How to analyze EPIC microarray data with MetKMR

Ruth Barral Arca

28 de junio de 2019

Set the enviroment

```
library(minfi)
library(MetKMR)
library(IlluminaHumanMethylationEPICanno.ilm10b4.hg19)
library(doParallel)
detectCores()
```

```
## [1] 8
```

```
registerDoParallel(cores = 8) #this analysis was also run smoothly with 4 cores
```

Prepare the data

Click here to download the raw data and uncompress it Create the Targets.csv file e.g.:

Sample_Name	Sample_Well	Sample_Plate	Sample_Group	Array	Basename
GSM2883348_G1	A02	Test_Plate	disease	R01C01	GSM2883348_G1
GSM2883350_G2	B02	Test_Plate	disease	R02C01	GSM2883350_G2
GSM2883353_G3	C02	Test_Plate	normal	R03C01	GSM2883353_G3
GSM2883351_G4	D02	Test_Plate	normal	R04C01	GSM2883351_G4
GSM2883352_G5	E02	Test_Plate	disease	R05C01	GSM2883352_G5
GSM2883349_G6	F02	Test_Plate	normal	R06C01	GSM2883349_G6

load the data

```
setwd("/home/ruth/Descargas/GSE107917_RAW")
idat.folder <- "/home/ruth/Descargas/GSE107917_RAW"
targets <- read.metharray.sheet(base=idat.folder)
```

```
## [1] "/home/ruth/Descargas/GSE107917_RAW/targets.csv"
```

```
targets$Basename<-targets$Sample_Name
###loading data
rgset <- read.metharray.exp(targets = targets)
#save(rgset,file="rgset.rda" )
#load("rgset.rda")
phenoData <- rgset$Sample_Group
```

Preprocess and normalize the data

The RGChannelSet stores also a manifest object that contains the probe design information of the array:

```
manifest <- getManifest(rgset)
```

A MethylSet objects contains only the methylated and unmethylated signals

```
MSet <- preprocessRaw(rgset)
```

A RatioSet object is a class designed to store Beta values and/or M values instead of the methylated and unmethylated signals. An optional copy number matrix, CN, the sum of the methylated and unmethylated signals, can be also stored. Mapping a MethylSet to a RatioSet may be irreversible, i.e. one cannot be guranteed to retrieve the methylated and unmethylated signals from a RatioSet. A RatioSet can be created with the function ratioConvert:

```
RSet <- ratioConvert(MSet, what = "both", keepCN = TRUE)
```

The functions getBeta, getM and getCN return respectively the Beta value matrix, M value matrix and the Copy Number matrix.

```
beta <- getBeta(RSet)
M <- getM(MSet)
```

The function mapToGenome applied to a RatioSet object will add genomic coordinates to each probe together with some additional annotation information. The output object is a GenomicRatioSet (class holding M or/and Beta values together with associated genomic coordinates). It is possible to merge the manifest object with the genomic locations by setting the option mergeManifest to TRUE.

```
GRset <- mapToGenome(RSet)
beta <- getBeta(GRset)
M <- getM(GRset)
CN <- getCN(GRset)
sampleNames <- sampleNames(GRset)
probeNames <- featureNames(GRset)
gr <- granges(GRset)
head(gr, n= 3)
```

```
## GRanges object with 3 ranges and 0 metadata columns:
##                seqnames    ranges strand
##                   <Rle> <IRanges>  <Rle>
##   cg14817997     chr1     10525      *
##   cg26928153     chr1     10848      *
##   cg16269199     chr1     10850      *
##   -------
##   seqinfo: 24 sequences from hg19 genome; no seqlengths
```

Annotation

```
annotation <- getAnnotation(GRset)
#names(annotation)
###Normalization
gRatioSet.quantile <- preprocessQuantile(rgset)##SQN
#####anotation

annEPIC<-getAnnotation(IlluminaHumanMethylationEPICanno.ilm10b4.hg19)
#head(annEPIC)
```

Remove probes with SNPs at CpG or SBE site

```
gRatioSet.quantile<- dropLociWithSnps(gRatioSet.quantile)
betas<-getBeta(gRatioSet.quantile)
M <- getM(gRatioSet.quantile)
```

We will pay attention only of CpGs that are in promoter and gene body

```
annEPIC <-annEPIC [grep("TSS1500|TSS200|5'UTR|1stExon|Body",annEPIC$UCSC_RefGene_Group),]
betas<-betas[rownames(betas) %in% rownames(annEPIC) ,]
annEPIC <-annEPIC[rownames(betas) %in% rownames(annEPIC) ,]
annotation2 <- data.frame(row = 1:length(annEPIC$UCSC_RefGene_Name),
                          pos = annEPIC$pos,
                          site=rownames(annEPIC),
                          chr=annEPIC$chr,
                          gene = annEPIC$UCSC_RefGene_Name,
                          stringsAsFactors = F)
```

MetKMR Differentially Methylated Region Analysis

```
analysis <- new("MetRKAT",
                data = betas,
                annotation = annotation2,
                distmethod =  c("euclidean"),
                wsize = 9, gap = 0, #adding a gap increasses the time needed
                max.na = 0.3,wmethod = "default")
```

```
## Discarding/imputing NA values... Done!
## Preparing annotation dataset... Done!
```

```
analysis <- toSQLite(analysis, "tuberculosis.sqlite")
analysis@intervals <- createIntervals(analysis)
y<-replace(phenoData,phenoData=="disease",1)
y<-replace(y,phenoData=="normal",0)
analysis@results <- applyRKAT(analysis, y = y)
```

Results

```
results_df <- as.data.frame(analysis@results)
#we are interested in the significant results
filtered_results <- results_df[results_df$pval<= 0.05, ]
#other them by pval
filtered_results <- filtered_results [order(filtered_results$pval), ]
head(filtered_results)
```

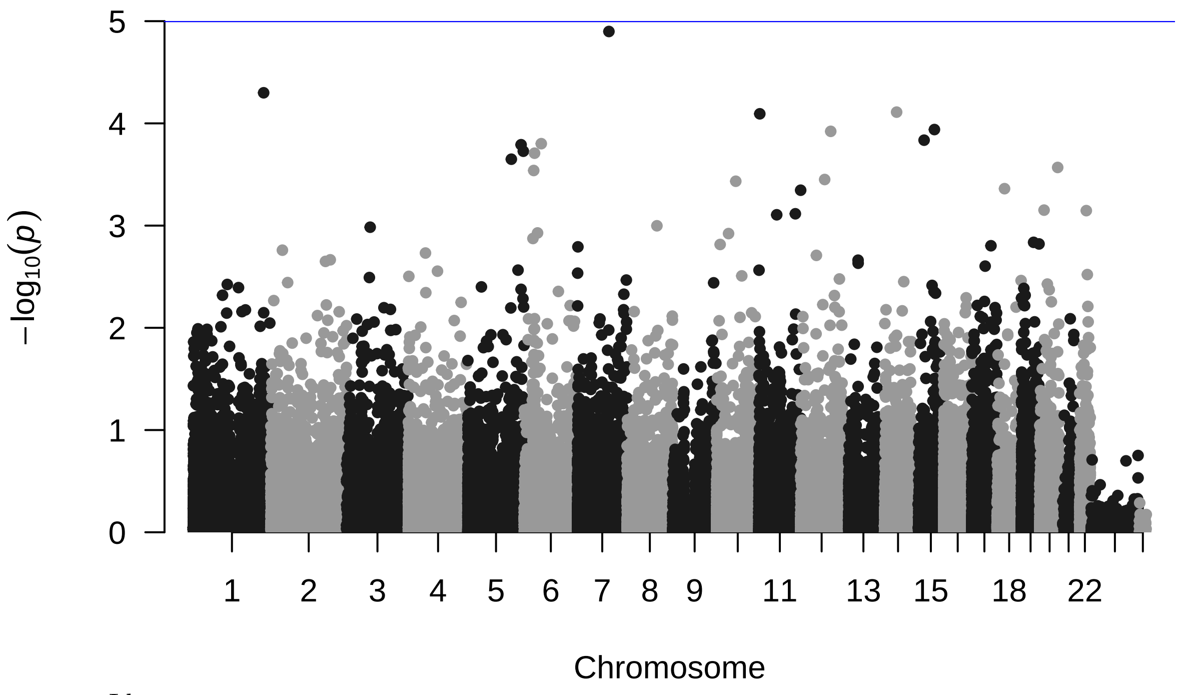
```
##       first_row last_row     start       end   chr         pval    kernel
## 56957    512522   512530 100886961 100888485  chr7 1.261927e-05 euclidean
## 5736      51615    51623 226250207 226251003  chr1 5.021111e-05 euclidean
## 18624    167584   167592  58619030  58619480 chr14 7.762606e-05 euclidean
## 9717      87433    87441   3013777   3014001 chr11 8.061431e-05 euclidean
## 21355    192160   192168  72978625  72979048 chr15 1.149596e-04 euclidean
## 15264    137355   137363  96184787  96251548 chr12 1.196753e-04 euclidean
##       omnibus
## 56957      NA
## 5736       NA
## 18624      NA
## 9717       NA
## 21355      NA
## 15264      NA
```

```
dim(filtered_results)
```

```
## [1] 899   8
```

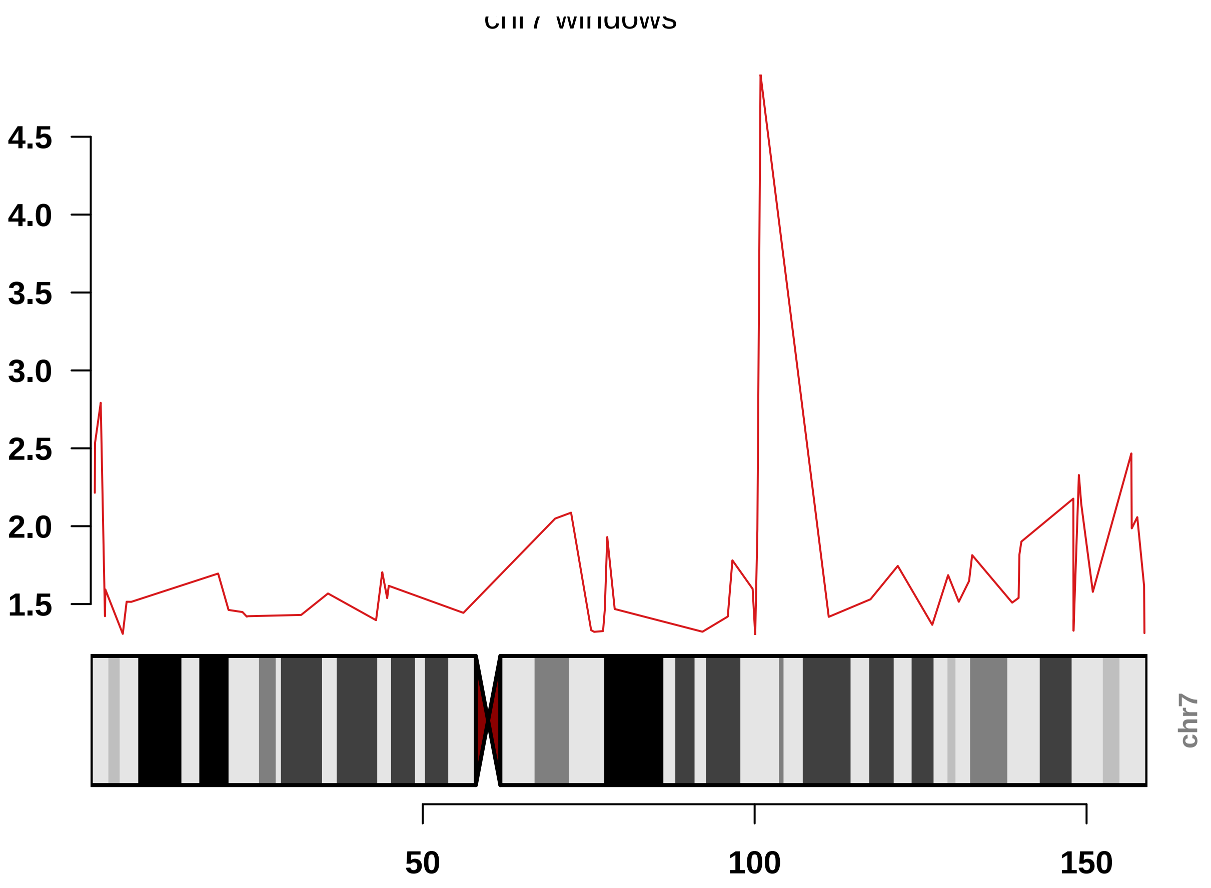
#Plots Manhattan plot

```
plotManhattan(analysis, pvals ='euclidean')
```



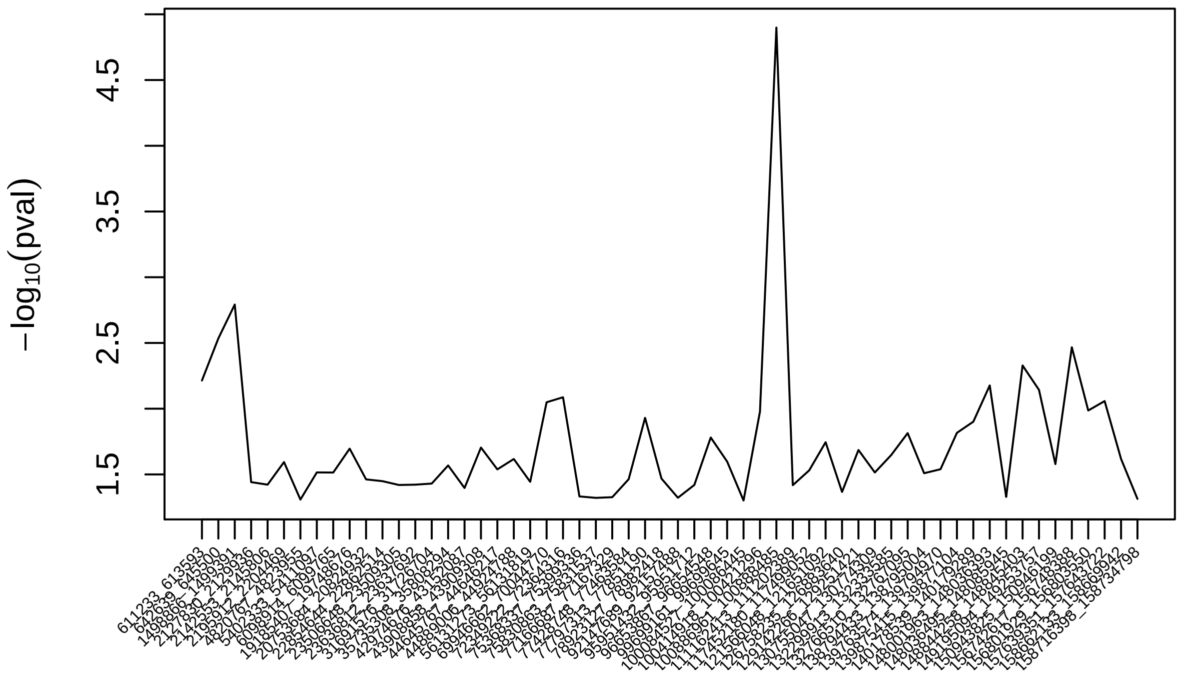
Chromosome Ideogram

```
library(rtracklayer)
plotChromosome(analysis, chrom = 'chr7', pvals = 'euclidean',cutoff = 0.05)
```



Windows plot

```r
plotWindows(analysis, chrom = 'chr7', pvals = 'euclidean')
```


−log10(pval)