

semisup: detecting SNPs with interactive effects on a quantitative trait

A Rauschenberger, RX Menezes, MA van de Wiel,
NM van Schoor, and MA Jonker

April 20, 2026

This vignette explains how to use the R package **semisup**. Use the function [mixtura](#) for model fitting, and the function [scrutor](#) for hypothesis testing.

1 Initialisation

Start with installing **semisup** from Bioconductor¹:

```
if (!requireNamespace("BiocManager", quietly=TRUE))
  install.packages("BiocManager")
BiocManager::install("semisup")
```

Then load and attach the package:

```
library(semisup)
```

If you want to reproduce the examples, you should attach the toy database:

```
attach(toydata)
```

The following commands access the reference manual:

```
help(semisup)
?semisup
```

¹[devtools](#) and [GitHub](#): `devtools::install_github("rauschenberger/semisup")`

2 Scope

Data is available for n samples. Let $\mathbf{y} = (y_1, \dots, y_n)^T$ represent the observations, $\mathbf{x} = (x_1, \dots, x_n)^T$ the groups, and $\mathbf{z} = (z_1, \dots, z_n)^T$ the classes. We assume all observations from the *labelled* group are in class **A**, and those from the *unlabelled* group are in class **A** or in class **B**.

	1	...	s	$s+1$...	$n = s + u$
\mathbf{y}	y_1	...	y_s	y_{s+1}	...	y_{s+u}
\mathbf{x}	0	...	0	1	...	1
\mathbf{z}	A	...	A	A/B	...	A/B

Table 1: Observations \mathbf{y} , groups \mathbf{x} , and classes \mathbf{z} . Here, the first s observations are *labelled* (class **A**), and the last u observations are *unlabelled* (class **A** or **B**).

We assume all observations come from the same probability distribution, but with different parameters for the two classes:

$$Y_i | (Z_i = \mathbf{A}) \sim F(\cdot, \boldsymbol{\theta}_a),$$

$$Y_i | (Z_i = \mathbf{B}) \sim F(\cdot, \boldsymbol{\theta}_b).$$

The mixing proportion τ is the probability that a random *unlabelled* observations is in class **B**. It is of interest to test whether τ is significantly larger than zero.

$$\tau = \mathbb{P}[Z_i = \mathbf{B} | X_i = 1],$$

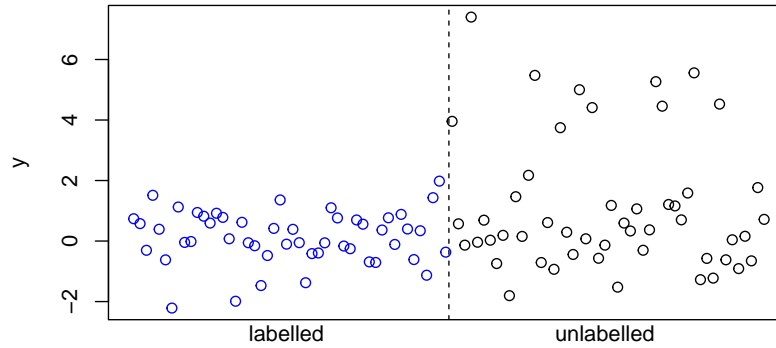
$$H_0 : \tau = 0,$$

$$H_1 : \tau > 0.$$

The function `mixture` estimates the unknown parameters ($\boldsymbol{\theta}_a$, $\boldsymbol{\theta}_b$, τ) and predicts the missing class labels in $\mathbf{z} = (z_1, \dots, z_n)^T$. The function `scrutor` tests homogeneity ($\tau = 0$) against heterogeneity ($\tau > 0$).

3 Model fitting

Observing two groups of observations, we assume the *labelled* observations are in class **A**, and the *unlabelled* observations are in class **A** or in class **B**.

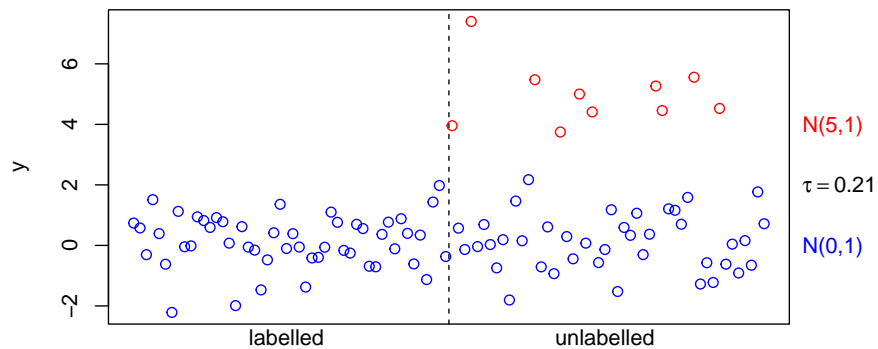


The function `mixtura` estimates the unknown parameters and predicts the missing class labels:

```
fit <- mixtura(y,z)
```

Here, 21% of the *unlabelled* observations are assigned to class **B**, and all other observations are assigned to class **A**:

```
class <- round(fit$posterior)
```



These are the parameter estimates:

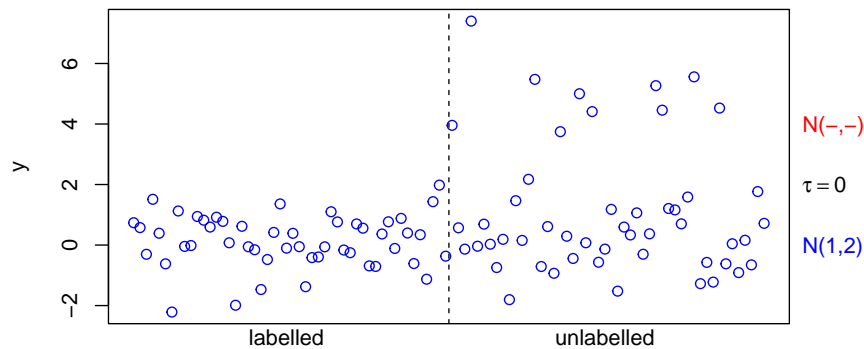
```
fit$estim1
```

4 Hypothesis testing

Under the null hypothesis, all observations are in class **A**. Under the alternative hypothesis, some *unlabelled* observations are in class **B**.

The function `mixture` not only fits the model under the alternative hypothesis (see above), but also under the null hypothesis:

```
fit$estim0
```



Because the null distribution of the likelihood-ratio test statistic is unknown, we compare the hypotheses by resampling. The function `scrutor` uses parametric bootstrapping or permutation:

```
scrutor(y,z)
```

If the p -value is less than or equal to the significance level, we reject the null hypothesis in favour of the alternative hypothesis.

Options

The functions `mixture` and `scrutor` have similar arguments. Set `dist` equal to "norm" or "nbinom" to choose between the Gaussian and the negative binomial distributions. In the latter case, optionally provide a dispersion estimate `phi` or an offset `gamma`. All other arguments are technical.

5 Application

5.1 Data preparation

Let n be the sample size, q the number of quantitative traits, and p the number of single nucleotide polymorphisms (SNPs).

- Transform the quantitative trait to a vector of length n , or transform the quantitative traits to a matrix with n rows (samples) and q columns (variables).
- Transform the SNP to a vector of length n , or transform the SNPs to a matrix with n rows (samples) and p columns (variables).
- Binarise the SNP(s), indicating the *labelled* group by zero, and the *unlabelled* group by a missing value.

For example, assign observations with zero minor alleles to the *labelled* group, and those with one or two minor alleles to the *unlabelled* group:

```
n <- length(snp)
```

```
z <- rep(NA, times=n)  
z[snp==0] <- 0
```

```
n <- nrow(SNPs)
```

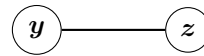
```
p <- ncol(SNPs)
```

```
Z <- matrix(NA, nrow=n, ncol=p)  
Z[SNPs==0] <- 0
```

5.2 Test of association

Use `scrutor` to test for association between a quantitative trait (vector) and a SNP (vector). The function returns a test statistic and a p -value.

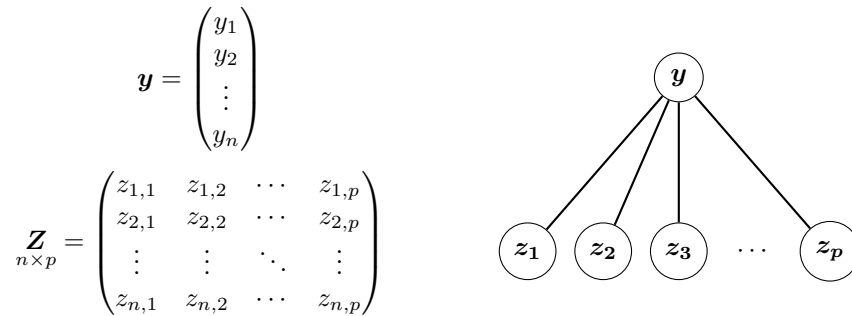
$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$$



```
scrutor(y, z)
```

5.3 Genome-wide association study

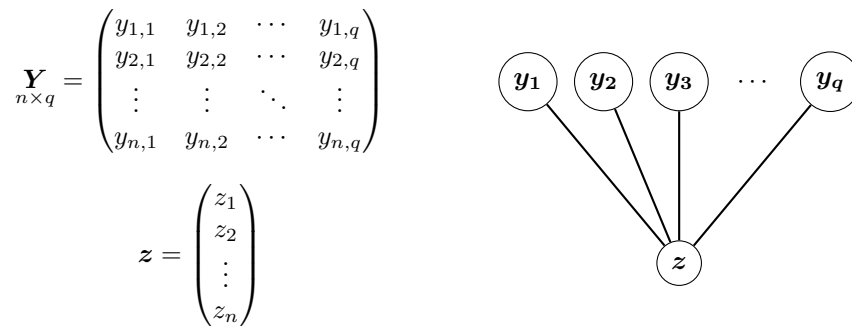
Use `scrutor` to test for association between a quantitative trait (vector) and several SNPs (matrix). For each SNP, the function returns a test statistic and a p -value.



```
scrutor(y,Z)
```

5.4 Differential expression analysis

Use `scrutor` to test for association between several quantitative traits (matrix) and a SNP (vector). For each quantitative trait, the function returns a test statistic and a p -value.



```
scrutor(Y,z)
```

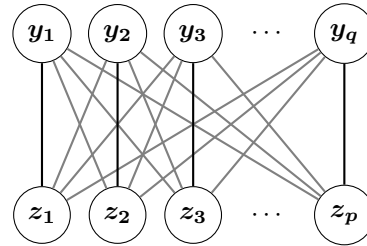
5.5 Expression quantitative trait loci analysis

Use `scrutor` to test for association between several quantitative traits (matrix) and several SNPs (matrix). If their numbers are different, all pairwise combinations are considered. If their numbers are equal, a one-to-one correspondence

is assumed. For each combination, the function returns a test statistic and a p -value.

$$\mathbf{Y}_{n \times q} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,q} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & \cdots & y_{n,q} \end{pmatrix}$$

$$\mathbf{Z}_{n \times p} = \begin{pmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,p} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n,1} & z_{n,2} & \cdots & z_{n,p} \end{pmatrix}$$



```
scrutor(Y,Z)
```

References

The R package `semisup` is based on Rauschenberger et al. [1], where detailed references to previous work are given. If you use `semisup` for publications, please cite Rauschenberger et al. [1].

Consider shrinkage estimation (Robinson et al. [3]) and scale normalisation (Robinson et al. [2]) to improve the negative binomial mixture model (R package `edgeR`). Use the non-parametric mixture test (van Wieringen et al. [4]) to increase robustness against outliers (R package `PDGETest`).

- [1] Armin Rauschenberger, Renée X Menezes, Mark A van de Wiel, Natasja M van Schoor, and Marianne A Jonker. Detecting SNPs with interactive effects on a quantitative trait. *Manuscript in preparation*, 0:0, 2018.
- [2] Mark D Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of RNA-Seq data. *Genome Biology*, 11(3):R25, 2010. [link](#).
- [3] Mark D Robinson and Gordon K Smyth. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, 9(2):321–332, 2008. [link](#).
- [4] Wessel N Van Wieringen, Mark A van de Wiel, and Aad W van der Vaart. A test for partial differential expression. *Journal of the American Statistical Association*, 103(483):1039–1049, 2008. [link](#).