TOPMed CAMP DNA methylation data QC and sample cleaning

reprk: PKachroo
CDNM, BWH

May 9, 2021

Contents

1 Setup

```
# restart R session
#.rs.restartR()
rm(list=ls())

options(mc.cores=5)
system("hostname")
print(Sys.Date())

## [1] "2021-05-08"

print(Sys.time())

## [1] "2021-05-08 15:21:29 EDT"

# To generate document:
# Change working directory to code directory
# Run this code on toques using:
# module load R/4.0.3
#R -e 'library(knitr);knit("TOPMed_CAMP_DNAm_processing.Rnw")'
# pdflatex TOPMed_CAMP_DNAm_processing.tex

# merging with WGS, uses hg38?
## load libraries
libs <- c("IlluminaHumanMethylationEPICanno.ilm10b4.hg19",
          "IlluminaHumanMethylationEPICmanifest", "minfi")

for (l in libs) {
  if (require(l, character.only = T)) {
    print(paste0(l, " loaded successfully"))
  } else {
    install.packages(l)
    require(l, character.only = T)
    print(paste0(l, " installed and loaded successfully"))
  }
}

## [1] "IlluminaHumanMethylationEPICanno.ilm10b4.hg19 loaded successfully"
## [1] "IlluminaHumanMethylationEPICmanifest loaded successfully"
## [1] "minfi loaded successfully"

sig_digits <- 2
sum_sd <- function(data, varname) {
  eval(parse(text = str_c("data[, round(summary(", varname, "), digits=2)] %>% print()")))
  eval(parse(text = str_c("print(str_c('SD: ', data[, sd(", varname, ", na.rm = T) %>%
                             round(sig_digits)]))")))
}
```

1.1 Packages, Data locations and loading

```
qc.dir = "/proj/regeps/regep00/studies/CAMP"
camp.dir = file.path(qc.dir,"data/epigenetic/methylation/TopMed/data/freezes/20200117")
RGSet.camp = readRDS(file=file.path(camp.dir, "LEVEL2/RGSet")) # 1616 samples
dim(RGSet.camp) # 1008711

## [1] 1008711    1616

RGSet.camp
```

```
## class: RGChannelSet
## dim: 1008711 1616
## metadata(0):
## assays(2): Green Red
## rownames(1008711): 1600101 1600111 ... 99810978 99810992
## rowData names(0):
## colnames(1616): TOE654293-BIS-v01_R04C01 TOE309577-BIS-v01_R04C01 ...
##   TOE290775-BIS-v01_R01C01 TOE402155-BIS-v01_R07C01
## colData names(7): Basename S_SAMPLEID ... S_STUDYID filenames
## Annotation
##   array: IlluminaHumanMethylationEPIC
##   annotation: ilm10b4.hg19
```

```
manifest = getManifest(RGSet.camp)
manifest
```

```
## IlluminaMethylationManifest object
## Annotation
##   array: IlluminaHumanMethylationEPIC
## Number of type I probes: 142262
## Number of type II probes: 724574
## Number of control probes: 635
## Number of SNP type I probes: 21
## Number of SNP type II probes: 38
```

```
data(IlluminaHumanMethylationEPICanno.ilm10b4.hg19)
data("Manifest")
table(Manifest$Type)
```

```
##
##      I     II
## 142137 723722
```

```
length(grep("^cg.", rownames(Manifest), value=TRUE)) # 862927 CG probes
```

```
## [1] 862927
```

```
length(grep("^ch.", rownames(Manifest), value=TRUE)) # 2932 CH probes
```

```
## [1] 2932
```

```
length(grep("^rs.", rownames(Manifest), value=TRUE)) # 0
```

```
## [1] 0
```

```
# Downloaded Illumina manifest file
festV1 <- read.csv("/proj/rerefs/reref00/Illumina/MethylationEPIC-v1-0-B4/lib/MethylationEPIC_v-1-0_B
                   skip=7,as.is=TRUE, sep=",", stringsAsFactors=FALSE)

# loading rest of the libraries
libs <- c("limma", "wateRmelon", "minfi", "gplots", "ggplot2", "knitr", "R.utils", "impute",
          "stats", "tidyverse", "data.table", "here", "e1071", "GGally", "ggrepel", "ENmix",
          "meffil", "data.table", "robustbase", "stringi", "geneplotter", "RColorBrewer",
          "colorRamps", "lumi", "ggrepel")

for (l in libs) {
  if (require(l, character.only = T)) {
    print(paste0(l, " loaded successfully"))
  } else {
    install.packages(l)
    require(l, character.only = T)
    print(paste0(l, " installed and loaded successfully"))
  }
}
```

```
## [1] "limma loaded successfully"
## [1] "wateRmelon loaded successfully"
## [1] "minfi loaded successfully"
## [1] "gplots loaded successfully"
## [1] "ggplot2 loaded successfully"
## [1] "knitr loaded successfully"
## [1] "R.utils loaded successfully"
## [1] "impute loaded successfully"
## [1] "stats loaded successfully"
## [1] "tidyverse loaded successfully"
## [1] "data.table loaded successfully"
## [1] "here loaded successfully"
## [1] "e1071 loaded successfully"
## [1] "GGally loaded successfully"
## [1] "ggrepel loaded successfully"
## [1] "ENmix loaded successfully"
## [1] "meffil loaded successfully"
## [1] "data.table loaded successfully"
## [1] "robustbase loaded successfully"
## [1] "stringi loaded successfully"
## [1] "geneplotter loaded successfully"
## [1] "RColorBrewer loaded successfully"
## [1] "colorRamps loaded successfully"
## [1] "lumi loaded successfully"
## [1] "ggrepel loaded successfully"
```

```
plots.dir = file.path(qc.dir,"analyses/reprk/methylation/plots")
results.dir = file.path(plots.dir,"../results")
meff.dir = file.path(qc.dir,"analyses/reprk/meffil_850K")

# modified RCP code
source("/udd/reprk/projects/TOPMed/scripts/RCP_mod.R")

pca.betas <- function (beta, npc = 50)
{
    if (!is.matrix(beta)) {
        stop("beta is not a data matirx")
    }
    cat("Analysis is running, please wait...!", "\n")
    npc <- min(ncol(beta), npc)
    svd <- prcomp(t(beta), center = TRUE, scale = TRUE, retx = TRUE)
    eigenvalue <- svd[["sdev"]]^2
    prop <- (sum(eigenvalue[1:npc])/sum(eigenvalue)) * 100
    cat("Top ", npc, " principal components can explain ", prop,
        "% of data \n    variation", "\n")
    save(svd, eigenvalue, prop, file=file.path(results.dir,"pca_betas_auto.RData"))
}

setwd("/udd/reprk/projects/TOPMed/scripts")
camp.pheno <- read.csv(file=file.path(qc.dir, "data/phenotype/camp_pheno_0421.csv"),
                       as.is=TRUE, sep=",", stringsAsFactors=FALSE)

samplesheet.camp <- read.csv(file=file.path(camp.dir, "LEVEL1/SampleSheet.csv"),
                             as.is=TRUE, sep = ",", fill=T, stringsAsFactors=FALSE)

sex.mismatch <- read.table(file=file.path(camp.dir, "LEVEL2/sex_mismatch.txt"),
                           sep="\t", header=F,stringsAsFactors=FALSE)

# camp chanmine issues
```

```
#https://chanmine.bwh.harvard.edu/issues/21110

# fam file format
#A text file with no header line, and one line per sample with the following six fields:

#    Family ID ('FID')
#    Within-family ID ('IID'; cannot be '0')
#    Within-family ID of father ('0' if father isn't in dataset)
#    Within-family ID of mother ('0' if mother isn't in dataset)
#    Sex code ('1' = male, '2' = female, '0' = unknown)
#    Phenotype value ('1' = control, '2' = case, '-9'/'0'/non-numeric = missing data if case/control,

camp.fam <- str_c(qc.dir, "/metadata/CAMP.fam")
camp.fam <- fread(camp.fam)
colnames(camp.fam) <- c("FID","IID","FatherID","MotherID","Sex","Phenotype")
camp.fam$sex[camp.fam$Sex==1]<-"M"; camp.fam$sex[camp.fam$Sex==2]<-"F"
dim(camp.fam) # 3062, 1416 F, 1619 M and 27 NAs
```

```
## [1] 3062    7
```

```
# Save result files with timeStamp
timeStamp <- as.character(round(unclass(Sys.time())))
print(timeStamp)
```

```
## [1] "1620502108"
```

```
# Resource: https://github.com/markgene/maxprobes
cross_probes_file = paste(camp.dir, "/LEVEL2/cross_reactive_probes.txt",
                          sep = "")
if (!file.size(cross_probes_file) == 0){
  cross_probes = read.table(cross_probes_file, sep = "\t",
                            header = F, quote = "\"", fill = T)
  colnames(cross_probes) = c("sample")
  n_cross_probes = nrow(cross_probes)
  n_cross_probes
} else {
  n_cross_probes = 0
}
```

```
## [1] 44570
```

```
n_cross_probes # 44,570
```

```
## [1] 44570
```

```
# load pcs based on autosomes from meffil qc pipeline
load(file=file.path(meff.dir, "qc/pcs5.norm.beta850K.hg19.CAMP_1618342534.Robj"))
fail.samps <- read.table(file=file.path(meff.dir,
                "qc/camp_failed_samples_metrics_hg19_1618342534.txt"),
                sep="\t", header=T,stringsAsFactors=FALSE)
```

2 Data preprocessing and filtering

2.1 Failed Samples filtering and sex mismatches

```
################################
# camp final analysis from RGSet
################################

sex.mismatch <- sex.mismatch$V1; length(sex.mismatch)
```

```
## [1] 29
```

```
sex.mismatch # 29
```

```
##  [1] "TOE542772-BIS-v01_R08C01" "TOE330027-BIS-v01_R02C01"
##  [3] "TOE702035-BIS-v01_R01C01" "TOE549208-BIS-v01_R01C01"
##  [5] "TOE505206-BIS-v01_R06C01" "TOE238326-BIS-v01_R05C01"
##  [7] "TOE459492-BIS-v01_R07C01" "TOE914494-BIS-v01_R01C01"
##  [9] "TOE467706-BIS-v01_R03C01" "TOE934432-BIS-v01_R01C01"
## [11] "TOE729846-BIS-v01_R08C01" "TOE912229-BIS-v01_R06C01"
## [13] "TOE888835-BIS-v01_R03C01" "TOE472940-BIS-v02_R01C01"
## [15] "TOE206521-BIS-v01_R06C01" "TOE333245-BIS-v01_R05C01"
## [17] "TOE921844-BIS-v01_R07C01" "TOE632995-BIS-v01_R07C01"
## [19] "TOE498880-BIS-v01_R05C01" "TOE331554-BIS-v01_R05C01"
## [21] "TOE507952-BIS-v01_R07C01" "TOE593863-BIS-v01_R04C01"
## [23] "TOE724478-BIS-v01_R08C01" "TOE508993-BIS-v01_R01C01"
## [25] "TOE693771-BIS-v01_R05C01" "TOE803419-BIS-v01_R05C01"
## [27] "TOE726946-BIS-v01_R04C01" "TOE421113-BIS-v01_R07C01"
## [29] "TOE133712-BIS-v01_R05C01"
```

```
# removed sex mismatches
RGSet.camp=RGSet.camp[,!colnames(RGSet.camp) %in% sex.mismatch]

#first six samples are sex mismatches so removed
# samples with mixed genotype distributions on the measured SNP probes (59 SNP probes), indicating p

# genotype concordance sample issues
rem <- c("TOE413480-BIS-v01_R08C01",
         "TOE926173-BIS-v01_R02C01",
         "TOE912812-BIS-v01_R01C01",
         "TOE481991-BIS-v01_R05C01",
         "TOE512382-BIS-v01_R05C01",
         "TOE828804-BIS-v01_R05C01",
         "TOE207810-BIS-v01_R08C01",
         "TOE780216-BIS-v01_R07C01",
         "TOE934432-BIS-v01_R01C01",
         "TOE774703-BIS-v01_R06C01",
         "TOE489508-BIS-v01_R05C01",
         "TOE888835-BIS-v01_R03C01",
         "TOE125555-BIS-v01_R04C01",
         "TOE490542-BIS-v01_R01C01",
         "TOE854851-BIS-v01_R03C01",
         "TOE982044-BIS-v01_R01C01",
         "TOE803419-BIS-v01_R05C01",
         "TOE394568-BIS-v01_R03C01")

RGSet.camp=RGSet.camp[,!colnames(RGSet.camp) %in% rem]
intersect(sex.mismatch, rem)
```

```
## [1] "TOE934432-BIS-v01_R01C01" "TOE888835-BIS-v01_R03C01"
## [3] "TOE803419-BIS-v01_R05C01"
```

```
##################################################
# Remove failed samples identified using meffil
##################################################

# loaded this file in file loading section
# selected samples to exclude based on QC report
index <- fail.samps$issue %in% c("Control probe (dye.bias)",
                                 "Methylated vs Unmethylated",
                                 "Control probe (bisulfite1)",
                                 "Control probe (bisulfite2)",
                                 "Control probe (hybe.21771417)",
                                 "Control probe (hybe.28684356)",
                                 "Control probe (hybe.39782321)")

outlier <- fail.samps[index,]
dim(outlier) # 71
```

```
## [1] 71  2
```

```
failed.ids <- unique(outlier$sample.name) # 41
length(failed.ids); failed.ids # finally samples that will be removed
```

```
## [1] 41
##  [1] "TOE122497-BIS-v01_R01C01" "TOE146219-BIS-v01_R08C01"
##  [3] "TOE178434-BIS-v02_R04C01" "TOE198427-BIS-v01_R01C01"
##  [5] "TOE246058-BIS-v01_R01C01" "TOE259805-BIS-v01_R02C01"
##  [7] "TOE285277-BIS-v01_R07C01" "TOE324111-BIS-v01_R08C01"
##  [9] "TOE354006-BIS-v01_R08C01" "TOE368936-BIS-v01_R01C01"
## [11] "TOE383549-BIS-v01_R08C01" "TOE414618-BIS-v01_R08C01"
## [13] "TOE415121-BIS-v01_R03C01" "TOE415449-BIS-v01_R04C01"
## [15] "TOE433638-BIS-v02_R07C01" "TOE453175-BIS-v02_R05C01"
## [17] "TOE467766-BIS-v01_R08C01" "TOE495615-BIS-v02_R02C01"
## [19] "TOE497653-BIS-v01_R08C01" "TOE515047-BIS-v01_R02C01"
## [21] "TOE520586-BIS-v01_R04C01" "TOE538012-BIS-v01_R01C01"
## [23] "TOE558157-BIS-v01_R02C01" "TOE575830-BIS-v01_R02C01"
## [25] "TOE613904-BIS-v01_R02C01" "TOE622588-BIS-v01_R03C01"
## [27] "TOE635128-BIS-v01_R07C01" "TOE642486-BIS-v01_R02C01"
## [29] "TOE646664-BIS-v01_R08C01" "TOE701110-BIS-v01_R08C01"
## [31] "TOE721376-BIS-v01_R08C01" "TOE728789-BIS-v01_R03C01"
## [33] "TOE730561-BIS-v01_R01C01" "TOE781222-BIS-v01_R08C01"
## [35] "TOE801427-BIS-v01_R02C01" "TOE807970-BIS-v01_R01C01"
## [37] "TOE869805-BIS-v01_R08C01" "TOE880818-BIS-v01_R05C01"
## [39] "TOE924783-BIS-v01_R05C01" "TOE951516-BIS-v01_R05C01"
## [41] "TOE958791-BIS-v01_R02C01"
```

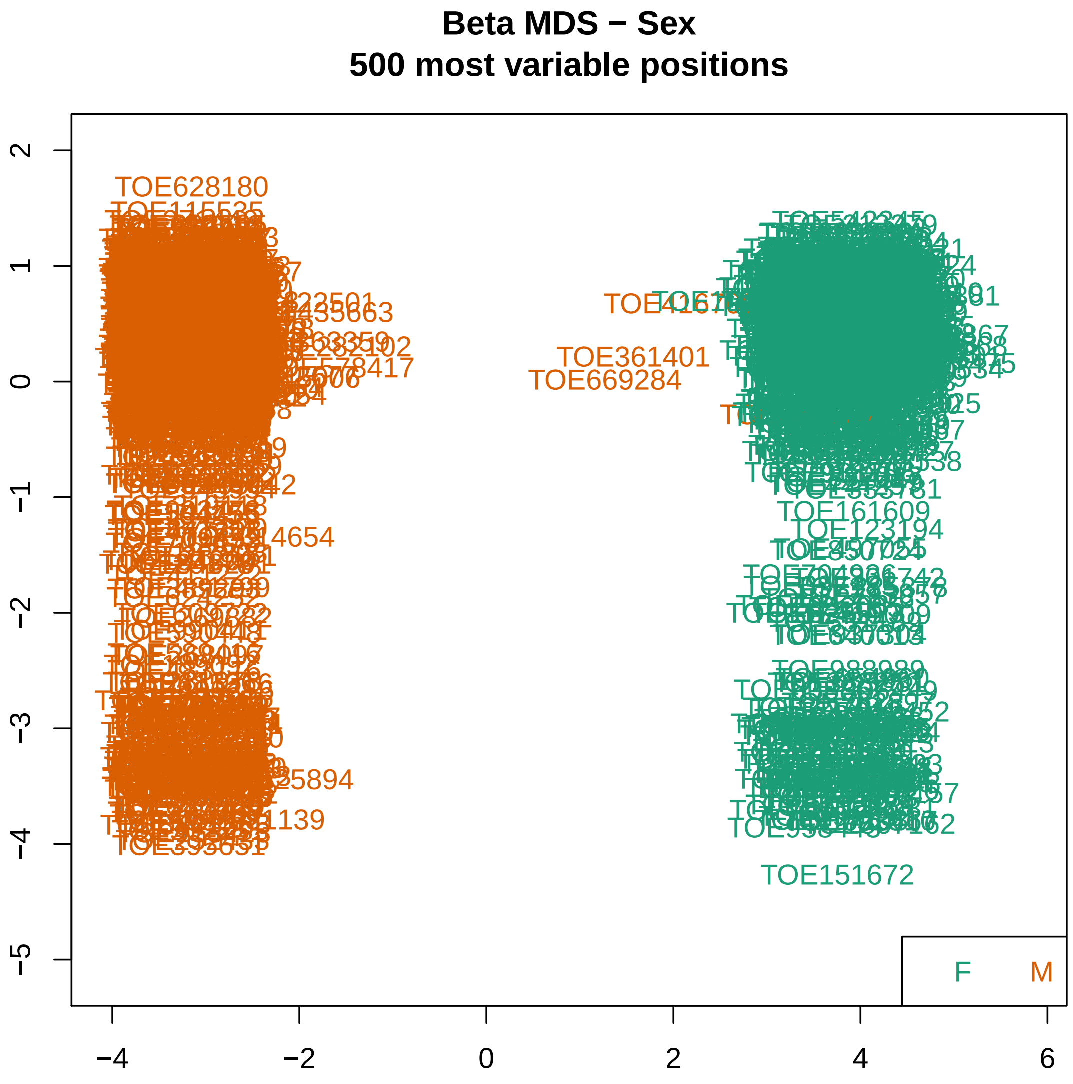
```
# checking overlap of sex mismatches + genotype issues with failed meffil samples
intersect(sex.mismatch, failed.ids); intersect(failed.ids, rem)
```

```
## character(0)
## character(0)
```

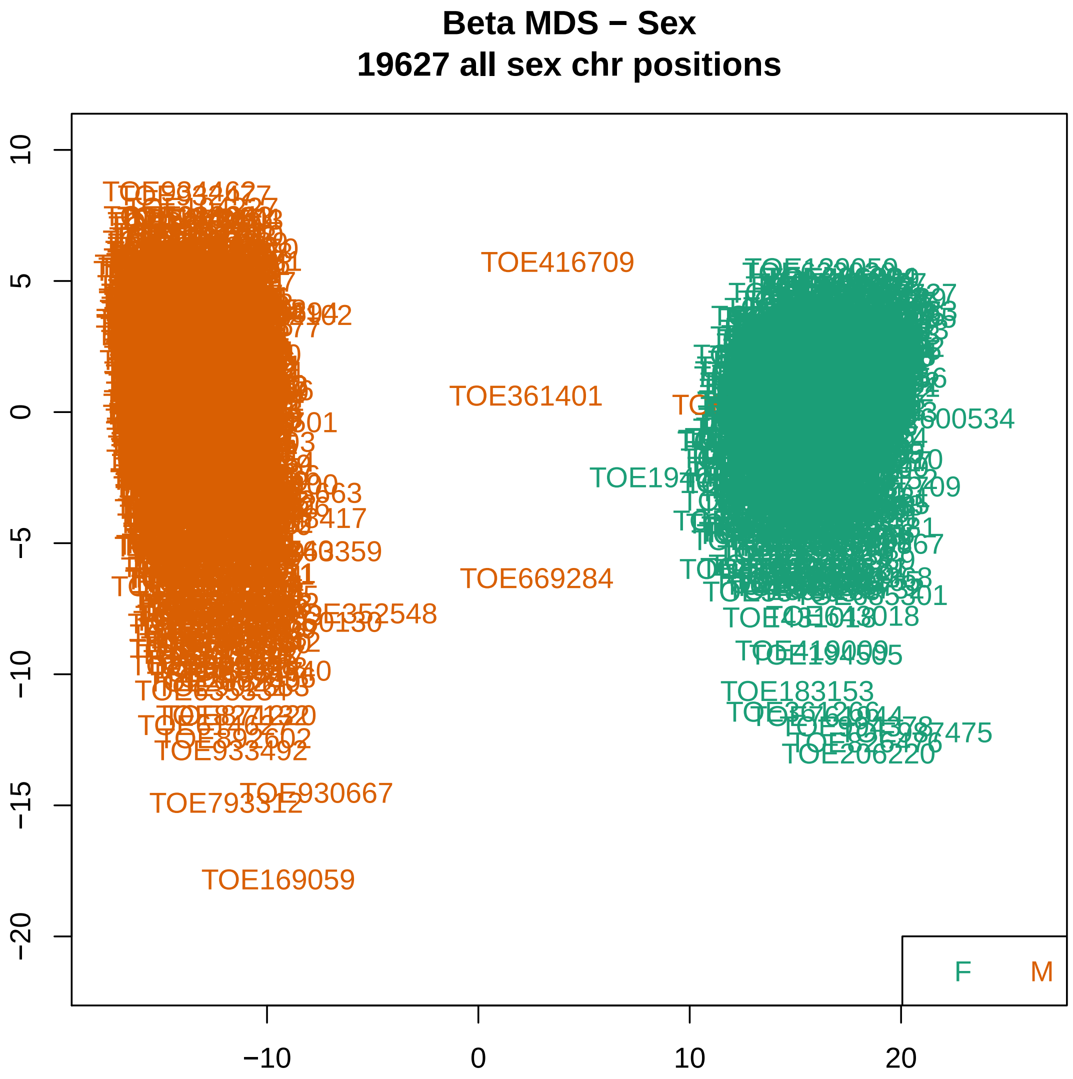
```
RGSet.camp=RGSet.camp[,!colnames(RGSet.camp) %in% failed.ids]
dim(RGSet.camp) # 1531
```

```
## [1] 1008711    1531
```

```
betas.chk <- getBeta(RGSet.camp)
pData.camp <- pData(RGSet.camp)
ann850k <- getAnnotation(RGSet.camp)
xychr = rownames(betas.chk) %in% ann850k$Name[ann850k$chr %in% c("chrX","chrY")]
betas.xy = betas.chk[xychr,]

# checking if you see any outliers
mdsPlot(as.matrix(betas.chk), numPositions=500, main=sprintf("Beta MDS - Sex\n%d most variable positi
```

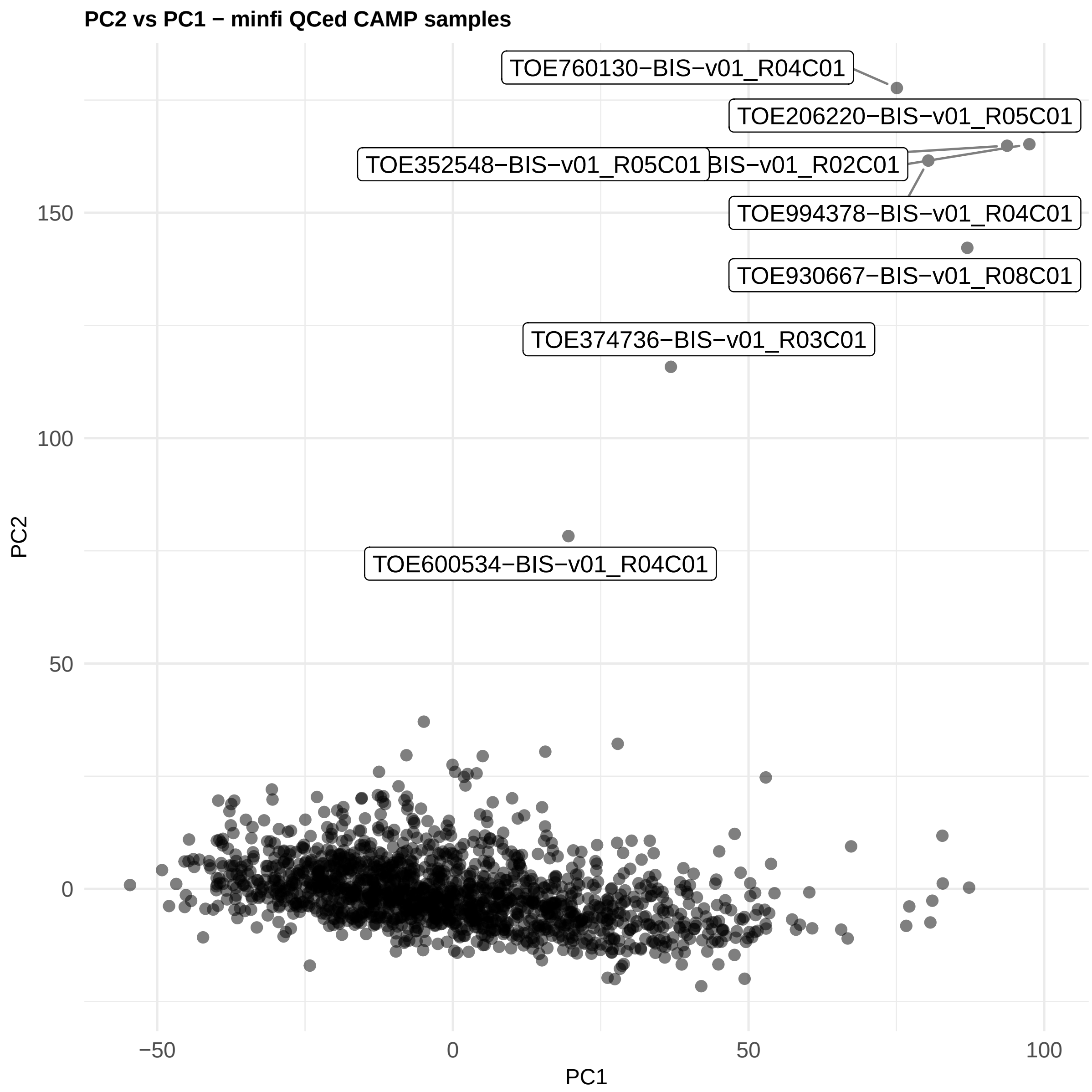


```
mdsPlot(as.matrix(betas.xy), numPositions=19627, main=sprintf("Beta MDS - Sex\n%d all sex chr positi
```

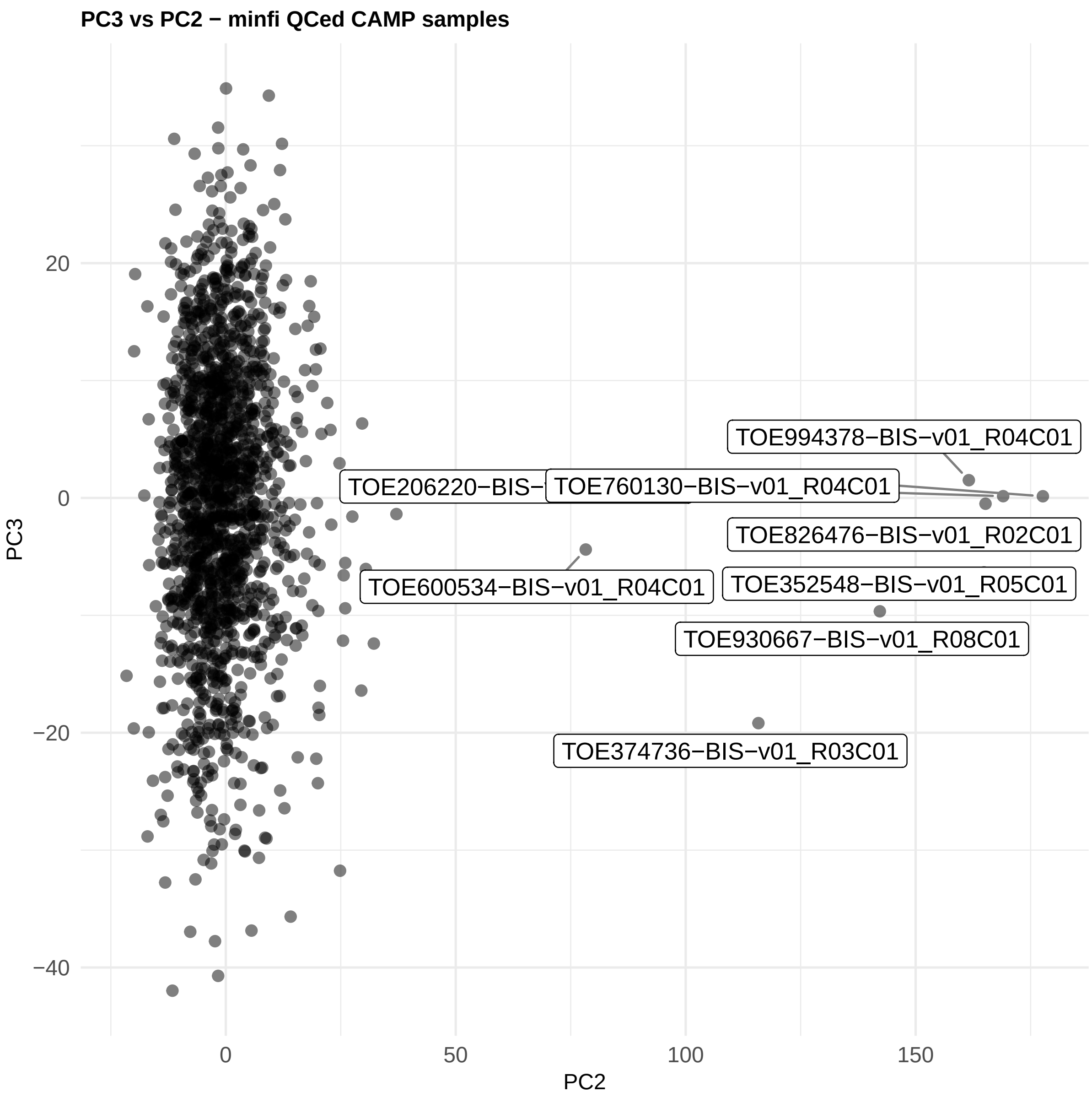


```
pdf(file = file.path(plots.dir, "MDS_sex_out_500pos_names.pdf"), width = 6, height = 6)
mdsPlot(as.matrix(betas.chk), numPositions=500, main=sprintf("Beta MDS - Sex\n%d most variable positi
dev.off()

## pdf
##   2

pdf(file = file.path(plots.dir, "MDS_sex_chr_out_19627pos_names.pdf"), width = 6, height = 6)
mdsPlot(as.matrix(betas.xy), numPositions=19627, main=sprintf("Beta MDS - Sex\n%d all sex chr positio
dev.off()

## pdf
##   2

# sex outliers identified as above
sex.out <- c("TOE416709-BIS-v01_R03C01","TOE361401-BIS-v01_R01C01","TOE669284-BIS-v02_R05C01", "TOE35

RGSet.camp=RGSet.camp[,!colnames(RGSet.camp) %in% sex.out]
dim(RGSet.camp)

## [1] 1008711    1527
```

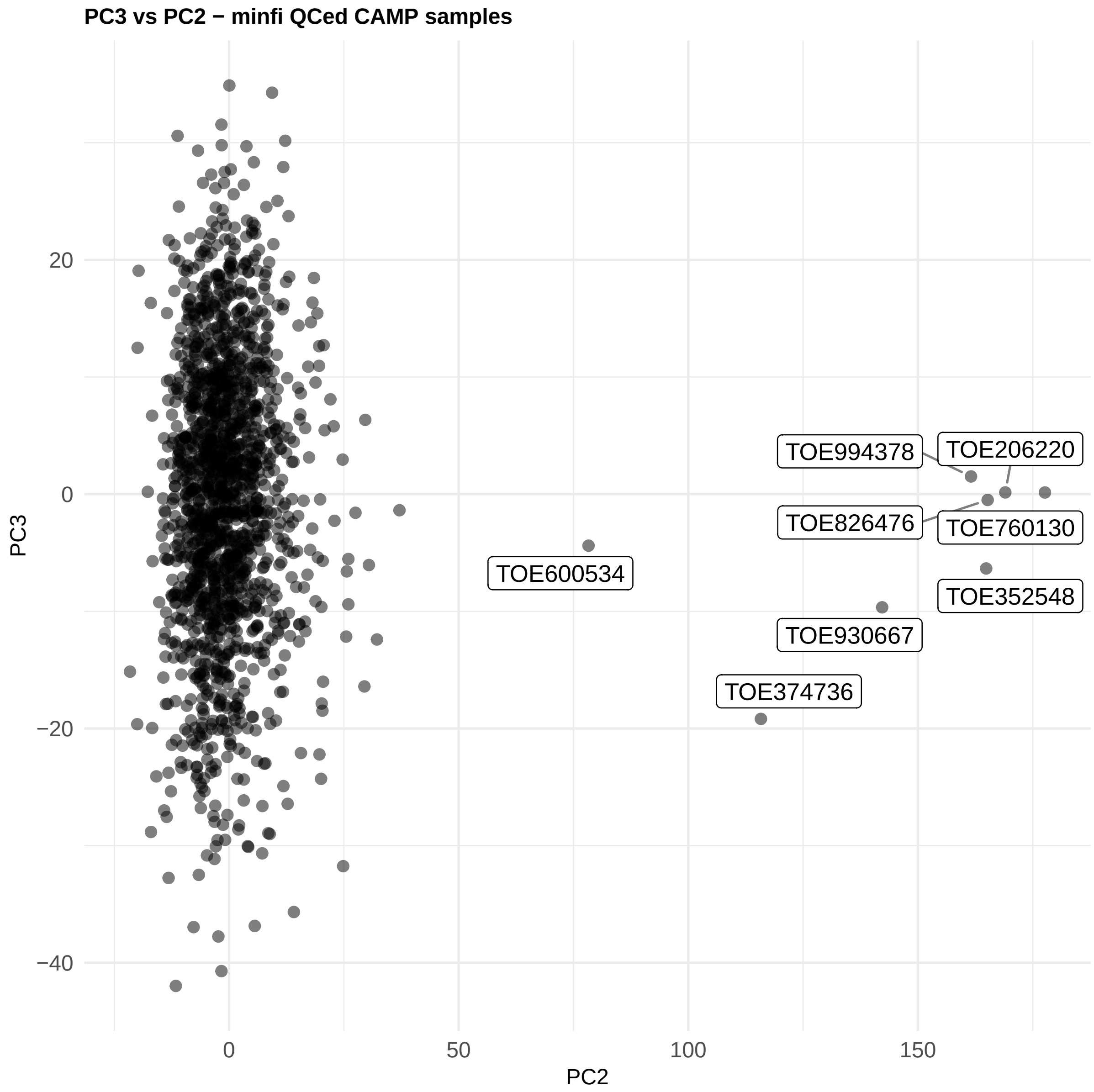
```
intersect(rem, sex.out);intersect(failed.ids, sex.out)

## character(0)
## character(0)

# outlier samples as identified in meffil norm report
pcs <- data.frame(pcs)
TOE <- data.frame(do.call('rbind', strsplit(as.character(rownames(pcs)),'_',fixed=TRUE)))
TOE <- data.frame(do.call('rbind', strsplit(as.character(TOE$X1),'-',fixed=TRUE)))
pcs$TOEID <- TOE$X1

ggplot(pcs, aes(x = PC1, y = PC2)) +
    geom_point(alpha = 0.5, size = 2) +
    labs(title = "PC2 vs PC1 - minfi QCed CAMP samples") +
    geom_label_repel(aes(label = ifelse(PC2 > 50, rownames(pcs), "")),
                     box.padding   = 0.25,
                     point.padding = 0.5,
                     segment.color = 'grey50') +
    theme_minimal() +
    theme(plot.title = element_text(size = 10, face = "bold"),
          axis.text = element_text(size = 10),
          axis.title = element_text(size = 10))
```



```
ggsave(path=plots.dir, "pc1_pc2_samp_outliers_camp.png", width = 8, height = 6)

ggplot(pcs, aes(x = PC2, y = PC3)) +
    geom_point(alpha = 0.5, size = 2) +
    labs(title = "PC3 vs PC2 - minfi QCed CAMP samples") +
    geom_label_repel(aes(label = ifelse(PC2 > 50, rownames(pcs), "")),
                     box.padding   = 0.25,
                     point.padding = 0.5,
                     segment.color = 'grey50') +
    theme_minimal() +
    theme(plot.title = element_text(size = 10, face = "bold"),
          axis.text = element_text(size = 10),
          axis.title = element_text(size = 10))
```



```
ggplot(pcs, aes(x = PC2, y = PC3)) +
    geom_point(alpha = 0.5, size = 2) +
    labs(title = "PC3 vs PC2 - minfi QCed CAMP samples") +
    geom_label_repel(aes(label = ifelse(PC2 > 50, TOEID, "")),
                     box.padding   = 0.25,
                     point.padding = 0.5,
                     segment.color = 'grey50') +
    theme_minimal() +
    theme(plot.title = element_text(size = 10, face = "bold"),
          axis.text = element_text(size = 10),
          axis.title = element_text(size = 10))
```



```
ggsave(path=plots.dir, "pc3_pc2_samp_outliers_camp.png", width = 8, height = 6)

# removing those outliers
samp.out <- c("TOE826476-BIS-v01_R02C01","TOE994378-BIS-v01_R04C01","TOE930667-BIS-v01_R08C01", "TOE2

RGSet.camp=RGSet.camp[,!colnames(RGSet.camp) %in% samp.out]
dim(RGSet.camp)

## [1] 1008711    1519

intersect(rem, sex.out);intersect(failed.ids, sex.out)

## character(0)
## character(0)

# probably not needed anymore as these plots are also generated using meffil
#library(ENmix)
#jpeg(file = file.path(plots.dir, "ENmixcontrol_plots_CAMP_bisulfite.jpg"),
#     width = 750, height = 1500)
#plotCtrl(RGSet.camp)
#dev.copy(jpeg,'ENmixcontrol_plots_CAMP_bisulfite.jpg')
#dev.off()
```

2.2 detP calculation

```
##############
# Detection P
##############

detP.camp <- detectionP(RGSet.camp, type="m+u")
print(table(detP.camp>0.05))
```

```
##
##      FALSE       TRUE
## 1309695901    5543920
```

```
print(table(detP.camp>0.01))
```

```
##
##      FALSE       TRUE
## 1306705356    8534465
```

```
#####################
# sample-wise thresh.
#####################
# colMedians(detP.camp) similar# robustbase r package
dim(detP.camp[,colMeans(detP.camp)>0.001])
```

```
## [1] 865859    576
```

```
dim(detP.camp[,colMeans(detP.camp)>0.002])
```

```
## [1] 865859    282
```

```
dim(detP.camp[,colMeans(detP.camp)>0.003])
```

```
## [1] 865859    167
```

```
dim(detP.camp[,colMeans(detP.camp)>0.004])
```

```
## [1] 865859    121
```

```
dim(detP.camp[,colMeans(detP.camp)>0.005])
```

```
## [1] 865859     80
```

```
dim(detP.camp[,colMeans(detP.camp)>0.01])
```

```
## [1] 865859     12
```

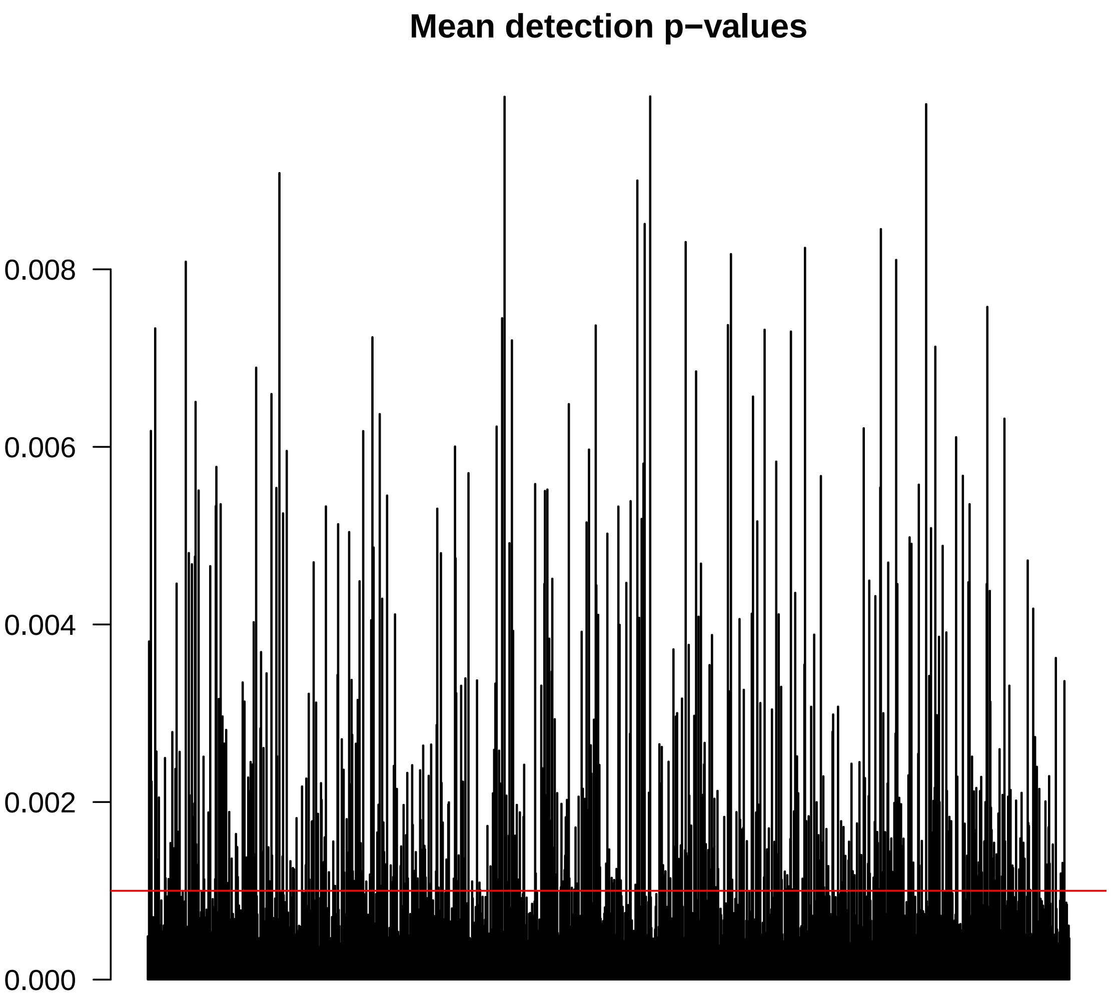
```
# was 35 before, histograms for these samples looked poor,
#but some of those were sex mismatches that may have been removed earlier now
# from RGSet so the no may be much less now
dim(detP.camp[,colMeans(detP.camp)>0.05])
```

```
## [1] 865859      0
```

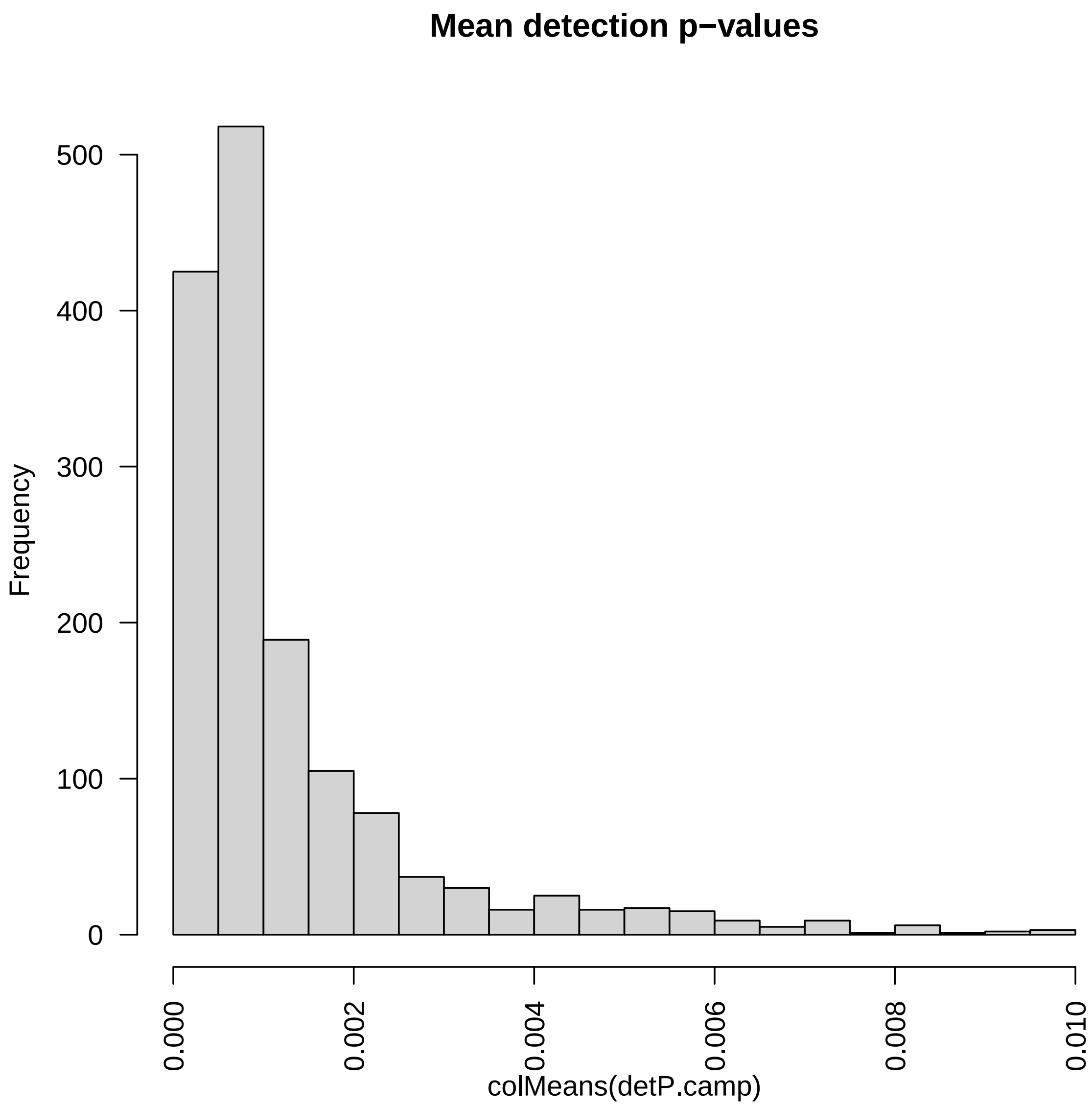
```
##################################################
# Remove failed samples identified using detP
##################################################
detP.samp.fail <- colnames(detP.camp[,colMeans(detP.camp)>0.01])
detP.samp.fail
```

```
##  [1] "TOE516265-BIS-v02_R08C01" "TOE643018-BIS-v02_R05C01"
##  [3] "TOE685301-BIS-v01_R04C01" "TOE933492-BIS-v01_R06C01"
##  [5] "TOE827132-BIS-v01_R05C01" "TOE169059-BIS-v01_R08C01"
##  [7] "TOE361266-BIS-v01_R08C01" "TOE183153-BIS-v01_R07C01"
##  [9] "TOE419009-BIS-v01_R08C01" "TOE987475-BIS-v01_R07C01"
## [11] "TOE200240-BIS-v01_R01C01" "TOE902653-BIS-v01_R04C01"
```

```
# some of these may be removed during filtering above, or present in meth-unmeth outliers
# sex outliers and control probe issues in meffil after filtering qc reports

RGSet.camp=RGSet.camp[,!colnames(RGSet.camp) %in% detP.samp.fail]
dim(RGSet.camp)

## [1] 1008711    1507

detP.camp <- detectionP(RGSet.camp, type="m+u")

save(detP.camp, file=file.path(results.dir,paste0("detP.camp_hg19_",
                                                   timeStamp,".RData")))

########################
# Failed detP probes minfi
########################
# Threshold of detP 0.01 in more than 25% of the samples using minfi stats
failed.01<-detP.camp > 0.01
#colMeans(failed.01) # Fraction of failed positions per sample
sum(colMeans(failed.01)>0.20) # >20% probes failed per sample

## [1] 0

sum(rowMeans(failed.01)>0.05)

## [1] 23919

sum(rowMeans(failed.01)>0.10)

## [1] 14294

sum(rowMeans(failed.01)>0.15)

## [1] 9815

sum(rowMeans(failed.01)>0.20) # should be same as length(failedProbes)

## [1] 7208

sum(rowMeans(failed.01)>0.25)

## [1] 5440

# How many positions failed in >20% of samples?
failedProbes <- rownames(failed.01)[rowMeans(failed.01)>0.20]
length(failedProbes)

## [1] 7208

save(failedProbes, file=file.path(results.dir,paste0("failedProbes_CAMP_hg19_", timeStamp,".RData")))

# plots
barplot(colMeans(detP.camp), las=2, axisnames = FALSE, main="Mean detection p-values")
abline(h=0.001,col="red")
```



```
hist(colMeans(detP.camp), las=2, main="Mean detection p-values", breaks=30)
```



```
pdf(file = file.path(plots.dir, "colMeans_detP_check_failed_samples_camp.pdf"),
    width = 10, height = 5)
barplot(colMeans(detP.camp), las=2, axisnames = FALSE, main="Mean detection p-values")
abline(h=0.001,col="red")
dev.off()

## pdf
##   2

pdf(file = file.path(plots.dir, "colMeans_detP_all_samples_hist_camp.pdf"),
    width = 6, height = 5)
hist(colMeans(detP.camp), las=2, main="Mean detection p-values", breaks=30)
dev.off()

## pdf
##   2

# plotting every sample, Not printing this because the loop will run for all and will likely print a
pdf(file = file.path(plots.dir, "detP_all_samples_hist_freq_camp.pdf"),
    width = 160, height = 164)
par(mfrow=c(40, 38))
```

```
colnames <- dimnames(detP.camp)[[2]]
for (i in 1:1519) {
  #print(i)
  hist(log10(detP.camp[,i]), las=2, breaks=50, main=colnames[i], col="gray", border="white")
}
```

```
## Error in h(simpleError(msg, call)):  error in evaluating the argument 'x' in selecting
a method for function 'hist':  subscript out of bounds
```

```
dev.off()
```

```
## pdf
##   2
```

2.3 svas and cell type count estimation

```
# generate surrogate variables derived based on intensity data
# for non-negative internal control probes
# this step is pretty quick
csva<-ctrlsva(RGSet.camp)
```

```
## 16  surrogate variables explain  95.25751 % of
##     data variation
```

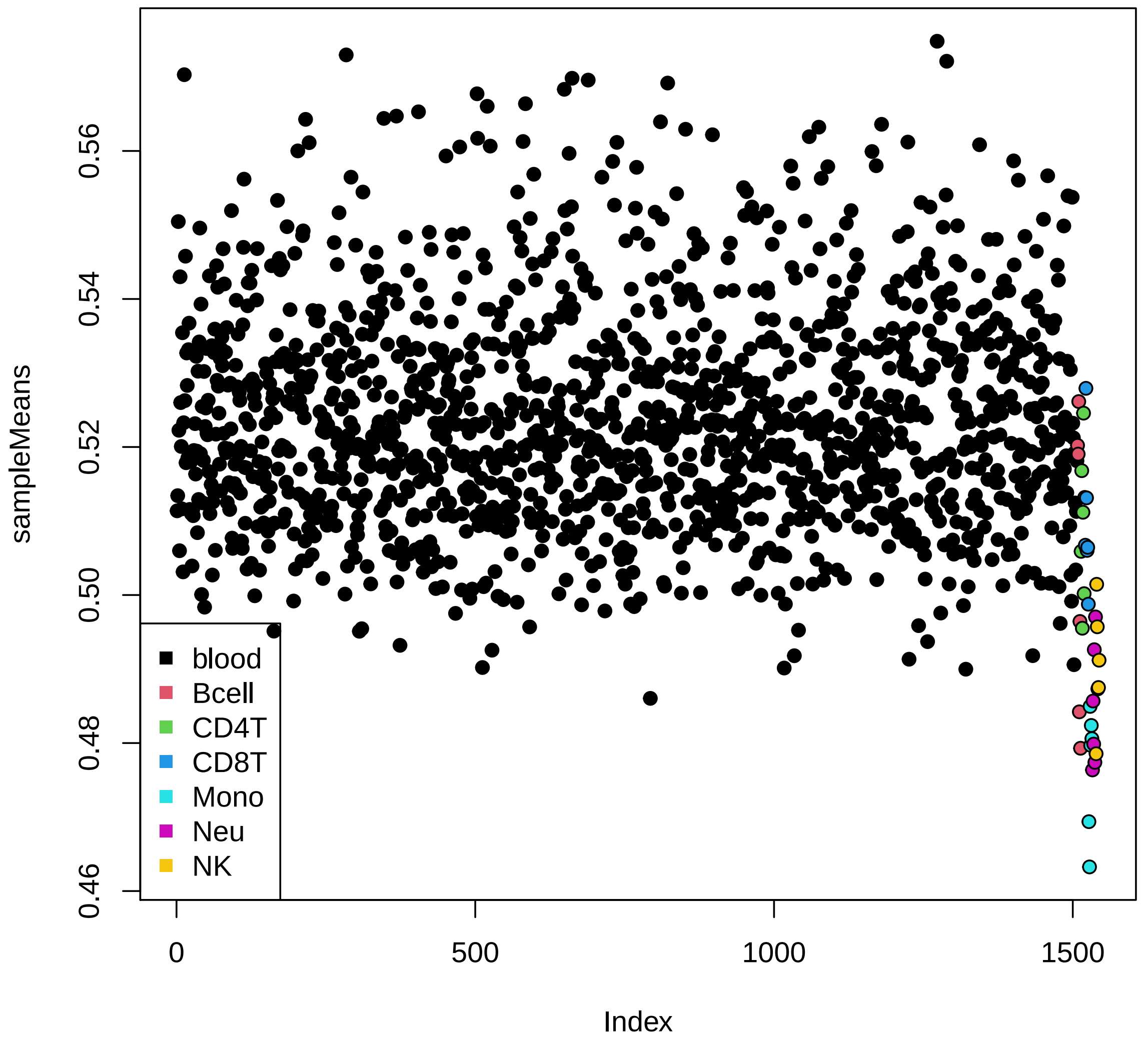
```
save(csva, file=file.path(results.dir,paste0("camp_svas_rawdata_hg19_",
                                             timeStamp, ".RData")))

# cell count estimates
library("FlowSorted.Blood.EPIC")
library(ExperimentHub)
hub <- ExperimentHub()
#> snapshotDate(): 2020-10-02
epicref <- query(hub, "FlowSorted.Blood.EPIC")
epicref; epicref$title
```

```
## ExperimentHub with 1 record
## # snapshotDate(): 2020-10-27
## # names(): EH1136
## # package(): FlowSorted.Blood.EPIC
## # $dataprovider: GEO
## # $species: Homo sapiens
## # $rdataclass: RGChannelSet
## # $rdatadateadded: 2018-04-20
## # $title: FlowSorted.Blood.EPIC: Illumina Human Methylation data from EPIC o...
## # $description: The FlowSorted.Blood.EPIC package contains Illumina HumanMet...
## # $taxonomyid: 9606
## # $genome: hg19
## # $sourcetype: tar.gz
## # $sourceurl: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE110554
## # $sourcesize: NA
## # $tags: c("ExperimentData", "Homo_sapiens_Data", "Tissue",
## #   "MicroarrayData", "Genome", "TissueMicroarrayData",
## #   "MethylationArrayData")
## # retrieve record with 'object[["EH1136"]]'
## [1] "FlowSorted.Blood.EPIC: Illumina Human Methylation data from EPIC on immunomagnetic sorted adu
```

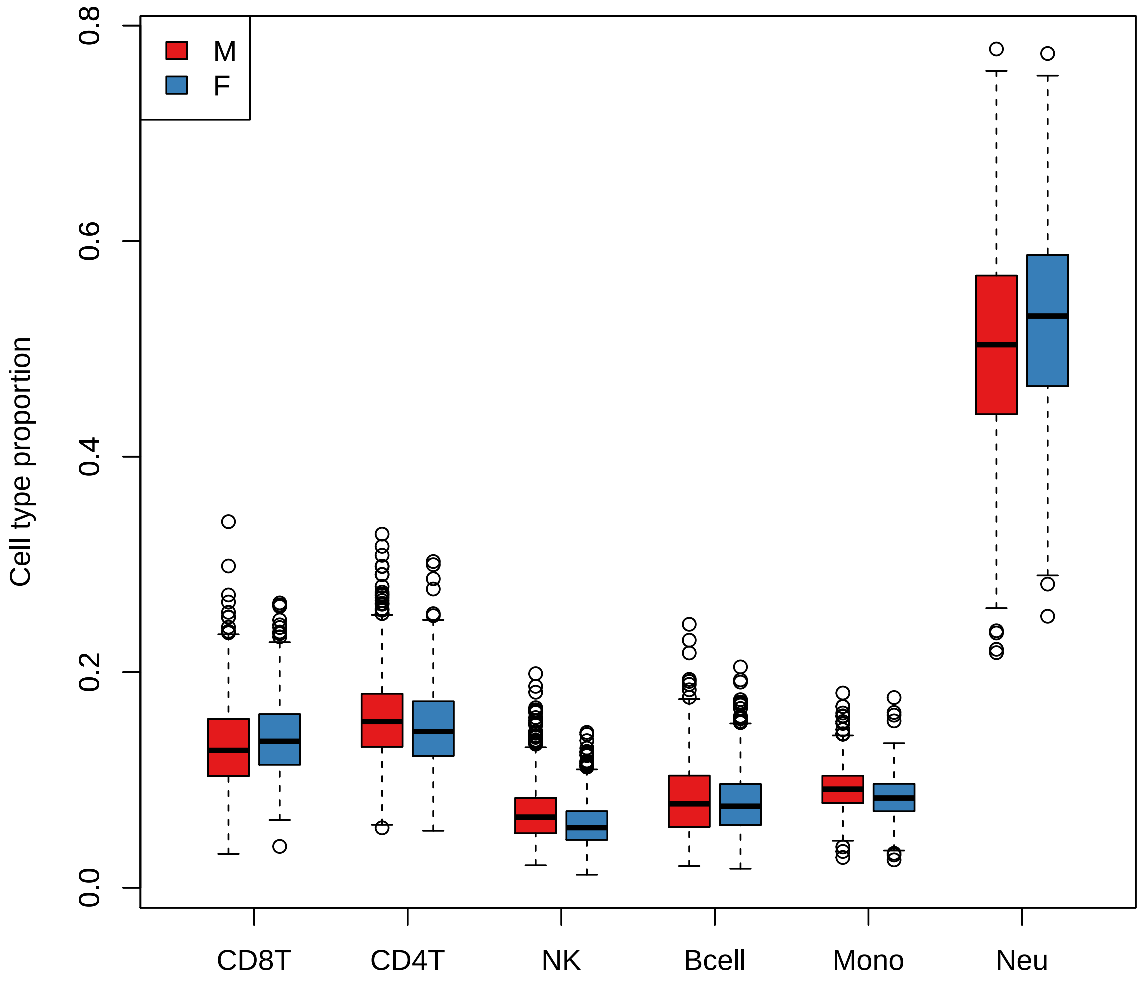
```
FlowSorted.Blood.EPIC.ref <- epicref[[1]]
FlowSorted.Blood.EPIC.ref
```

```
## class: RGChannelSet
## dim: 1051815 49
## metadata(0):
## assays(2): Green Red
## rownames(1051815): 1600101 1600111 ... 99810990 99810992
## rowData names(0):
## colnames(49): 201868500150_R01C01 201868500150_R03C01 ...
##   201870610111_R06C01 201870610111_R07C01
## colData names(32): Sample_Plate Sample_Well ... filenames normalmix
## Annotation
##   array: IlluminaHumanMethylationEPIC
##   annotation: ilm10b4.hg19
```

```
if (memory.limit()>8000){
  countsEPIC<-estimateCellCounts2(RGSet.camp, compositeCellType = "Blood",
                                processMethod = "preprocessFunnorm",
                                cellTypes = c("CD8T", "CD4T", "NK", "Bcell",
                                              "Mono", "Neu"),
                                referencePlatform =
                                  "IlluminaHumanMethylationEPIC",
                                referenceset = "FlowSorted.Blood.EPIC.ref",
                                IDOLOptimizedCpGs =NULL,
                                returnAll = TRUE,
                                meanPlot = TRUE,
                                verbose = TRUE)
}
```



```
save(countsEPIC, file=file.path(results.dir,paste0("camp_EPIC_estimatecellcounts_hg19_",
                                                   timeStamp,".RData")))
celltype.est.2 <- countsEPIC$counts
save(celltype.est.2, file=file.path(results.dir,
                    paste0("camp_EPIC_estimatecellcounts2_result_hg19_",
                           timeStamp, ".RData")))

TOE <- data.frame(do.call('rbind', strsplit(as.character(samplesheet.camp$Basename),'/',fixed=TRUE)))
samplesheet.camp$TOE_RC <- TOE$X14
ct.sampsheet <- merge(celltype.est.2, samplesheet.camp, by.x="row.names", by.y="TOE_RC", sort=F)
#camp.pheno.sel=camp.pheno[,c("camp","S_SUBJECTID"), drop=FALSE]
#ct.pheno <- merge(ct.sampsheet, camp.pheno.sel, by="S_SUBJECTID", sort=F)
par(mfrow=c(1,1));sex.pal <- brewer.pal(8,"Set1")
a = celltype.est.2[ct.sampsheet$Gender == "M",]
b = celltype.est.2[ct.sampsheet$Gender == "F",]
boxplot(a, at=0:5*3 + 1, xlim=c(0, 18), ylim=range(a, b), xaxt="n",
        col=sex.pal[1], main="", ylab="Cell type proportion")
boxplot(b, at=0:5*3 + 2, xaxt="n", add=TRUE, col=sex.pal[2])
axis(1, at=0:5*3 + 1.5, labels=colnames(a), tick=TRUE)
legend("topleft", legend=c("M","F"), fill=sex.pal)
```



2.4 Noob normalization and funnorm

```
# clearing up some objects from memory no longer needed
rm(countsEPIC)
rm(celltype.est.2)
rm(detP.camp)
rm(csva)
######################
# NOOB Normalization
######################

MSet.noob.camp <- preprocessNoob(RGSet.camp, offset = 15, dyeCorr = TRUE, verbose = TRUE)

################################################################################
# This object is sample cleaned but probe filtering has not been performed for this,
# check probe filtering steps in betas cleaning code
# if you use this object for any further downstream analysis
################################################################################
```

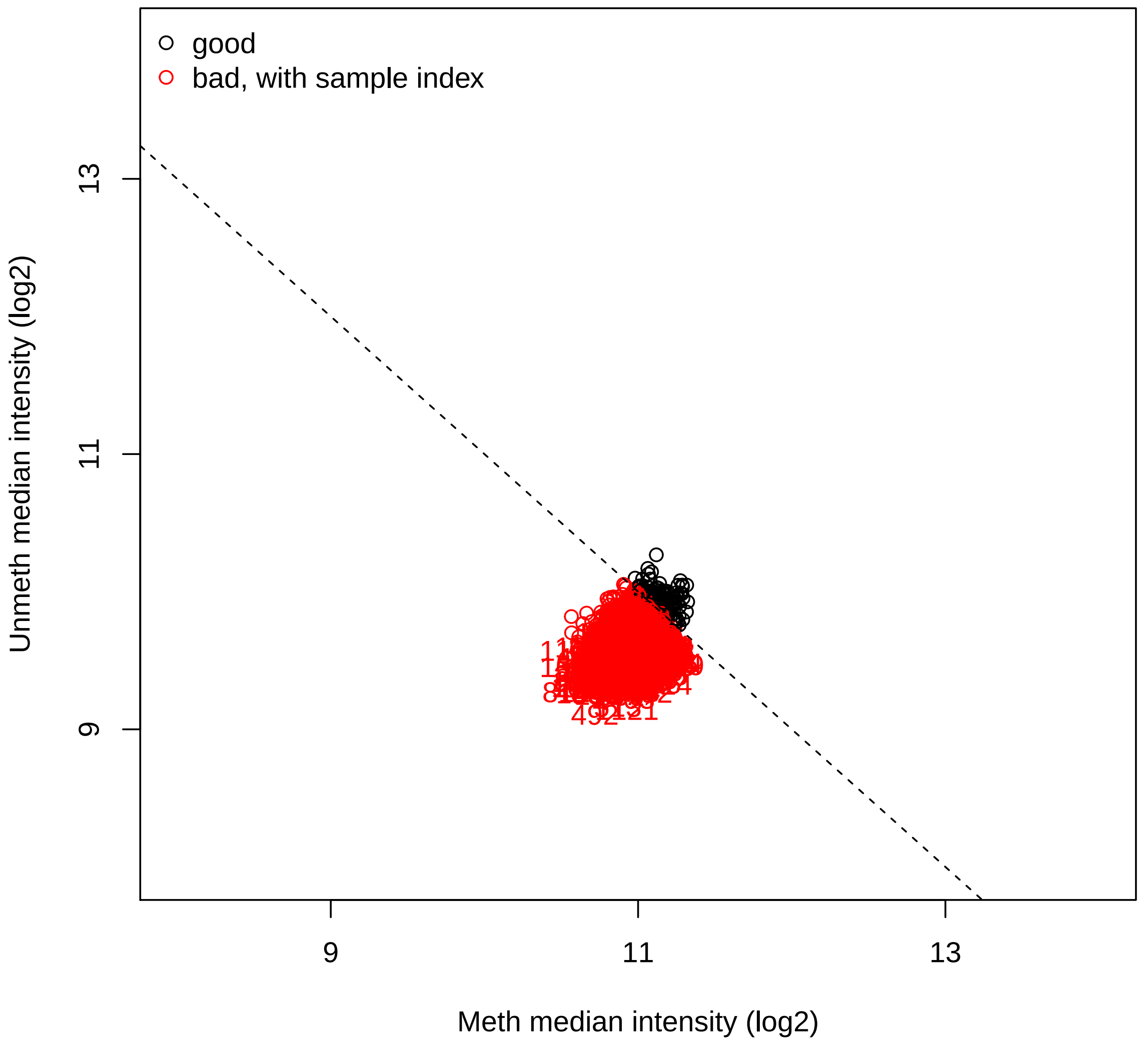
```
save(MSet.noob.camp, file=file.path(results.dir,paste0("Mset.noob.camp_hg19_",
                                                       timeStamp,".RData")))
MSet.noob.camp <- NULL

################################################################################
# shows that phenotype file and LIMS/samplesheet genders match for the probands
################################################################################
pData.camp <- pData(RGSet.camp)
pData.pheno <- merge(pData.camp, camp.pheno, by="S_SUBJECTID", sort=F)
table(pData.pheno$Gender, pData.pheno$SEX)
```

```
##
##         1   2
##   F   0 288
##   M 437   0
```

```
pData.camp$Sex[pData.camp$Gender=="F"]<-0
pData.camp$Sex[pData.camp$Gender=="M"]<-1
sex <- pData.camp$Sex

# Runs for few hours, I wanted the output to be in mset format, otherwise, I can not extract methyla
mset.camp.funnorm <- preprocessFunnorm(RGSet.camp, nPCs=5, sex=sex, ratioConvert = FALSE,
                                       bgCorr=TRUE, dyeCorr=TRUE, verbose=TRUE)
mset.camp.funnorm <- addSex(mset.camp.funnorm)

############################################
# Dataset after functional normalization
############################################
save(mset.camp.funnorm, file=file.path(results.dir,paste0("mset.camp.funnorm_hg19_",
                                                          timeStamp,".RData")))

betas <- getBeta(mset.camp.funnorm)
ann850k <- getAnnotation(mset.camp.funnorm)
#pData.camp <- pData(mset.camp.funnorm)

# median meth and unmeth intensities plot
qc <- getQC(mset.camp.funnorm)
meds <- (qc$mMed + qc$uMed)/2
mMed <- qc@listData$mMed
uMed <- qc@listData$uMed
qc.camp <- data.frame(mMed, uMed, meds)
rownames(qc.camp) <- qc@rownames
dim(qc.camp[qc.camp$meds<10.5,])
```

```
## [1] 1356    3
```

```
plotQC(qc)
```



```
# Plots
pdf(file = file.path(plots.dir, "QC_samples_minfi_detP0.01_20per_samples_camp.pdf"),
    width = 5, height = 5)
plotQC(qc)
dev.off()

## pdf
##   2
```

3 Session information

[1] "2021-05-09" [1] "2021-05-09 09:26:43 EDT"

- R version 4.0.3 (2020-10-10), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=en_US.UTF-8, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Running under: CentOS Linux 7 (Core)

- Matrix products: default
- BLAS: `/app/R-4.0.3@i86-rhel7.0/lib64/R/lib/libRblas.so`
- LAPACK: `/app/R-4.0.3@i86-rhel7.0/lib64/R/lib/libRlapack.so`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, stats4, utils
- Other packages: annotate 1.68.0, AnnotationDbi 1.52.0, AnnotationHub 2.22.1, Biobase 2.50.0, BiocFileCache 1.14.0, BiocGenerics 0.36.1, BiocParallel 1.24.1, Biostrings 2.58.0, bumphunter 1.32.0, Cairo 1.5-12.2, colorRamps 2.3, data.table 1.14.0, dbplyr 2.1.0, DNAcopy 1.64.0, doParallel 1.0.16, dplyr 1.0.3, e1071 1.7-6, ENmix 1.26.10, ExperimentHub 1.16.1, fastICA 1.2-2, FDb.InfiniumMethylation.hg19 2.2.0, FlowSorted.Blood.EPIC 1.8.0, forcats 0.5.1, foreach 1.5.1, gdsfmt 1.26.1, genefilter 1.72.1, geneplotter 1.68.0, GenomeInfoDb 1.26.7, GenomicFeatures 1.42.3, GenomicRanges 1.42.0, GGally 2.1.0, ggplot2 3.3.3, ggrepel 0.9.1, gplots 3.1.1, gridExtra 2.3, here 1.0.1, IlluminaHumanMethylation450kanno.ilmn12.hg19 0.6.0, IlluminaHumanMethylationEPICanno.ilm10b4.hg19 0.6.0, IlluminaHumanMethylationEPICmanifest 0.3.0, illuminaio 0.32.0, impute 1.64.0, IRanges 2.24.1, isva 1.9, iterators 1.0.13, JADE 2.0-3, knitr 1.33, lattice 0.20-44, limma 3.46.0, lme4 1.1-26, locfit 1.5-9.4, lumi 2.42.0, markdown 1.1, MASS 7.3-54, Matrix 1.3-3, MatrixGenerics 1.2.1, matrixStats 0.58.0, meffil 1.1.1, methylumi 2.36.0, mgcv 1.8-35, minfi 1.36.0, multcomp 1.4-17, mvtnorm 1.1-1, nlme 3.1-152, org.Hs.eg.db 3.12.0, plyr 1.8.6, preprocessCore 1.52.1, purrr 0.3.4, quadprog 1.5-8, qvalue 2.22.0, R.methodsS3 1.8.1, R.oo 1.24.0, R.utils 2.10.1, RColorBrewer 1.1-2, readr 1.4.0, reshape2 1.4.4, robustbase 0.93-7, ROC 1.66.0, RSpectra 0.16-0, S4Vectors 0.28.1, scales 1.1.1, SmartSVA 0.1.3, statmod 1.4.35, stringi 1.5.3, stringr 1.4.0, SummarizedExperiment 1.20.0, survival 3.2-11, sva 3.38.0, TH.data 1.0-10, tibble 3.1.1, tidyr 1.1.3, tidyverse 1.3.0, TxDb.Hsapiens.UCSC.hg19.knownGene 3.2.2, wateRmelon 1.34.0, XML 3.99-0.6, XVector 0.30.0
- Loaded via a namespace (and not attached): affy 1.68.0, affyio 1.60.0, askpass 1.1, assertthat 0.2.1, backports 1.2.1, base64 2.0, beanplot 1.2, BiocManager 1.30.12, BiocVersion 3.12.0, biomaRt 2.46.3, bit 4.0.4, bit64 4.0.5, bitops 1.0-7, blob 1.2.1, boot 1.3-28, broom 0.7.6, cachem 1.0.4, caTools 1.18.2, cellranger 1.1.0, class 7.3-19, cli 2.5.0, clue 0.3-59, cluster 2.1.2, codetools 0.2-18, colorspace 2.0-1, compiler 4.0.3, crayon 1.4.1, curl 4.3.1, DBI 1.1.1, DelayedArray 0.16.3, DelayedMatrixStats 1.12.3, DEoptimR 1.0-8, digest 0.6.27, doRNG 1.8.2, dynamicTreeCut 1.63-1, edgeR 3.32.1, ellipsis 0.3.2, evaluate 0.14, fansi 0.4.2, farver 2.1.0, fastmap 1.1.0, fs 1.5.0, generics 0.1.0, GenomeInfoDbData 1.2.4, GenomicAlignments 1.26.0, GEOquery 2.58.0, glue 1.4.2, grid 4.0.3, gtable 0.3.0, gtools 3.8.2, haven 2.4.1, HDF5Array 1.18.1, highr 0.9, hms 1.0.0, htmltools 0.5.1.1, httpuv 1.6.0, httr 1.4.2, interactiveDisplayBase 1.28.0, irr 0.84.1, jsonlite 1.7.2, KernSmooth 2.23-20, labeling 0.4.2, later 1.2.0, lifecycle 0.2.0, lpSolve 5.6.15, lubridate 1.7.10, magrittr 2.0.1, mclust 5.4.7, memoise 2.0.0, mime 0.10, minqa 1.2.4, modelr 0.1.8, multtest 2.46.0, munsell 0.5.0, nleqslv 3.3.2, nloptr 1.2.2.2, nor1mix 1.3-0, openssl 1.4.4, pillar 1.6.0, pkgconfig 2.0.3, prettyunits 1.1.1, progress 1.2.2, promises 1.2.0.1, proxy 0.4-25, ps 1.6.0, R6 2.5.0, rappdirs 0.3.3, Rcpp 1.0.6, RCurl 1.98-1.3, readxl 1.3.1, reprex 2.0.0, reshape 0.8.8, rhdf5 2.34.0, rhdf5filters 1.2.1, Rhdf5lib 1.12.1, rlang 0.4.9, rngtools 1.5, RPMM 1.25, rprojroot 2.0.2, Rsamtools 2.6.0, RSQLite 2.2.3, rstudioapi 0.13, rtracklayer 1.50.0, rvest 0.3.6, sandwich 3.0-0, scrime 1.3.5, shiny 1.6.0, siggenes 1.64.0, sparseMatrixStats 1.2.1, splines 4.0.3, tidyselect 1.1.1, tools 4.0.3, utf8 1.2.1, vctrs 0.3.6, withr 2.4.2, xfun 0.22, xml2 1.3.2, xtable 1.8-4, yaml 2.2.1, zlibbioc 1.36.0, zoo 1.8-9