

# Package ‘MiPP’

April 16, 2019

**Version** 1.54.0

**Date** 2007-01-31

**Title** Misclassification Penalized Posterior Classification

**Author** HyungJun Cho <hj4cho@korea.ac.kr>,  
Sukwoo Kim <s4kim@korea.ac.kr>,  
Mat Soukup <soukup@fda.gov>, and  
Jae K. Lee <jaeklee@virginia.edu>

**Maintainer** Sukwoo Kim <s4kim@korea.ac.kr>

**Depends** R (>= 2.4)

**Imports** Biobase, e1071, MASS, stats

**Description** This package finds optimal sets of genes that separate samples into two or more classes.

**License** GPL (>= 2)

**URL** <http://www.healthsystem.virginia.edu/internet/hes/biostat/bioinformatics/>

**biocViews** Microarray, Classification

**git\_url** <https://git.bioconductor.org/packages/MiPP>

**git\_branch** RELEASE\_3\_8

**git\_last\_commit** 4f52b11

**git\_last\_commit\_date** 2018-10-30

**Date/Publication** 2019-04-15

## R topics documented:

colon . . . . .	2
cv.mipp.rule . . . . .	2
get.mipp . . . . .	2
get.mipp.lda . . . . .	3
get.mipp.logistic . . . . .	3
get.mipp.qda . . . . .	3
get.mipp.svm.linear . . . . .	3
get.mipp.svm.rbf . . . . .	3
leuk1 . . . . .	4
leuk2 . . . . .	4
leukemia . . . . .	5
linearkernel.decision.function . . . . .	5

mipp . . . . .	5
mipp.preproc . . . . .	7
mipp.rule . . . . .	8
mipp.seq . . . . .	8
pre.select . . . . .	11
quant.normal . . . . .	11
quant.normal2 . . . . .	11
rbfkernel.decision.function . . . . .	11
<b>Index</b>	<b>12</b>

---

colon	<i>Gene expression data for colon cancer</i>
-------	--

---

**Description**

This data set consists of gene expression of colon cancer study.

**Usage**

data(colon)

**Format**

A matrix containing 2000 probe sets and 2 classes (T, F)

**Source**

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J. (1999). Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues probed by Oligonucleotide Arrays, PNAS, 96(12), 6745–6750.

---

cv.mipp.rule	<i>Fitting cross-validation MiPP</i>
--------------	--------------------------------------

---

**Description**

Fits cross-validation MiPP

---

get.mipp	<i>Choosing a rule</i>
----------	------------------------

---

**Description**

Choose a rule to compute MiPP

---

get.mipp.lda	<i>Fitting LDA to compute MiPP</i>
--------------	------------------------------------

---

**Description**

Fits LDA to compute MiPP

---

get.mipp.logistic	<i>Fitting logistic model to compute MiPP</i>
-------------------	---

---

**Description**

Fits logistic model to compute MiPP

---

get.mipp.qda	<i>Fitting QDA to compute MiPP</i>
--------------	------------------------------------

---

**Description**

Fits QDA to compute MiPP

---

get.mipp.svm.linear	<i>Fitting SVM (linear) to compute MiPP</i>
---------------------	---

---

**Description**

Fits SVM (linear) to compute MiPP

---

get.mipp.svm.rbf	<i>Fitting SVM (RBF) to compute MiPP</i>
------------------	--

---

**Description**

Fits SVM (RBF) to compute MiPP

---

leuk1	<i>Gene expression data for leukemia</i>
-------	--

---

**Description**

This data set consists of gene expression of leukemia study.

**Usage**

```
data(leukemia)
```

**Format**

A matrix containing 6817 probe sets and 38 samples (2 classes: AML, ALL)

**Source**

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, P., Coller, H., Loh, M.L., Downing, J.R., Caliguri, M.A., Bloomfield, C.D., and Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537.

---

leuk2	<i>Gene expression data for leukemia</i>
-------	--

---

**Description**

This data set consists of gene expression of leukemia study.

**Usage**

```
data(leukemia)
```

**Format**

A matrix containing 6817 probe sets and 34 samples (2 classes: AML, ALL)

**Source**

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, P., Coller, H., Loh, M.L., Downing, J.R., Caliguri, M.A., Bloomfield, C.D., and Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537.

leukemia

*Gene expression data for leukemia***Description**

This data set consists of gene expression of leukemia study.

**Usage**

```
data(leukemia)
```

**Format**

A matrix containing 6817 probe sets and 2 classes (AML, ALL)

**Source**

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, P., Coller, H., Loh, M.L., Downing, J.R., Caliguri, M.A., Bloomfield, C.D., and Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537.

```
linearkernel.decision.function
```

*SVM (linear) kernel to compute MiPP*

**Description**

SVM (linear) kernel to compute MiPP

mipp

*MiPP-based Classification***Description**

Finds optimal sets of genes for classification

**Usage**

```
mipp(x, y, x.test = NULL, y.test = NULL, probe.ID = NULL,
      rule = "lda", method.cut = "t.test", percent.cut = 0.01,
      model.sMiPP.margin = 0.01, min.sMiPP = 0.85, n.drops = 2,
      n.fold = 5, p.test = 1/3, n.split = 20,
      n.split.eval = 100)
```

**Arguments**

<code>x</code>	data matrix
<code>y</code>	class vector
<code>x.test</code>	test data matrix if available
<code>y.test</code>	test class vector if available
<code>probe.ID</code>	probe set IDs; if NULL, row numbers are assigned.
<code>rule</code>	classification rule: "lda", "qda", "logistic", "svmlin", "svmrbf"; the default is "lda".
<code>method.cut</code>	method for pre-selection; t-test is available.
<code>percent.cut</code>	proportion of pre-selected genes; the default is 0.01.
<code>model.sMiPP.margin</code>	smallest set of genes s.t. $sMiPP \leq (\max sMiPP - \text{model.sMiPP.margin})$ ; the default is 0.01.
<code>min.sMiPP</code>	Adding genes stops if max sMiPP is at least min.sMiPP; the default is 0.85.
<code>n.drops</code>	Adding genes stops if sMiPP decreases (n.drops) times, in addition to min.sMiPP criterion.; the default is 2.
<code>n.fold</code>	number of folds; default is 5.
<code>p.test</code>	partition percent of train and test samples when test samples are not available; the default is 1/3 for test set.
<code>n.split</code>	number of splits; the default is 20.
<code>n.split.eval</code>	numbr of splits for evalutation; the default is 100.

**Value**

<code>model</code>	candiadate genes (for each split if no indep set is available)
<code>model.eval</code>	Optimal sets of genes for each split when no indep set is available

**Author(s)**

Soukup M, Cho H, and Lee JK

**References**

Soukup M, Cho H, and Lee JK (2005). Robust classification modeling on microarray data using misclassification penalized posterior, *Bioinformatics*, 21 (Suppl): i423-i430.

Soukup M and Lee JK (2004). Developing optimal prediction models for cancer classification using gene expression data, *Journal of Bioinformatics and Computational Biology*, 1(4) 681-694

**Examples**

```
#####
#Example 1: When an independent test set is available

data(leukemia)

#Normalize combined data
leukemia <- cbind(leuk1, leuk2)
leukemia <- mipp.preproc(leukemia, data.type="MAS4")
```

```

#Train set
x.train <- leukemia[,1:38]
y.train <- factor(c(rep("ALL",27),rep("AML",11)))

#Test set
x.test <- leukemia[,39:72]
y.test <- factor(c(rep("ALL",20),rep("AML",14)))

#Compute MiPP
out <- mipp(x=x.train, y=y.train, x.test=x.test, y.test=y.test, probe.ID = 1:nrow(x.train), n.fold=5, percent

#Print candidate models
out$model

#####
#Example 2: When an independent test set is not available

data(colon)

#Normalize data
x <- mipp.preproc(colon)
y <- factor(c("T", "N", "T", "N", "T", "N", "T", "N", "T", "N",
              "T", "N", "T", "N", "T", "N", "T", "N", "T", "N",
              "T", "N", "T", "N", "T", "T", "T", "T", "T", "T",
              "T", "T", "T", "T", "T", "T", "T", "T", "N", "T",
              "T", "N", "N", "T", "T", "T", "T", "N", "T", "N",
              "N", "T", "T", "N", "N", "T", "T", "T", "T", "N",
              "T", "N"))

#Deleting contaminated chips
x <- x[,-c(51,55,45,49,56)]
y <- y[ -c(51,55,45,49,56)]

#Compute MiPP
out <- mipp(x=x, y=y, probe.ID = 1:nrow(x), n.fold=5, p.test=1/3, n.split=5, n.split.eval=100,
percent.cut= 0.1, rule="lda")

#Print candidate models for each split
out$model

#Print optimal models and independent evaluation for each split
out$model.eval

```

---

mipp.preproc

Preprocessing

---

## Description

Performs IQR normalization, thesholding, and log2-transformation

Usage

```
mipp.preproc(x, data.type = "MAS5")
```

Arguments

x	data
data.type	data type is MAS5, MAS4, or dChip

See Also

[mipp](#)

Examples

```
library(MiPP)

data(colon)
colon.nor <- mipp.preproc(colon)
```

---

mipp.rule	<i>Computing MiPP</i>
-----------	-----------------------

---

Description

Computes MiPP

---

mipp.seq	<i>MiPP-based Classification</i>
----------	----------------------------------

---

Description

sequentially finds optimal sets of genes for classification

Usage

```
mipp.seq(x, y, x.test = NULL, y.test = NULL, probe.ID = NULL,
rule = "lda", method.cut = "t.test", percent.cut = 0.01,
model.sMiPP.margin = 0.01, min.sMiPP = 0.85, n.drops = 2,
n.fold = 5, p.test = 1/3, n.split = 20, n.split.eval = 100,
n.seq=3, cutoff.sMiPP=0.7, remove.gene.each.model="all")
```



**Arguments**

<code>x</code>	data matrix
<code>y</code>	class vector
<code>x.test</code>	test data matrix if available
<code>y.test</code>	test class vector if available
<code>probe.ID</code>	probe set IDs; if NULL, row numbers are assigned.
<code>rule</code>	classification rule: "lda", "qda", "logistic", "svmlin", "svmrbf"; the default is "lda".
<code>method.cut</code>	method for pre-selection; t-test is available.
<code>percent.cut</code>	proportion of pre-selected genes; the default is 0.01.
<code>model.sMiPP.margin</code>	smallest set of genes s.t. $sMiPP \leq (\max sMiPP - model.sMiPP.margin)$ ; the default is 0.01.
<code>min.sMiPP</code>	Adding genes stops if max sMiPP is at least min.sMiPP; the default is 0.85.
<code>n.drops</code>	Adding genes stops if sMiPP decreases (n.drops) times, in addition to min.sMiPP criterion.; the default is 2.
<code>n.fold</code>	number of folds; default is 5.
<code>p.test</code>	partition percent of train and test samples when test samples are not available; the default is 1/3 for test set.
<code>n.split</code>	number of splits; the default is 20.
<code>n.split.eval</code>	numbr of splits for evaluation; the default is 100.
<code>n.seq</code>	Number of sequential gene model selection; the default is 3.
<code>cutoff.sMiPP</code>	Cutoff point of 5 percent sMiPP to select gene models
<code>remove.gene.each.model</code>	Re-run after removing all genes in the selected models if "all" and the first gene for each of the selected models if "first"

**Value**

<code>model</code>	candiadate genes (for each split if no indep set is available)
<code>model.eval</code>	Optimal sets of genes for each split when no indep set is available
<code>genes.selected</code>	a list of genes selected by sequential selection

**Author(s)**

Soukup M, Cho H, and Lee JK

**References**

- Soukup M, Cho H, and Lee JK (2005). Robust classification modeling on microarray data using misclassification penalized posterior, *Bioinformatics*, 21 (Suppl): i423-i430.
- Soukup M and Lee JK (2004). Developing optimal prediction models for cancer classification using gene expression data, *Journal of Bioinformatics and Computational Biology*, 1(4) 681-694

## Examples

```
#####
#Example 1: When an independent test set is available

data(leukemia)

#Normalize combined data
leukemia <- cbind(leuk1, leuk2)
leukemia <- mipp.preproc(leukemia, data.type="MAS4")

#Train set
x.train <- leukemia[,1:38]
y.train <- factor(c(rep("ALL",27),rep("AML",11)))

#Test set
x.test <- leukemia[,39:72]
y.test <- factor(c(rep("ALL",20),rep("AML",14)))

#Compute MiPP
out <- mipp.seq(x=x.train, y=y.train, x.test=x.test, y.test=y.test, n.fold=5, percent.cut=0.01, rule="lda",

#Print candidate models
out$model

#Print the genes selected
out$genes.selected

#####
#Example 2: When an independent test set is not available

data(colon)

#Normalize data
x <- mipp.preproc(colon)
y <- factor(c("T", "N", "T", "N", "T", "N", "T", "N", "T", "N",
              "T", "N", "T", "N", "T", "N", "T", "N", "T", "N",
              "T", "N", "T", "N", "T", "N", "T", "N", "T", "N",
              "T", "T", "T", "T", "T", "T", "T", "T", "N", "T",
              "T", "N", "N", "T", "T", "T", "T", "N", "T", "N",
              "N", "T", "T", "N", "N", "T", "T", "T", "T", "N",
              "T", "N"))

#Deleting contaminated chips
x <- x[,-c(51,55,45,49,56)]
y <- y[ -c(51,55,45,49,56)]

#Compute MiPP
out <- mipp.seq(x=x, y=y, n.fold=5, p.test=1/3, n.split=5, n.split.eval=100,
percent.cut= 0.05, rule="lda", n.seq=2)

#Print candidate models for each split
```

```

out$model

#Print optimal models and independent evaluation for each split
out$model.eval

#Print the genes selected
out$genes.selected

```

pre.select

*Pre-selection***Description**

Pre-select genes

quant.normal

*Quantile normalization***Description**

Performs quantile normalization

quant.normal2

*Quantile normalization***Description**

Performs quantile normalization

rbfkernel.decision.function

*SVM (RBF) kernel to compute MiPP***Description**

SVM (RBF) kernel to compute MiPP

# Index

## \*Topic **datasets**

- colon, [2](#)
- leuk1, [4](#)
- leuk2, [4](#)
- leukemia, [5](#)

## \*Topic **models**

- cv.mipp.rule, [2](#)
- get.mipp, [2](#)
- get.mipp.lda, [3](#)
- get.mipp.logistic, [3](#)
- get.mipp.qda, [3](#)
- get.mipp.svm.linear, [3](#)
- get.mipp.svm.rbf, [3](#)
- linearkernel.decision.function, [5](#)
- mipp, [5](#)
- mipp.preproc, [7](#)
- mipp.rule, [8](#)
- mipp.seq, [8](#)
- pre.select, [11](#)
- quant.normal, [11](#)
- quant.normal2, [11](#)
- rbfkernel.decision.function, [11](#)

- colon, [2](#)
- cv.mipp.rule, [2](#)

- get.mipp, [2](#)
- get.mipp.lda, [3](#)
- get.mipp.logistic, [3](#)
- get.mipp.qda, [3](#)
- get.mipp.svm.linear, [3](#)
- get.mipp.svm.rbf, [3](#)

- leuk1, [4](#)
- leuk2, [4](#)
- leukemia, [5](#)
- leukimia (leukemia), [5](#)
- linearkernel.decision.function, [5](#)

- mipp, [5, 8](#)
- mipp.preproc, [7](#)
- mipp.rule, [8](#)
- mipp.seq, [8](#)

- pre.select, [11](#)

- quant.normal, [11](#)
- quant.normal2, [11](#)

- rbfkernel.decision.function, [11](#)