an interactive effect. Implicitly, the membership probabilities may suggest potential interacting variables.

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

Depends R (>= 3.0.0)

Imports SummarizedExperiment, VGAM

Suggests knitr, testthat

VignetteBuilder knitr

License GPL-3

LazyData true

RoxygenNote 6.0.1

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

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

NeedsCompilation no

Author Armin Rauschenberger [aut, cre]

Maintainer Armin Rauschenberger <a.rauschenberger@vumc.nl>

R topics documented:

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

Index 6
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:

```r
utils::vignette("semisup")
?semisup::mixtura
?semisup::scrutor
```

More information


<mixtura>

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

```r
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 binomial), or "zinp" (zero-inflated negative binomial)
- `phi` dispersion parameter: positive numeric, or NULL
- `pi` zero-inflation parameter: numeric between 0 and 1, or NULL
- `gamma` offset: numeric vector of length `n`, or NULL
- `test` resampling procedure: character "perm" (permutation) or "boot" (parametric bootstrap), or NULL
mixtura

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

See Also
Use scrutinor 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")
Hypothesis testing

Description

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

Usage

```r
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 binomial), or "zinf" (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

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

Examples
```r
# 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

*Topic documentation
  semisup-package, 2

*Topic 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