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
dmCDF ................................................................. 3
iccFun ................................................................. 4
MLM.beadarray ............................................................. 6
varianceplot .............................................................. 7

Index 9
beadtypeFilter  

Filter Illumina bead arrays

Description

This is a wrapper function for the iccFun function and it implements the filtering method of Forcheh 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

beadsum  
A normalized data object, which can be an ExpressionSetIllumina, LumiBatch or a normalized data.frame.

quantile  
It refers to the number summary described in Forcheh et al. (2012). It is the quantile at which summarization of the ICC is done. For example, iccQuant = 0 means that summarization of the ICCs be done at the minimum ICC, 0.75 means summarization be done at the 75 quantile. Note that filtering will also be done at this specified quantile. For example, iccQuant = 1 means that the ICCs be summarized only using the maximum ICC and that beads be filtered using the maximum ICC.

keepdata  
If TRUE, then, the filtered object (i.e., filtered ExpressionSetIllumina object or filtered data.frame) and the identification names of the informative bead types are returned. If FALSE, only the identification names of the informative bead types will be returned.

delta  
The proposed cutoff point at which a bead type should be declared informative or non-informative. The default is 0.5.

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)


References


Examples

```r
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDataNorm <- normaliseIllumina(channel(exampleSummaryData, "G"),
method = "quantile", transform = "none")
iccResults <- beadtypeFilter(exampleSummaryDataNorm[1:100,,Quartile=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

```r
emCDF(iccResults, iccQuant = 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

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)


References


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)$assayData$Observations)[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))
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, iccQuant = c(0, 0.25, 0.5, 0.75, 1) 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. iccQuant = 1 means that the ICCs be summarized only using the maximum ICC.
diffIcc If TRUE, then iccQuant must be a vector, if FALSE, iccQuant 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

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

Author(s)


References


Examples

```r
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDataLog2 <- channel(exampleSummaryData, "G"[1:40,])
exampleSummaryDataNorm <- normaliseIllumina(exampleSummaryDataLog2,
method = "quantile", transform = "none")[1:40,]
aaa <- na.omit(data.frame(rownames(exprs(exampleSummaryDataNorm)),
exprs(exampleSummaryDataNorm))[1:40,])
ProbElID <- aac[,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, ProbElID = ProbElID, iccQuant = 1,
diffIc = FALSE, keepData = TRUE)

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

MLM.beadarray  
Multi-level Mixed model

Description


Usage

```r
MLM.beadarray(sig, stderr, nbads, 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
- **nbads**: The number of beads used for summarization
- **groups**: This refers to the groups to be compared. For filtering, group refers to the number of columns 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
varianceplot

Author(s)

Ryung S. Kim and Juan Lin

References


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$pvalue <- 2*(1-pt(abs(fit1$st.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

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

Author(s)


Examples

```r
require(beadarrayExampleData)
data(exampleSummaryData)
exampleSummaryDatalog2 <- channel(exampleSummaryData, "G")
exampleSummaryDataNorm <- normaliseIllumina(exampleSummaryDatalog2,
                                           method = "quantile", transform = "none")
aaa<-na.omit(data.frame(I(rownames(exprs(exampleSummaryDataNorm)))[,1]),exprs(exampleSummaryDataNorm))
Probes <- a[,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,eSet,nSet,Probes,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