

# Package: optBiomarker (via r-universe)

August 30, 2024

**Type** Package

**Title** Estimation of Optimal Number of Biomarkers for Two-Group  
Microarray Based Classifications at a Given Error Tolerance  
Level for Various Classification Rules

**Version** 1.0-28

**Date** 2020-12-23

**Depends** R (>= 2.10), rpanel

**Imports** MASS, randomForest, e1071, ipred, msm, rgl, Matrix

**Author** Mizanur Khondoker <mizanur.khondoker@gmail.com>

**Maintainer** Mizanur Khondoker <mizanur.khondoker@gmail.com>

**Description** Estimates optimal number of biomarkers for two-group  
classification based on microarray data.

**License** GPL (>= 2)

**Date/Publication** 2021-01-18 23:50:02 UTC

**NeedsCompilation** no

**Repository** <https://mkhondoker.r-universe.dev>

**RemoteUrl** <https://github.com/cran/optBiomarker>

**RemoteRef** HEAD

**RemoteSha** 468d5a83aad251b76a637c1a4f4f377412517834

## Contents

optBiomarker-package . . . . .	2
classificationError . . . . .	3
errorDbase . . . . .	5
optimiseBiomarker . . . . .	6
realBiomarker . . . . .	8
simData . . . . .	8

<b>Index</b>	<b>11</b>
--------------	-----------

---

optBiomarker-package *R package for estimating optimal number of biomarkers at a given error tolerance level for various classification rules*

---

## Description

Using interactive control panel ([rpanel](#)) and 3D real-time rendering system ([rgl](#)), this package provides a user friendly GUI for estimating the minimum number of biomarkers (variables) needed to achieve a given level of accuracy for two-group classification problems based on microarray data.

## Details

The function [optimiseBiomarker](#) is a user friendly GUI for interrogating the database of leave-one-out cross-validation errors, [errorDbase](#), to estimate optimal number of biomarkers for microarray based classifications. The database is built on the basis of simulated data using the [classificationError](#) function. The function [simData](#) is used for simulating microarray data for various combinations of factors such as the number of biomarkers, training set size, biological variation, experimental variation, fold change, replication, and correlation.

## Author(s)

Mizanur Khondoker, Till Bachmann, Peter Ghazal

Maintainer: Mizanur Khondoker <mizanur.khondoker@gmail.com>.

## References

Khondoker, M. R., Till T. Bachmann, T. T., Mewissen, M., Dickinson, P. *et al.* (2010). Multifactorial analysis of class prediction error: estimating optimal number of biomarkers for various classification rules. *Journal of Bioinformatics and Computational Biology*, **8**, 945-965.

Breiman, L. (2001). *Random Forests*, *Machine Learning* **45**(1), 5-32.

Chang, Chih-Chung and Lin, Chih-Jen: *LIBSVM: a library for Support Vector Machines*, <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

Efron, B. and Tibshirani, R. (1997). Improvements on Cross-Validation: The .632+ Bootstrap Estimator. *Journal of the American Statistical Association* **92**(438), 548-560.

Bowman, A., Crawford, E., Alexander, G. and Bowman, R. W. (2007). rpanel: Simple interactive controls for R functions using the tcltk package. *Journal of Statistical Software* **17**(9).

## See Also

[simData](#) [classificationError](#) [optimiseBiomarker](#)

**Examples**

```

if(interactive()){
  data(errorDbase)
  optimiseBiomarker(error=errorDbase)
}

```

---

classificationError     *Estimation of misclassification errors (generalisation errors) based on statistical and various machine learning methods*

---

**Description**

Estimates misclassification errors (generalisation errors), sensitivity and specificity using cross-validation, bootstrap and 632plus bias corrected bootstrap methods based on Random Forest, Support Vector Machines, Linear Discriminant Analysis and k-Nearest Neighbour methods.

**Usage**

```

## S3 method for class 'data.frame'
classificationError(
  formula,
  data,
  method=c("RF", "SVM", "LDA", "KNN"),
  errorType = c("cv", "boot", "six32plus"),
  senSpec=TRUE,
  negLevLowest=TRUE,
  na.action=na.omit,
  control=control.errorest(k=NROW(na.action(data)), nboot=100),
  ...)

```

**Arguments**

formula	A formula of the form lhs ~ rhs relating response (class) variable and the explanatory variables. See <a href="#">lm</a> for more detail.
data	A data frame containing the response (class membership) variable and the explanatory variables in the formula.
method	A character vector of length 1 to 4 representing the classification methods to be used. Can be one or more of "RF" (Random Forest), "SVM" (Support Vector Machines), "LDA" (Linear Discriminant Analysis) and "KNN" (k-Nearest Neighbour). Defaults to all four methods.
errorType	A character vector of length 1 to 3 representing the type of estimators to be used for computing misclassification errors. Can be one or more of the "cv" (cross-validation), "boot" (bootstrap) and "632plus" (632plus bias corrected bootstrap) estimators. Defaults to all three estimators.
senSpec	Logical. Should sensitivity and specificity (for cross-validation estimator only) be computed? Defaults to TRUE.

negLevLowest	Logical. Is the lowest of the ordered levels of the class variable represents the negative control? Defaults to TRUE.
na.action	Function which indicates what should happen when the data contains NA's, defaults to <code>na.omit</code> .
control	Control parameters of the the function <code>errorest</code> .
...	additional parameters to method.

### Details

In the current version of the package, estimation of sensitivity and specificity is limited to cross-validation estimator only. For LDA sample size must be greater than the number of explanatory variables to avoid singularity. The function `classificationError` does not check if this is satisfied, but the underlying function `lda` produces warnings if this condition is violated.

### Value

Returns an object of class `classificationError` with components

call	The call of the <code>classificationError</code> function.
errorRate	A <code>length(errorType)</code> by <code>length(method)</code> matrix of classification errors.
rocData	A 2 by <code>length(method)</code> matrix of sensitivities (first row) and specificities (second row).

### Author(s)

Mizanur Khondoker, Till Bachmann, Peter Ghazal  
 Maintainer: Mizanur Khondoker <mizanur.khondoker@gmail.com>.

### References

- Khondoker, M. R., Till T. Bachmann, T. T., Mewissen, M., Dickinson, P. *et al.* (2010). Multi-factorial analysis of class prediction error: estimating optimal number of biomarkers for various classification rules. *Journal of Bioinformatics and Computational Biology*, **8**, 945-965.
- Breiman, L. (2001). *Random Forests*, Machine Learning **45**(1), 5-32.
- Chang, Chih-Chung and Lin, Chih-Jen: *LIBSVM: a library for Support Vector Machines*, <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.
- Efron, B. and Tibshirani, R. (1997). Improvements on Cross-Validation: The .632+ Bootstrap Estimator. *Journal of the American Statistical Association* **92**(438), 548-560.

### See Also

[simData](#)

**Examples**

```
## Not run:
mydata<-simData(nTrain=30,nBiom=3)$data
classificationError(formula=class~., data=mydata)

## End(Not run)
```

---

errorDbase	<i>Database of leave-one-out cross validation errors for various combinations of data characteristics</i>
------------	---

---

**Description**

This is a 7-dimensional array (database) of leave-one-out cross validation errors for Random Forest, Support Vector Machines, Linear Discriminant Analysis and k-Nearest Neighbour classifiers. The database is the basis for estimating the optimal number of biomarkers at a given error tolerance level using [optimiseBiomarker](#) function. See **Details** for more information.

**Usage**

```
data(errorDbase)
```

**Format**

7-dimensional numeric array.

**Details**

The following table gives the dimension names, lengths and values/levels of the data object errorDbase.

<b>Dimension name</b>	<b>Length</b>	<b>Values/Levels</b>
No. of biomarkers	14	(1-6, 7, 9, 11, 15, 20, 30, 40, 50, 100)
Size of replication	5	(1, 3, 5, 7, 10)
Biological variation ( $\sigma_b$ )	4	(0.5, 1.0, 1.5, 2.5)
Experimental variation ( $\sigma_e$ )	4	(0.1, 0.5, 1.0, 1.5)
Minimum (Average) fold change	4	(1 (1.73), 2(2.88), 3(4.03), 5(6.33))
Training set size	5	(10, 20, 50, 100, 250)
Classification method	3	(Random Forest, Support Vector Machine, k-Nearest Neighbour)

We have a plan to expand the database to a 8-dimensional one by adding another dimension to store error rates at different level of correlation between biomarkers. Length of each dimension will also be increased leading to a bigger database with a wider coverage of the parameter space. Current version of the database contain error rates for independent (correlation = 0) biomarkers only. Also, it does not contain error rates for Linear Discriminant Analysis, which we plan to implement in the next release of the package. With the current version of the database, optimal number of biomarkers can be estimated using the [optimiseBiomarker](#) function for any intermediate values of the factors represented by the dimensions of the database.

**Author(s)**

Mizanur Khondoker, Till Bachmann, Peter Ghazal  
 Maintainer: Mizanur Khondoker <mizanur.khondoker@gmail.com>.

**References**

Khondoker, M. R., Till T. Bachmann, T. T., Mewissen, M., Dickinson, P. *et al.* (2010). Multi-factorial analysis of class prediction error: estimating optimal number of biomarkers for various classification rules. *Journal of Bioinformatics and Computational Biology*, **8**, 945-965.

**See Also**

[optimiseBiomarker](#)

---

optimiseBiomarker	<i>Estimates optimal number of biomarkers at a given error tolerance level for various classification rules</i>
-------------------	---

---

**Description**

Using interactive control panel (see [rpanel](#)) and 3D real-time rendering system ([rgl](#)), this package provides a user friendly GUI for estimating the minimum number of biomarkers (variables) needed to achieve a given level of accuracy for two-group classification problems based on microarray data.

**Usage**

```
optimiseBiomarker (error,
                  errorTol = 0.05,
                  method = "RF", nTrain = 100,
                  sdB = 1.5,
                  sdW = 1,
                  foldAvg = 2.88,
                  nRep = 3)
```

**Arguments**

error	The database of classification errors. See errorDbase for details.
errorTol	Error tolerance limit.
method	Classification method. Can be one of "RF", "SVM", and "KNN" for Random Forest, Support Vector Machines, Linear Discriminant Analysis and k-Nearest Neighbour respectively.
nTrain	Training set size, i.e., the total number of biological samples in group 1 and group 2.
sdB	Biological variation ( $\sigma_b$ ) of data in log (base 2) scale.

sdW	Experimental (technical) variation ( $\sigma_e$ ) of data in log (base 2) scale.
foldAvg	Average fold change of the biomarkers.
nRep	Number of technical replications.

## Details

The function `optimiseBiomarker` is a user friendly GUI for interrogating the database of leave-one-out cross-validation errors, `errorDbase`, to estimate optimal number of biomarkers for microarray based classifications. The database is built on the basis of simulated data using the `classificationError` function. The function `simData` is used for simulating microarray data for various combinations of factors such as the number of biomarkers, training set size, biological variation, experimental variation, fold change, replication, and correlation.

## Author(s)

Mizanur Khondoker, Till Bachmann, Peter Ghazal

Maintainer: Mizanur Khondoker <mizanur.khondoker@gmail.com>.

## References

Khondoker, M. R., Till T. Bachmann, T. T., Mewissen, M., Dickinson, P. *et al.* (2010). Multi-factorial analysis of class prediction error: estimating optimal number of biomarkers for various classification rules. *Journal of Bioinformatics and Computational Biology*, **8**, 945-965.

Breiman, L. (2001). *Random Forests*, Machine Learning **45**(1), 5–32.

Chang, Chih-Chung and Lin, Chih-Jen: *LIBSVM: a library for Support Vector Machines*, <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

Efron, B. and Tibshirani, R. (1997). Improvements on Cross-Validation: The .632+ Bootstrap Estimator. *Journal of the American Statistical Association* **92**(438), 548–560.

Bowman, A., Crawford, E., Alexander, G. and Bowman, R. W. (2007). rpanel: Simple interactive controls for R functions using the tcltk package. *Journal of Statistical Software* **17**(9).

## See Also

`simData` `classificationError`

## Examples

```
if(interactive()){
  data(errorDbase)
  optimiseBiomarker(error=errorDbase)
}
```

---

realBiomarker	<i>A set of 54359 median gene expressions in log (base 2) scale</i>
---------------	---

---

### Description

This data set contains a set of 54359 log base 2 gene expression values from a neonatal whole blood gene expression study described in Smith *et al.* (2007). The data represent the median of 28 microarrays corresponding to 28 control (healthy) patients of the neonatal study. This data set is used as a base expressions set for simulating biomarker data using `simData` function of the `optBiomarker` package.

### Usage

```
data(realBiomarker)
```

### Format

A vector of 54359 gene expressions in log (base 2) scale.

### References

Smith, C. L., Dickinson, P., Forster, T., Khondoker, M. R., Craigon, M., Ross, A., Storm, P., Burgess, S., Lacaze, P., Stenson, B. J. and Ghazal, P. (2007). Quantitative assessment of whole blood RNA as a potential biomarker for infectious disease. *Analyst* **132**, 1200–1209.

Khondoker, M. R., Till T. Bachmann, T. T., Mewissen, M., Dickinson, P. *et al.* (2010). Multi-factorial analysis of class prediction error: estimating optimal number of biomarkers for various classification rules. *Journal of Bioinformatics and Computational Biology*, **8**, 945-965.

---

simData	<i>Simulation of microarray data</i>
---------	--------------------------------------

---

### Description

The function simulates microarray data for two-group comparison with user supplied parameters such as number of biomarkers (genes or proteins), sample size, biological and experimental (technical) variation, replication, differential expression, and correlation between biomarkers.

### Usage

```
simData(nTrain=100,
        nGr1=floor(nTrain/2),
        nBiom=50, nRep=3,
        sdW=1.0,
        sdB=1.0,
        rhoMax=NULL, rhoMin=NULL, nBlock=NULL, bsMin=3, bSizes=NULL, gamma=NULL,
```



```
sigma=0.1,diffExpr=TRUE,
foldMin=2,
orderBiom=TRUE,
baseExpr=NULL)
```

### Arguments

nTrain	Training set size, i.e., the total number of biological samples in group 1 (nGr1) and group 2.
nGr1	Size of group 1. Defaults to $\text{floor}(\text{nTrain}/2)$ .
nBiom	Number of biomarkers (genes, probes or proteins).
nRep	Number of technical replications.
sdW	Experimental (technical) variation ( $\sigma_e$ ) of data in log (base 2) scale.
sdB	Biological variation ( $\sigma_b$ ) of data in log (base 2) scale.
rhoMax	Maximum Pearson's correlation coefficient between biomarkers. To ensure positive definiteness, allowed values are restricted between 0 and 0.95 inclusive. If NULL, set to $\text{runif}(1, \text{min}=0.6, \text{max}=0.8)$ .
rhoMin	Minimum Pearson's correlation coefficient between biomarkers. To ensure positive definiteness, allowed values are restricted between 0 and 0.95 inclusive. If NULL, set to $\text{runif}(1, \text{min}=0.2, \text{max}=0.4)$ .
nBlock	Number of blocks in the block diagonal (Hub-Toeplitz) correlation matrix. If NULL, set to 1 for $\text{nBiom} < 5$ and randomly selected from $c(1 : \text{floor}(\text{nBiom}/\text{bsMin}))$ for $\text{nBiom} \geq 5$ .
bsMin	Minimum block size. $\text{bsMin}=3$ by default.
bSizes	A vector of length nBlock representing the block sizes (should sum to nBlock). If NULL, set to $c(\text{bs} + \text{mod}, \text{rep}(\text{bs}, \text{nBlock}-1))$ , where bs is the integer part of $\text{nBiom}/\text{nBlock}$ and mod is the remainder after integer division.
gamma	Specifies a correlation structure. If NULL, assumes independence. $\text{gamma}=0$ indicates a single block exchangeable correlation matrix with constant correlation $\text{rho}=0.5 * (\text{rhoMin} + \text{rhoMax})$ . A value greater than zero indicates block diagonal (Hub-Toeplitz) correlation matrix with decline rate determined by the value of gamma. Decline rate is linear for $\text{gamma}=1$ .
sigma	Standard deviation of the normal distribution (before truncation) where fold changes are generated from. See details.
diffExpr	Logical. Should systematic difference be introduced between the data of the two groups?
foldMin	Minimum value of fold changes. See details.
orderBiom	Logical. Should columns (biomarkers) be arranged in order of differential expression?
baseExpr	A vector of length nBiom to be used as base expressions $\mu$ . See <code>realBiomarker</code> for details.

## Details

Differential expressions are introduced by adding  $z\delta$  to the data of group 2 where  $\delta$  values are generated from a truncated normal distribution and  $z$  is randomly selected from  $(-1, 1)$  to characterise up- or down-regulation.

Assuming that  $Y$  is  $N(\mu, \sigma^2)$ , and  $A = [a_1, a_2]$ , a subset of  $-Inf < y < Inf$ , the conditional distribution of  $Y$  given  $A$  is called truncated normal distribution:

$$f(y, \mu, \sigma) = (1/\sigma)\phi((y - \mu)/\sigma)/(\Phi((a_2 - \mu)/\sigma) - \Phi((a_1 - \mu)/\sigma))$$

for  $a_1 \leq y \leq a_2$ , and 0 otherwise,

where  $\mu$  is the mean of the original Normal distribution before truncation,  $\sigma$  is the corresponding standard deviation,  $a_2$  is the upper truncation point,  $a_1$  is the lower truncation point,  $\phi(x)$  is the density of the standard normal distribution, and  $\Phi(x)$  is the distribution function of the standard normal distribution. For `simData` function, we consider  $a_1 = \log_2(\text{foldMin})$  and  $a_2 = Inf$ . This ensures that the biomarkers are differentially expressed by a fold change of `foldMin` or more.

## Value

A dataframe of dimension `nTrain` by `nBiom+1`. The first column is a factor (`class`) representing the group memberships of the samples.

## Author(s)

Mizanur Khondoker, Till Bachmann, Peter Ghazal  
Maintainer: Mizanur Khondoker <mizanur.khondoker@gmail.com>.

## References

Khondoker, M. R., Till T. Bachmann, T. T., Mewissen, M., Dickinson, P. *et al.* (2010). Multifactorial analysis of class prediction error: estimating optimal number of biomarkers for various classification rules. *Journal of Bioinformatics and Computational Biology*, **8**, 945-965.

## See Also

[classificationError](#)

## Examples

```
simData(nTrain=10, nBiom=3)
```

# Index

- \* **classif**
  - classificationError, 3
- \* **datagen**
  - simData, 8
- \* **datasets**
  - errorDbase, 5
  - realBiomarker, 8
- \* **optimize**
  - optimiseBiomarker, 6
- \* **package**
  - optBiomarker-package, 2

classificationError, 2, 3, 7, 10

errorDbase, 2, 5, 7

errorest, 4

lda, 4

lm, 3

na.omit, 4

optBiomarker, 8

optBiomarker (optBiomarker-package), 2

optBiomarker-package, 2

optimiseBiomarker, 2, 5, 6, 6, 7

realBiomarker, 8

rgl, 2, 6

rpanel, 2, 6

simData, 2, 4, 7, 8, 8