

# Package ‘MiChip’

January 19, 2025

**Version** 1.60.0

**Date** 2009-09-10

**Title** MiChip Parsing and Summarizing Functions

**Author** Jonathon Blake <blake@embl.de>

**Maintainer** Jonathon Blake <blake@embl.de>

**Depends** R (>= 2.3.0), Biobase

**Imports** Biobase

**Description** This package takes the MiChip miRNA microarray .grp scanner output files and parses these out, providing summary and plotting functions to analyse MiChip hybridizations. A set of hybridizations is packaged into an ExpressionSet allowing it to be used by other BioConductor packages.

**License** GPL (>= 2)

**biocViews** Microarray, Preprocessing

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

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** 7f352a5

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2025-01-19

## Contents

|                                     |   |
|-------------------------------------|---|
| boxplotData . . . . .               | 2 |
| boxplotDataNoFile . . . . .         | 2 |
| correctForFlags . . . . .           | 3 |
| MiChip . . . . .                    | 3 |
| myForgivingMedian . . . . .         | 4 |
| naOmitMedian . . . . .              | 4 |
| normalizePerChipMedian . . . . .    | 5 |
| outputAnnotatedDataMatrix . . . . . | 5 |
| panelCor . . . . .                  | 6 |
| parseRawData . . . . .              | 6 |
| plotIntensitiesScatter . . . . .    | 7 |
| removeUnwantedRows . . . . .        | 8 |

|  |    |
|--|----|
| returnAnnotatedDataMatrix . . . . .      | 8  |
| setIntensityCutoff . . . . .             | 9  |
| standardRemoveRows . . . . .             | 9  |
| summarizeIntensitiesAsMedian . . . . .   | 10 |
| workedExampleMedianNormalize . . . . .   | 10 |
| workedExampleNotNormalizedData . . . . . | 11 |

## Index 12

---

|             |                               |
|-------------|-------------------------------|
| boxplotData | <i>Create Boxplot of data</i> |
|-------------|-------------------------------|

---

### Description

Creates a boxplot of expression data contained in a matrix and writes this to a file.

### Usage

```
boxplotData(dmat, exptname, dlevel)
```

### Arguments

|          |  |
|----------|--|
| dmat     | matrix containing expression data to be boxplotted       |
| exptname | Name of the experiment, used to build filename           |
| dlevel   | Stage of the experiment e.g. raw, summarized, normalized |

### Examples

```
#Create a file of a boxplot containing normalized expression data for myexpt
## Not run:
boxplotData(dmat, "MyExpt", "mednormed")

## End(Not run)
```

---

|                   |                               |
|-------------------|-------------------------------|
| boxplotDataNoFile | <i>Create Boxplot of data</i> |
|-------------------|-------------------------------|

---

### Description

Creates a boxplot of expression data contained in a matrix.

### Usage

```
boxplotDataNoFile(dmat, exptname, dlevel)
```

### Arguments

|          |  |
|----------|--|
| dmat     | matrix containing expression data to be boxplotted       |
| exptname | Name of the experiment, used to build plot title         |
| dlevel   | Stage of the experiment e.g. raw, summarized, normalized |

**Examples**

```
#Create a boxplot containing normalized expression data for myexpt
## Not run: boxplotDataNoFile(dmat, "MyExpt", "mednormed")
```

---

|                 |  |
|-----------------|--|
| correctForFlags | <i>Corrects for spots flagged as not present</i> |
|-----------------|--|

---

**Description**

Spots flagged with a -ve quality flag value by the scanner may be regarded as not present. This method sets their intensity to NA.

**Usage**

```
correctForFlags(eset, intensityCutoff=0)
```

**Arguments**

|                 |  |
|-----------------|--|
| eset            | ExpressionSet containing intensity values and flags to be filtered |
| intensityCutoff | value of lowest acceptable intensity value in the experiment       |

**Examples**

```
#Correct ExpressionSet for flags of spots marked as unreadable
## Not run:
myCorrectedEset <-correctForFlags(eset, intensityCutoff=0)

## End(Not run)
```

---

|        |   |
|--------|---|
| MiChip | <i>Introduction to the MiChip Package</i> |
|--------|---|

---

**Description**

A library for processing MiChip hybridizations

**Author(s)**

Jonathon Blake

---

myForgivingMedian      *Produce Median from Probe Intensity values*

---

### Description

Creates a median to summarize the intensities for individual probes, giving that not all probes will have a valid intensity reading

### Usage

```
myForgivingMedian(mat, minSumlength=0)
```

### Arguments

mat                      matrix of data to calculate the median from  
minSumlength      The lowest acceptable length of the matrix to calculate a median

### Examples

```
#Calculate the median of a matrix omiting NAs
## Not run:
myForgivingMedian(mat, minSumlength=0)

## End(Not run)
```

---

naOmitMedian              *Calculates the median of an array excluding NAs*

---

### Description

Calculates the median of an array excluding NAs

### Usage

```
naOmitMedian(mat, madAdjust=FALSE)
```

### Arguments

mat                      A single dimensional matrix  
madAdjust              if TRUE then summarized data will be filtered according to the MAD median absolute deviation and set to NA if the median is less than MAD

### Examples

```
#Calculate the median of matrix mat omiting NAs
## Not run:
myMedian <-naOmitMedian(mat, madAdjust=TRUE)

## End(Not run)
```

---

 normalizePerChipMedian

*Normalize to median intensity*


---

**Description**

Normalizes intensity values to the median of each chip

**Usage**

```
normalizePerChipMedian(eset)
```

**Arguments**

eset                    ExpressionSet containing chip intensity values to be normalized

**Examples**

```
#Normalize expression data in an Eset to the median
## Not run:
normedDataEset <- normalizePerChipMedian(eset)

## End(Not run)
```

---

 outputAnnotatedDataMatrix

*Outputs a tab delimited file from an ExpressionSet*


---

**Description**

Takes an ExpressionSet and outputs a tab delimited file containing feature annotation to the left and hyb specific expression/flag data to the right

**Usage**

```
outputAnnotatedDataMatrix(eset, exptname, stage, dataElement)
```

**Arguments**

eset                    ExpressionSet containing the matrix and annotation to output  
 exptname                a string containing the name of the experiment. Used to build file name  
 stage                    a string containing the stage of the data in the matrix e.g. normalized  
 dataElement             a string containing the name of the data element in the ExpressionSet to be output

**Examples**

```
#Write out an annotated tab delimited file for the normalized data
## Not run:
outputAnnotatedDataMatrix(normedEset, "MyMicroArrayExpt", "Median_Normalized", "exprs")
## End(Not run)
```

---

panelCor *Add Pearson Correlation value to plots*

---

### Description

Adds a pearson correlation value to the scatter plots

### Usage

```
panelCor(x,y, digits=2, prefix="r=")
```

### Arguments

|        |   |
|--------|---|
| x      | matrix of x values  |
| y      | matrix of y values to correlate with x                            |
| digits | number of digits to display                                       |
| prefix | The string prefix that should be display on the scatterplot panel |

### Examples

```
#Calculate the median of a matrix omiting NAs
## Not run:
panelCor(x,y, digits=2, prefix="r=")

## End(Not run)
```

---

parseRawData *Parse raw data files to create an ExpressionSet*

---

### Description

Loads all the gpr scanner output files in a particular directory and returns an ExpressionSet of the hybridizations in a MiChip experiment

### Usage

```
parseRawData(datadir=".", pat="gpr")
```

### Arguments

|         |   |
|---------|---|
| datadir | a directory containing one or my files of scanner output from MiChip hybridizations |
| pat     | a string containing the three letter extension of the scanner output files          |

**Examples**

```
## Not run:
## Load all *.gpr files in current directory
parseRawData(datadir=".", pat="gpr")

## Load all *.gpr files in a specified directory , windows
parseRawData(datadir="c:\mydata\grpdata\expt1\", pat="gpr")

## Load all *.gpr files in a specified directory, linux
parseRawData(datadir="/home/myuser/gprdata/expt1/", pat="gpr")

## End(Not run)
```

---

plotIntensitiesScatter

*Plot pairwise intensity scatter*

---

**Description**

Creates a pairwise set of scatter plots from a data matrix and writes it out to file

**Usage**

```
plotIntensitiesScatter(dmat, controls=NULL, exptname, maintitle)
```

**Arguments**

|           |   |
|-----------|---|
| dmat      | matrix containing data from an experiment to be plotted                           |
| controls  | matrix of row numbers containing control data to be plotted in a different colour |
| exptname  | Name of the experiment, used for build the filename                               |
| maintitle | String used to build the maintitle of the graph                                   |

**Examples**

```
#Plot the pairwise intensities from myexpt
## Not run:
plotIntensitiesScatter(dmat, NULL, "MyExpt", "Median_Normalized")

## End(Not run)
```

---

removeUnwantedRows      *Removes unwanted rows from data matrix*

---

### Description

Due to the requirements of spotting the chips, some of the spots are empty. Others contain controls or features from another species that may not be wanted in the analysis. This method removes them

### Usage

```
removeUnwantedRows(rawData, filters)
```

### Arguments

rawData                  ExpressionSet containing matrix of data to be filtered  
 filters                  list of strings to be to be filtered from annotation gene name column

### Examples

```
#Removes empty and control spots from data matrix
## Not run:
filters=c("empty", "control")
filteredData <- removeUnwantedRows(rawData, filters)
## End(Not run)
```

---

returnAnnotatedDataMatrix  
                                  *returns and annotated data matrix from an ExpressionSet*

---

### Description

Takes an ExpressionSet and returns a data matrix of feature annotation to the left and hyb specific expression/flag data to the right

### Usage

```
returnAnnotatedDataMatrix(eset, dataElement)
```

### Arguments

eset                      ExpressionSet containing the matrix and annotation to output  
 dataElement            a string containing the name of the data element in the ExpressionSet to be output

### Examples

```
#Write out an annotated tab delimited file for the normalized data
## Not run:
returnAnnotatedDataMatrix(normedEset,"exprs")
## End(Not run)
```



---

setIntensityCutoff     *Sets a cutoff for the lowest intensity value*

---

**Description**

Any value less than the cutoff value will be set to NA. This allows near background intensity values to be excluded

**Usage**

```
setIntensityCutoff(dmat, intensityCutoff)
```

**Arguments**

dmat                    matrix of intensity values to which the cutoff value is applied  
intensityCutoff        value of lowest acceptable intensity value in the experiment

**Examples**

```
#Set all the values under 50 in a matrix to NA  
## Not run:  
dmatOver50 <- setIntensityCutoff(dmat, 50)  
  
## End(Not run)
```

---

standardRemoveRows     *Removes a standard list of features for MiChip processing*

---

**Description**

Removes all empty spots, control spots, U6 RNA, non human spots from an ExpressionSet in the standard fashion. A wrapper for removeUnwantedRows

**Usage**

```
standardRemoveRows(rawData)
```

**Arguments**

rawData                ExpressionSet containing the matrix to be filtered

**Examples**

```
#Filter standard rows from an ExpressionSet  
## Not run:  
myfilterdEset <-standardRemoveRows(rawData)  
  
## End(Not run)
```

---

```
summarizeIntensitiesAsMedian
```

*Summarizes the probe intensity as median of replicates spotted*

---

### Description

As the probes are spotted onto the in quaduplet or duplicate the values have to be combined in some way. This function takes the median of the intensities for the spots. Effectively the mean for duplicates. If less than half of the spots are present an NA is added

### Usage

```
summarizeIntensitiesAsMedian(eset,minSumlength=0, madAdjust=FALSE)
```

### Arguments

|              |   |
|--------------|---|
| eset         | ExpressionSet containing probe intensity data to be summarized  |
| minSumlength | The lowest acceptable length of the matrix to calculate a median  |
| madAdjust    | if TRUE then summarized data will be filtered according to the MAD median absolute deviation and set to NA if the median is less than MAD |

### Examples

```
#Calculate the median of a matrix omiting NAs
## Not run:
summarizeIntensitiesAsMedian(eset,minSumlength=0,madAdjust=TRUE)

## End(Not run)
```

---

```
workedExampleMedianNormalize
```

*Worked Example of MiChip Processing*

---

### Description

Loads a set of hybridizations into a matrix and them proceeds to filter, summarize and median normalize them

### Usage

```
workedExampleMedianNormalize(exptname, intensityCutoff=0, datadir=".", minSumlength, madAdjust = F
```

### Arguments

|                 |   |
|-----------------|---|
| exptname        | string indicating the name of the experiment  |
| intensityCutoff | The intensity value for accepting the spots intensity value in the experiment   |
| datadir         | The directory where hybridization files are found.  |
| minSumlength    | Minimum exceptable number of values to summarize intensity value.   |
| madAdjust       | if TRUE then summarized data will be filtered according to the MAD median absolute deviation and set to NA if the median is less than MAD |

**Examples**

```
#Normalize data in the current directory to the median per chip
datadir <- system.file("extdata", package="MiChip")
myNormedEset <-workedExampleMedianNormalize("MyExpt", intensityCutoff=0, datadir, minSumlength=0, madAdjust=
```

---

```
workedExampleNotNormalizedData
```

*Worked Example of MiChip Processing*

---

**Description**

Loads a set of hybridizations into a matrix and then proceeds to filter and summarize these data

**Usage**

```
workedExampleNotNormalizedData(exptname, intensityCutoff=0, datadir=".", minSumlength, madAdjust =
```

**Arguments**

|                 |   |
|-----------------|---|
| exptname        | string indicating the name of the experiment  |
| intensityCutoff | The intensity value for accepting the spots intensity value in the experiment   |
| datadir         | The directory contain data from the experiment  |
| minSumlength    | Minimum exceptable number of values to summarize intensity value.   |
| madAdjust       | if TRUE then summarized data will be filtered according to the MAD median absolute deviation and set to NA if the median is less than MAD |

**Examples**

```
#Summarizes the data in the current directory
## Not run:
mySummarizedEset <-workedExampleNotNormalizedData("MyExpt", intensityCutoff=0, datadir=".", minSumlength=0,
## End(Not run)
```

# Index

- \* **aplot**
    - boxplotData, [2](#)
    - boxplotDataNoFile, [2](#)
    - panelCor, [6](#)
  - \* **datagen**
    - myForgivingMedian, [4](#)
    - naOmitMedian, [4](#)
    - summarizeIntensitiesAsMedian, [10](#)
  - \* **data**
    - correctForFlags, [3](#)
    - removeUnwantedRows, [8](#)
    - setIntensityCutoff, [9](#)
    - standardRemoveRows, [9](#)
  - \* **design**
    - workedExampleMedianNormalize, [10](#)
    - workedExampleNotNormalizedData, [11](#)
  - \* **documentation**
    - MiChip, [3](#)
  - \* **file**
    - outputAnnotatedDataMatrix, [5](#)
    - parseRawData, [6](#)
    - returnAnnotatedDataMatrix, [8](#)
  - \* **hplot**
    - plotIntensitiesScatter, [7](#)
  - \* **models**
    - normalizePerChipMedian, [5](#)
- boxplotData, [2](#)  
boxplotDataNoFile, [2](#)
- correctForFlags, [3](#)
- Introduction (MiChip), [3](#)
- MiChip, [3](#)  
myForgivingMedian, [4](#)
- naOmitMedian, [4](#)  
normalizePerChipMedian, [5](#)
- outputAnnotatedDataMatrix, [5](#)
- panelCor, [6](#)  
parseRawData, [6](#)  
plotIntensitiesScatter, [7](#)
- removeUnwantedRows, [8](#)  
returnAnnotatedDataMatrix, [8](#)
- setIntensityCutoff, [9](#)  
standardRemoveRows, [9](#)  
summarizeIntensitiesAsMedian, [10](#)
- workedExampleMedianNormalize, [10](#)  
workedExampleNotNormalizedData, [11](#)