Package ‘optBiomarker’

March 4, 2013

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 1.0-24

Date 2013-03-03

Depends R (>= 2.10), rpanel, rgl

Imports MASS, randomForest, e1071,ipred, msm

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

Maintainer Mizanur Khondoker <mizanur.khondoker@googlemail.com>

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

License GPL (>= 2)

Repository CRAN

Date/Publication 2013-03-04 07:46:02

NeedsCompilation no

R topics documented:

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

Index 11
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


See Also

`simData` `classificationError` `optimiseBiomarker`
classificationError

**Examples**

```r
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**

```r
## S3 method for class 'data.frame'
classificationError(
  formula,  # A formula of the form lhs ~ rhs relating response (class) variable and the explanatory variables. See `lm` for more detail.
  data,     # A data frame containing the response (class membership) variable and the explanatory variables in the formula.
  method=c("RF","SVM","LDA","KNN"),  # 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 = c("cv", "boot", "six32plus"),  # 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=TRUE,  # Logical. Should sensitivity and specificity (for cross-validation estimator only) be computed? Defaults to TRUE.
  na.action=na.omit,  # 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.
  control=control.errorest(k=NROW(na.action(data)),nboot=100),  # 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.
  ...)
```

**Arguments**

- `formula`: A formula of the form `lhs ~ rhs` relating response (class) variable and the explanatory variables. See `lm` 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.
classificationError

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 na.omit.

control Control parameters of the the function errorest.

... 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 classificationError function.

errorRate A length(errorType) by length(method) matrix of classification errors.

rocData A 2 by length(method) 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


See Also

simData
Examples

```r
mydata<-simData(nTrain=30,nBiom=3)
classificationError(formula=class~., data=mydata)
```

---

**errorDbase**

*Database of leave-one-out cross validation errors for various combinations of data characteristics*

**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**

```r
data(errorDbase)
```

**Format**

7-dimensional numeric array.

**Details**

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

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

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

See Also
optimiseBiomarker

optimiseBiomarker Estimates optimal number of biomarkers at a given error tolerance level for various classification rules

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

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


**See Also**

`simData`, `classificationError`

**Examples**

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

A set of 54359 median gene expressions in log (base 2) scale

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


simData  

Simulation of microarray data

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, rho=0, sigma=0.1, diffExpr=TRUE,
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 floor(nTrain/2).
- **nBiom**: Number of biomarkers (genes, probes or proteins).
- **nRep**: Number of technical replications.
- **sdW**: Experimental (technical) variation (σₑ) of data in log (base 2) scale.
- **sdB**: Biological variation (σᵇ) of data in log (base 2) scale.
- **rho**: Common Pearson correlation between biomarkers. To ensure positive definiteness, allowed values of rho are restricted between 0 and 0.95 inclusive.
- **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 `realBiomarker` 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 \( Y \sim N(\mu, \sigma^2) \), and \( A = [a_1, a_2] \), a subset of \(-Inf < y < Inf\), the conditional distribution of \( Y \) given \( A \) is called truncated normal distribution:

\[
f(y, \mu, \sigma) = \frac{1/\sigma \phi((y - \mu)/\sigma) / (\Phi((a_2 - \mu)/\sigma) - \Phi((a_1 - \mu)/\sigma))}{\Phi((a_2 - \mu)/\sigma) - \Phi((a_1 - \mu)/\sigma)}
\]

for \( a_1 <= y <= a_2 \), 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 \( \text{foldMin} \) 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

See Also
classificationError

Examples

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

*Topic classif
  classificationError, 3
*Topic datagen
  simData, 8
*Topic datasets
  errorDbase, 5
  realBiomarker, 8
*Topic optimize
  optimiseBiomarker, 6
*Topic 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