

# Package ‘beadarrayFilter’

February 19, 2015

**Type** Package

**Title** Bead filtering for Illumina bead arrays

**Version** 1.1.0

**Date** 2013-02-04

**Author** Anyiawung Chiara Forcheh, Geert Verbeke, Adetayo Kasim, Dan Lin, Ziv Shkedy, Willem Talloen, Hinrich WH Gohlmann, Lieven Clement.

**Maintainer** Lieven Clement <lieven.clement@gmail.com>

**Description** This package contains functions to fit the filtering model of Forcheh et al., (2012) which is used to derive the intra-cluster correlation (ICC). Model fitting is done using the modified version of the “MLM.beadarray” function of Kim and Lin (2011).

**Depends** beadarray, RColorBrewer

**Suggests** beadarrayExampleData, lumi

**License** GPL-2

**Repository** CRAN

**Date/Publication** 2013-02-07 20:25:27

**NeedsCompilation** no

## R topics documented:

beadtypeFilter . . . . .	2
emCDF . . . . .	3
iccFun . . . . .	4
MLM.beadarray . . . . .	6
varianceplot . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

beadtypeFilter

*Filter Illumina bead arrays*


---

### Description

This is a wrapper function for the `iccFun` function and it implements the filtering method of Forchheh et al. (2012). It takes a normalized `ExpressionSetIllumina` object, a `LumiBatch` object or a normalized `data.frame` and returns the names of the informative bead types. Optionally, the filtered `ExpressionSetIllumina` object or the filtered `data.frame` can also be returned.

### Usage

```
beadtypeFilter(beadsum, Quantile = 1, keepData = TRUE, delta = 0.5)
```

### Arguments

<code>beadsum</code>	A normalized data object, which can be an <code>ExpressionSetIllumina</code> , <code>LumiBatch</code> or a normalized <code>data.frame</code> .
<code>Quantile</code>	It refers to the number summary described in Forchheh et al. (2012). It is the quantile at which summarization of the ICC is done. For example, <code>iccQuant = 0</code> means that summarization of the ICCs be done at the minimum ICC, <code>0.75</code> means summarization be done at the 75 quantile. Note that filtering will also be done at this specified quantile. For example, <code>iccQuant = 1</code> means that the ICCs be summarized only using the maximum ICC and that beads be filtered using the maximum ICC.
<code>keepData</code>	If <code>TRUE</code> , then, the filtered object (i.e., filtered <code>ExpressionSetIllumina</code> object or filtered <code>data.frame</code> ) and the identification names of the informative bead types are returned. If <code>FALSE</code> , only the identification names of the informative bead types will be returned.
<code>delta</code>	The proposed cutoff point at which a bead type should be declared informative or non-informative. The default is <code>0.5</code> .

### Details

Bead filtering aims at removing non-informative bead types prior to the downstream analysis.

This function is a wrapper of the `iccFun` function and it performs bead filtering. It takes a normalized `ExpressionSetIllumina` object or a normalized `data.frame` as input and so, the user should do the appropriate data normalization prior to submitting the `ExpressionSetIllumina` object or `data.frame` to the function. Refer to the “beadarray” package documentation for more on how to obtain a normalized `ExpressionSetIllumina` object. If a `data.frame` is used, make sure that the column names are similarly to the ones from `BeadStudio`: column names for bead averages should end with “Signal”, standard errors should end with “STDERR” and number of beads used for the data summarization should end with “NBEADS”.

**Value**

InformProbeNames  
Names of the Informative bead types

informData  
The filtered ExpressionSetIllumina object or filtered data.frame

**Author(s)**

Anyiawung Chiara Forcheh, Geert Verbeke, Adetayo Kasim, Dan Lin, Ziv Shkedy, Willem Talloen, Hinrich WH Gohlmann, Lieven Clement.

**References**

Dunning, M.J., Smith, M.L., Ritchie, M.E., Tavare, S. (2007). beadarray: R classes and methods for Illumina bead-based data, *Bioinformatics*. 23(16):2183-2184.

Dunning, M.J. (2011). beadarrayExampleData: Example data for the beadarray package. R package version 1.0.4.

Forcheh, A.C, Verbeke, G., Kasim, A., Lin, D., Shkedy, Z., Talloen, W., Gohlmann, H.W.H, Clement, L. (2012). Gene Filtering in the Analysis of Illumina Microarray Experiments, *Statistical Applications in Genetics and Molecular Biology*, Berkeley Electronic Press, vol. 11(2), Article 3.

Kim, R.S. and Lin, J. (2011). Multi-level mixed effects models for bead arrays. *Bioinformatics*, 27(5):633-640.

**Examples**

```
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDataNorm <- normaliseIllumina(channel(exampleSummaryData, "G"),
method = "quantile", transform = "none")
iccResults<-beadtypeFilter(exampleSummaryDataNorm[1:100,],Quantile=1,keepData=FALSE,delta=0.5)
```

---

emCDF

*Function to plot the empirical cumulative density functions (ecdf)*


---

**Description**

Plots the ecdf for different thresholding strategies as discussed in Forcheh et al. (2012).

**Usage**

```
emCDF(iccResults, iccQuant = 1)
```

**Arguments**

- `iccResults` A list containing a number of components obtained from applying the `iccFun` function. It containing the ICCs to be used for plotting.
- `iccQuant` Refers to the number summary described in Forcheh et al. (2012). It can be a vector of quantiles or a single value. For example, `iccQuant = c(0,0.25,0.5,0.75,1)` means that the ecdf be plotted for the minimum ICC, 0.25, 0.5, 0.75 for the 25, 50, and 75 quantiles and the maximum ICC respectively, while `iccQuant = 1` means that the ecdf be plotted only for one number summary of interest.

**Author(s)**

Anyiawung Chiara Forcheh, Geert Verbeke, Adetayo Kasim, Dan Lin, Ziv Shkedy, Willem Talloen, Hinrich WH Gohlmann, Lieven Clement.

**References**

Neuwirth, E. RColorBrewer: ColorBrewer palettes. R package version 1.0-2., 2007.

**Examples**

```
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDataLog2 <- channel(exampleSummaryData, "G")
exampleSummaryDataNorm <- normaliseIllumina(exampleSummaryDataLog2,
method = "quantile", transform = "none")
aaa<-na.omit(data.frame(I(rownames(exprs(exampleSummaryDataNorm))),exprs(exampleSummaryDataNorm)))
ProbeID <- aaa[,1][1:100]
eSet <- na.omit(exprs(exampleSummaryDataNorm))[1:100,]
stddev <- na.omit(se.exprs(exampleSummaryDataNorm))[1:100,]
nSet <- na.omit(attributes(exampleSummaryDataNorm)$assayData$nObservations)[1:100,]
seSet<-stddev/sqrt(nSet)
iccResults<-iccFun(eSet,seSet,nSet,ProbeID =ProbeID ,iccQuant=c(0,0.25,0.5,1),
diffIcc=TRUE,keepData=TRUE)
emCDF(iccResults, iccQuant= c(0,0.25,0.5,1))
```

---

 iccFun

---

*Intra-cluster correlations (ICC) for Illumina bead arrays*


---

**Description**

Fits the filtering model of Forcheh et al. (2012) and derives the ICC that can be used for filtering. Fitting is done using a modified version of the "MLM.beadarray" function of Kim and Lin (2011).

**Usage**

```
iccFun(eSet, seSet, nSet, ProbeID, iccQuant, diffIcc = TRUE, keepData = TRUE)
```

**Arguments**

eSet	The summarized and normalized average intensities
seSet	The standard errors of the average intensities
nSet	The number of beads used for summarization
ProbeID	The bead types identification variable e.g, ProbeID
iccQuant	It refers to the number summary described in Forcheh et al. (2012). It can be a vector of quantiles or a single value (scalar). For example, <code>iccQuant = c(0, 0.25, 0.5, 0.75, 1)</code> means that summarization of the ICCs be done at the minimum ICC, 0.25, 0.5, 0.75 for the 25, 50, and 75 quantiles and the maximum ICC respectively. <code>iccQuant = 1</code> means that the ICCs be summarized only using the maximum ICC.
diffIcc	If TRUE, then <code>iccQuant</code> must be a vector, if FALSE, <code>iccQuant</code> must be a single value. The default is TRUE.
keepData	If TRUE, then, the ICC at the number summary, the within array variances, the between array variances as well as all the ICCs will be returned. If FALSE, only the ICCs at the number summary will be returned.

**Details**

Bead filtering aims at removing non-informative bead types prior to the downstream analysis. This function takes as input the summarized and normalized average intensities (`eSet`), their standard errors (`seSet`) and the number of beads used for summarization (`nSet`). It fits the filtering model of Forcheh et al. (2012) and derives the ICCs. The obtained ICC can then be used for filtering bead types or to assess the filtering strategies. See examples for more on this.

**Value**

<code>icc</code>	The summarized ICCs at the number summary
<code>withinvar</code>	The within array variances
<code>betweenvar</code>	The between array variances
<code>iccAll</code>	All the ICCs, i.e., one ICC per bead type per array

**Author(s)**

Anyiauwung Chiara Forcheh, Geert Verbeke, Adetayo Kasim, Dan Lin, Ziv Shkedy, Willem Talloen, Hinrich WH Gohlmann, Lieven Clement.

**References**

- Dunning, M.J., Smith, M.L., Ritchie, M.E., Tavaré, S. (2007). `beadarray`: R classes and methods for Illumina bead-based data, *Bioinformatics*. 23(16):2183-2184.
- Dunning, M.J. (2011). `beadarrayExampleData`: Example data for the `beadarray` package. R package version 1.0.4.
- Forcheh, A.C, Verbeke, G., Kasim, A., Lin, D., Shkedy, Z., Talloen, W., Gohlmann, H.W.H, Clement, L. (2012). Gene Filtering in the Analysis of Illumina Microarray Experiments, *Statistical Applications in Genetics and Molecular Biology*, Berkeley Electronic Press, vol. 11(2), Article 3.

Kim, R.S. and Lin, J. (2011). Multi-level mixed effects models for bead arrays. *Bioinformatics*, 27(5):633-640.

### Examples

```
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDataLog2 <- channel(exampleSummaryData, "G")[1:40,]
exampleSummaryDataNorm <- normaliseIllumina(exampleSummaryDataLog2,
method = "quantile", transform = "none")[1:40,]
aaa<-na.omit(data.frame(I(rownames(exprs(exampleSummaryDataNorm))),exprs(exampleSummaryDataNorm)))[1:40,]
ProbeID <- aaa[,1][1:40]
  eSet <- na.omit(exprs(exampleSummaryDataNorm))[1:40,]
  stddev <- na.omit(se.exprs(exampleSummaryDataNorm))[1:40,]
  nSet <- na.omit(attributes(exampleSummaryDataNorm)$assayData$nObservations)[1:40,]
  seSet<-stddev/sqrt(nSet)
  iccResults<-iccFun(eSet,seSet,nSet,ProbeID =ProbeID ,iccQuant=1,
diffIcc=FALSE,keepData=TRUE)

  informProbeNames<- subset(iccResults$icc, iccResults$icc[,2]>=0.5)[,1]
  filterDataNorm<-exampleSummaryDataNorm[informProbeNames,]
```

---

MLM.beadarray

*Multi-level Mixed model*

---

### Description

Function for differential expression analysis of bead array data using the Multi-level Mixed model of Kim and Lin (2011).

### Usage

```
MLM.beadarray(sig, stderr, nbeads, groups, var.equal = FALSE, max.iteration = 20,
epsilon = 1e-06, method = "REML")
```

### Arguments

sig	The summarized and normalized average intensities
stderr	The standard errors of the means
nbeads	The number of beads used for summarization
groups	This refers to the groups to be compared. For filtering, group refers to the number of cloumns of sig. However, for the differential expression analysis, the user should define the group variable as appropriate for his/her data.
var.equal	i.e assuming equal variance for the variance of the array random effects
max.iteration	The maximum number of iteration to perform
epsilon	control limit for convergence
method	Allows one to choose between restricted maximum likelihood (REML) or maximum likelihood (ML) estimations

**Author(s)**

Ryung S. Kim and Juan Lin

**References**

Kim, R.S. and Lin, J. (2011). Multi-level mixed effects models for bead arrays. *Bioinformatics*, 27(5):633-640.

**Examples**

```
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDataLog2 <- channel(exampleSummaryData, "G")[1:40,]

exampleSummaryDataNorm <- normaliseIllumina(exampleSummaryDataLog2,
method = "quantile", transform = "none")[1:40,]

eSet <- na.omit(exprs(exampleSummaryDataNorm))[1:40,]

seSet <- na.omit(se.exprs(exampleSummaryDataNorm))[1:40,]

nSet <- na.omit(attributes(exampleSummaryDataNorm)$assayData$nObservations)[1:40,]

stderrs<-seSet/sqrt(nSet)

##define group variable as appropriate for your data
group1 <- c(1:6)
group2 <- c(7:12)
fit1 <- MLM.beadarray(eSet, stderrs, nSet, list(group1,group2), var.equal = TRUE,
max.iteration = 20, method = "ML")

df<-length(group1)+length(group2)-2
fit1$pvalue<-2*(1-pt(abs(fit1$t.statistics),df))
fit1$PvalADjust<-p.adjust(fit1$pvalue, method ="fdr", n = length(fit1$pvalue))
length(which(fit1$PvalADjust<0.05))
```

---

varianceplot

---

*Observe between-array and within-array variances.*


---

**Description**

plots the between array and the within array variances.

**Usage**

```
varianceplot(iccResults, q = 1, delta = 0.5)
```

**Arguments**

<code>iccResults</code>	Output from the <code>iccFun</code> function.
<code>q</code>	The quantile use for ICC at which one wants to declare bead types informative. If <code>iccQuant</code> is a vector, then <code>q</code> can take any value from 1 up to the length of <code>iccQuant</code> .
<code>delta</code>	The proposed cutoff point at which a bead type should be declared informative or non-informative.

**Author(s)**

Anyiawung Chiara Forcheh, Geert Verbeke, Adetayo Kasim, Dan Lin, Ziv Shkedy, Willem Talloen, Hinrich WH Gohlmann, Lieven Clement.

**Examples**

```
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDataLog2 <- channel(exampleSummaryData, "G")
exampleSummaryDataNorm <- normaliseIllumina(exampleSummaryDataLog2,
method = "quantile", transform = "none")
aaa<-na.omit(data.frame(I(rownames(exprs(exampleSummaryDataNorm))),exprs(exampleSummaryDataNorm)))
ProbeID <- aaa[,1][1:100]
  eSet <- na.omit(exprs(exampleSummaryDataNorm))[1:100,]
  stddev <- na.omit(se.exprs(exampleSummaryDataNorm))[1:100,]
  nSet <- na.omit(attributes(exampleSummaryDataNorm)$assayData$nObservations)[1:100,]
  seSet<-stddev/sqrt(nSet)
  iccResults<-iccFun(eSet,seSet,nSet,ProbeID =ProbeID ,iccQuant=c(0,0.25,0.5,1),
  diffIcc=TRUE,keepData=TRUE)
  varianceplot(iccResults,q=1,delta=0.8)
```



# Index

- \*Topic **Bead types**
  - beadtypeFilter, 2
- \*Topic **Filter**
  - beadtypeFilter, 2
- \*Topic **ICC**
  - iccFun, 4
- \*Topic **Plot**
  - emCDF, 3
  - varianceplot, 7
- \*Topic **Variances**
  - varianceplot, 7
- \*Topic **differential**
  - MLM.beadarray, 6
- \*Topic **ecdf**
  - emCDF, 3
- \*Topic **expression**
  - MLM.beadarray, 6

beadtypeFilter, 2

emCDF, 3

iccFun, 4

MLM.beadarray, 6

varianceplot, 7