

# CAMP IgE EWAS

reprk: PKachroo  
CDNM, BWH

November 30, 2021

## Contents

<b>1</b>	<b>Setup</b>	<b>2</b>
1.1	Packages, Data locations and loading . . . . .	8
<b>2</b>	<b>Data wrangling and prep for SVA</b>	<b>9</b>
<b>3</b>	<b>SVA estimation</b>	<b>16</b>
<b>4</b>	<b>Correlations PCs and SVs and pc regression plots</b>	<b>17</b>
<b>5</b>	<b>IgE EWAS</b>	<b>27</b>
<b>6</b>	<b>Applying Bacon to tstats</b>	<b>35</b>
<b>7</b>	<b>Regional analysis</b>	<b>38</b>
<b>8</b>	<b>Session information</b>	<b>40</b>

## 1 Setup

```

"gmodels", "tidyverse", "dplyr", "gridExtra", "isva", "sva", "base", "DMRcate", "readxl",
"SmartSVA", "corrplot", "RColorBrewer", "bacon")

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

## Warning: replacing previous import 'minfi::getMeth' by 'bsseq::getMeth' when loading
'DMRcate'

## [1] "DMRcate loaded successfully"
## [1] "meffil loaded successfully"
## [1] "ggpubr loaded successfully"
## [1] "gdata loaded successfully"
## [1] "gmodels loaded successfully"
## [1] "tidyverse loaded successfully"
## [1] "dplyr loaded successfully"
## [1] "gridExtra loaded successfully"
## [1] "isva loaded successfully"
## [1] "sva loaded successfully"
## [1] "base loaded successfully"
## [1] "DMRcate loaded successfully"
## [1] "readxl loaded successfully"
## [1] "SmartSVA loaded successfully"
## [1] "corrplot loaded successfully"

```

```

## [1] "RColorBrewer loaded successfully"
## [1] "bacon loaded successfully"

source("/udd/reprk/projects/PPG_methylation/450K_DATA/analysis_knitr/scripts/qqman.r")

# Functions for Smart-SVA analysis
# Adapted from Smart-sva tutorial page/github
# # https://rdrr.io/cran/SmartSVA/src/R/SmartSVA.R
edge.lfdr <- function (p, trunc = TRUE, monotone = TRUE,
                        transf = c("probit", "logit"), adj = 1.5, eps = 10^-8,
                        lambda = 0.8, ...)
{
  pi0 <- mean(p >= lambda)/(1 - lambda)
  pi0 <- min(pi0, 1)
  n = length(p)
  transf = match.arg(transf)
  if (transf == "probit") {
    p = pmax(p, eps)
    p = pmin(p, 1 - eps)
    x = qnorm(p)
    myd = density(x, adjust = adj)
    mys = smooth.spline(x = myd$x, y = myd$y)
    y = predict(mys, x)$y
    lfd = pi0 * dnorm(x)/y
  }
  if (transf == "logit") {
    x = log((p + eps)/(1 - p + eps))
    myd = density(x, adjust = adj)
    mys = smooth.spline(x = myd$x, y = myd$y)
    y = predict(mys, x)$y
    dx = exp(x)/(1 + exp(x))^2
    lfd = pi0 * dx/y
  }
  if (trunc) {
    lfd[lfdr > 1] = 1
  }
}

# Get 'mono' function from 'SVA' package.
mono <- getFromNamespace("mono", ns="sva")

if (monotone) {
  lfd = lfd[order(p)]
  lfd = mono(lfd)
  lfd = lfd[rank(p)]
}
return(lfd)
}

f.pval <- function (dat, orth11, orth01, y.norm, rss00, df00) {
  n <- dim(dat)[2]

  df11 <- dim(orth11)[2]
  df01 <- dim(orth01)[2]

  prj11 <- dat %*% orth11
  prj01 <- dat %*% orth01

  rss11 <- y.norm - rowSums(prj11 * prj11)
}

```

```

rss01 <- y.norm - rowSums(prj01 * prj01)

fstats <- ((rss01 - rss11)/(df11 - df01))/(rss11/(n - df11))
p1 <- 1 - pf(fstats, df1 = (df11 - df01), df2 = (n - df11))

fstats <- ((rss00 - rss01)/(df01 - df00))/(rss01/(n - df01))
p2 <- 1 - pf(fstats, df1 = (df01 - df00), df2 = (n - df01))

return(list(p1=p1, p2=p2))
}

smartsva <- function(dat, mod, mod0 = NULL, n.sv, B = 100,
                      alpha=0.25, epsilon=1e-3, VERBOSE = F) {
  if (is.null(mod0)) {
    mod0 <- mod[, 1]
  }

  qr.obj <- qr(mod)
  orth1 <- qr.Q(qr.obj)
  uu <- eigen(crossprod(dat - tcrossprod(dat %*% orth1, orth1)),
              symmetric=TRUE)$vectors[, 1:n.sv, drop=F]

  # Precompute the quantites
  y.norm <- rowSums(dat * dat)
  mod00 <- cbind(mod0)
  orth00 <- qr.Q(qr(mod00))
  prj00 <- dat %*% orth00
  rss00 <- y.norm - rowSums(prj00 * prj00)
  df00 <- dim(orth00)[2]

  if (VERBOSE)
    cat(paste("Iteration (out of", B, "):\n"))

  i = 0
  rho = 0

  while (i < B && rho < 1 - epsilon) {
    i <- i + 1
    mod11 <- cbind(mod, uu)
    mod01 <- cbind(mod0, uu)

    orth11 <- qr.Q(qr(mod11))
    orth01<- qr.Q(qr(mod01))

    ptmp <- f.pval(dat, orth11, orth01, y.norm, rss00, df00)

    if (i == 1) {
      pprob.b <- (1 - edge.lfdr(ptmp[['p1']])^alpha)
    } else {
      pprob.b <- (1 - edge.lfdr(ptmp[['p1']]))

      pprob.gam <- (1 - edge.lfdr(ptmp[['p2']]))

      pprob <- pprob.gam * (1 - pprob.b)

      uu <- eigen(crossprod(dat * pprob - rowMeans(dat * pprob)),
                  symmetric=TRUE)$vectors[, 1:n.sv, drop=F]
      # Update spearman Rho.
    }
  }
}

```

```

        if (i > 1) {
            rho <- cor(x=pprob, y=p.prev, use="pairwise.complete.obs",
                         method="spearman")
            p.prev <- pprob
        }else{
            p.prev <- pprob
        }
        if (VERBOSE)
            cat(paste(i, " ", rho, "\n"))
    }

    sv <- uu[, 1:n.sv, drop=F]
    retval <- list(sv = sv, n.sv = n.sv, pprob.gam = pprob.gam,
                   pprob.b = pprob.b, rho = rho, iter = i)
    return(retval)
}

`EstDimRMT` <-
function(data.m,plot=TRUE){
    ## standardise matrix
    M <- apply(data.m,2,function(x){ (x - mean(x))/sqrt(var(x))});

    sigma2 <- var(as.vector(M));
    Q <- nrow(data.m)/ncol(data.m);
    ns <- ncol(data.m);
    lambdaMAX <- sigma2*(1+1/Q + 2*sqrt(1/Q));
    lambdaMIN <- sigma2*(1+1/Q - 2*sqrt(1/Q));
    delta <- lambdaMAX - lambdaMIN;# print(delta);

    roundN <- 3;
    step <- round(delta/ns,roundN);
    while(step==0){
        roundN <- roundN+1;
        step <- round(delta/ns,roundN);
    }

    lambda.v <- seq(lambdaMIN,lambdaMAX,by=step);
    dens.v <- vector();
    ii <- 1;
    for(i in lambda.v){
        dens.v[ii] <- (Q/(2*pi*sigma2))*sqrt( (lambdaMAX-i)*(i-lambdaMIN) )/i;
        ii <- ii+1;
    }
    ## theoretical density
    thdens.o <- list(min=lambdaMIN,max=lambdaMAX,step=step,
                      lambda=lambda.v,dens=dens.v);
    C <- 1/nrow(M) * t(M) %*% M;
    eigen.o <- eigen(C,symmetric=TRUE);
    ## empirical density
    estdens.o <- density(eigen.o$values,from=min(eigen.o$values),
                           to=max(eigen.o$values),cut=0);
    intdim <- length(which(eigen.o$values > thdens.o$max));
    evals.v <- eigen.o$values;
    ## plot
    if(plot){
        minx <- min(min(thdens.o$lambda),min(evals.v));
        maxx <- max(max(thdens.o$lambda),max(evals.v));
    }
}

```

```

miny <- min(min(thdens.o$dens),min(estdens.o$y));
maxy <- max(max(thdens.o$dens),max(estdens.o$y));
pdf("RMTplot.pdf",width=4,height=4);
plot(thdens.o$lambda,thdens.o$dens,xlim=c(0.5,maxx),ylim=c(miny,maxy),
      type="b",col="green",xlab="Folded Eigenvalues",ylab="density",lwd=1.25);
i <- min(which(estdens.o$x > min(evalues.v)));
f <- max(which(estdens.o$x < max(evalues.v)));
points(x=estdens.o$x[i:f],y=estdens.o$y[i:f],type="b",col="red",cex=0.5);
for(i in 1:intdim){
  abline(v=evalues.v[i],col="red",lwd=2);
}
dev.off();
}

return(list(cor=C,dim=intdim,estdens=estdens.o,thdens=thdens.o,evals=eigen.o$values));
}

# smartsva.cpp
smartsva.cpp <- function(dat, mod, mod0 = NULL, n.sv, B = 100,
                           alpha=0.25, epsilon=1e-3, VERBOSE = F) {
  if (is.null(mod0)) {
    mod0 <- mod[, 1]
  }

  qr.obj <- qr(mod)
  orth1 <- qr.Q(qr.obj)
  uu <- eigs_sym(crossprodCpp(dat - tcrossprodCpp(prodCpp(dat, orth1), orth1)),
                 k=n.sv)$vectors[, 1:n.sv, drop=F]

  # Precompute the quantites
  y.norm <- rowSums(dat * dat)
  mod00 <- cbind(mod0)
  orth00 <- qr.Q(qr(mod00))
  prj00 <- prodCpp(dat, orth00)
  rss00 <- y.norm - rowSums(prj00 * prj00)
  df00 <- dim(orth00)[2]

  if (VERBOSE)
    cat(paste("Iteration (out of", B, "):\n"))

  i = 0
  rho = 0

  while (i < B && rho < 1 - epsilon) {
    i <- i + 1
    mod11 <- cbind(mod, uu)
    mod01 <- cbind(mod0, uu)

    orth11 <- qr.Q(qr(mod11))
    orth01<- qr.Q(qr(mod01))

    ptmp <- f.pval.cpp(dat, orth11, orth01, y.norm, rss00, df00)

    if (i == 1) {
      pprob.b <- (1 - edge.lfdr(ptmp[['p1']])^alpha)
    } else {
      pprob.b <- (1 - edge.lfdr(ptmp[['p1']])))
    }
  }
}

```

```

pprob.gam <- (1 - edge.lfdr(ptmp[['p2']]))

pprob <- pprob.gam * (1 - pprob.b)

uu <- eigs_sym(crossprodCpp(dat * pprob - rowMeans(dat * pprob)),
  k=n.sv)$vectors[, 1:n.sv, drop=F]
# Update spearman Rho.
if (i > 1) {
  rho <- cor(x=pprob, y=p.prev, use="pairwise.complete.obs",
    method="spearman")
  p.prev <- pprob
} else{
  p.prev <- pprob
}
if (VERBOSE)
  cat(paste(i, " ", rho, "\n"))
}

sv <- uu[, 1:n.sv, drop=FALSE]
retval <- list(sv = sv, n.sv = n.sv, pprob.gam = pprob.gam,
  pprob.b = pprob.b, rho = rho, iter = i)
return(retval)
}

# for correlations
cor.mtest <- function(mat, ...) {
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat<- matrix(NA, n, n)
  diag(p.mat) <- 0
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      tmp <- cor.test(mat[, i], mat[, j], ...)
      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
    }
  }
  colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
  p.mat
}

```

## 1.1 Packages, Data locations and loading

```

camp.dir="/proj/regeps/regep00/studies/CAMP"
results.dir = file.path(camp.dir, "analyses/reprk/methylation/results/IgE_paper")
plots.dir = file.path(results.dir, "plots")
load(file=file.path(results.dir,
  "../norm.betas.camp_hg19_clean_N0sexchr_probands_1620830489.RData"))
load(file=file.path(results.dir, "../mset.camp.funnorm_hg19_1620502108.RData"))
pData.camp <- pData(mset.camp.funnorm)
pData.camp$toe_ids <- rownames(pData.camp)
ann850k <- getAnnotation(mset.camp.funnorm)

load(file=file.path(results.dir, "../pca_betas_auto_CAMP.RData"))
pcs <- svd$x
pcs=pcs[,1:10]

load(file=file.path(results.dir,
  "../camp_EPIC_estimatecellcounts2_result_hg19_1620502108.RData"))

```

```

camp.pheno <- read.csv(file=file.path(camp.dir,
                                       "data/phenotype/camp_pheno_0421.csv"),
                        as.is=TRUE, sep=",", stringsAsFactors=FALSE)

samplesheet.camp <- read.csv(file=file.path(camp.dir,
                                             "data/epigenetic/methylation/TopMed/data/freezes/20200117/LEVEL1/SampleSheet.csv"),
                               as.is=TRUE, sep = ",", fill=T, stringsAsFactors=FALSE)

metadata <- read_excel(
  "/proj/regeps/regep00/studies/CAMP/analyses/reprk/CAMPmetadata_methylation_internal_vc.xlsx")

rm(mset.camp.funnorm) # clear memory as we don't need this object anymore

```

## 2 Data wrangling and prep for SVA

```

samplesheet.camp$Slide <- as.factor(samplesheet.camp$SentrixID)
samplesheet.camp$Sample_Plate <- as.factor(samplesheet.camp$BATCH)
samplesheet.camp$Array <- as.factor(samplesheet.camp$SentrixPosition)
samplesheet.camp$SentrixID <- NULL; samplesheet.camp$BATCH <- NULL;
samplesheet.camp$SentrixPosition <- NULL

# merge all information necessary for models to phenotype file for camp
samplesheet.camp=samplesheet.camp[,c("ALIAS","Slide","Sample_Plate","Array"),
                                 drop=FALSE]
#setnames(pData.camp, "LEVEL1.TOEID", "TOEID")
pData.camp <- merge(pData.camp, samplesheet.camp, by.x="TOEID", by.y="ALIAS",
                     sort=F)
pData.camp <- merge(pData.camp, celltype.est.2, by.x="toe_ids", by.y="row.names",
                     sort=F)
pData.camp <- merge(pData.camp, pcs, by.x="toe_ids", by.y="row.names", sort=F)

# Get subjects around F48 using metadata file read in earlier
length(unique(metadata$s_subjectid))

## [1] 1699

camp.pheno.meta <- merge(camp.pheno, metadata, by.x="S SUBJECTID",
                           by.y="s_subjectid", sort=F)
dim(camp.pheno.meta)
## [1] 833 1056

length(unique(camp.pheno.meta$S SUBJECTID))
## [1] 833

table(camp.pheno.meta$visit_code)

##
## c60 f44 f48 f52 f56 f60 t2 v2 v3m v3s v4 y2 y3
##   1 16 760 16 13 1 7 1 6 2 2 6 2

# c60 f44 f48 f52 f56 f60 t2 v2 v3m v3s v4 y2 y3
#   1 16 760 16 13 1 7 1 6 2 2 6 2
camp.pheno.meth <- camp.pheno.meta[camp.pheno.meta$visit_code=="f44" |
                                      camp.pheno.meta$visit_code=="f48" | camp.pheno.meta$visit_code=="f52",]

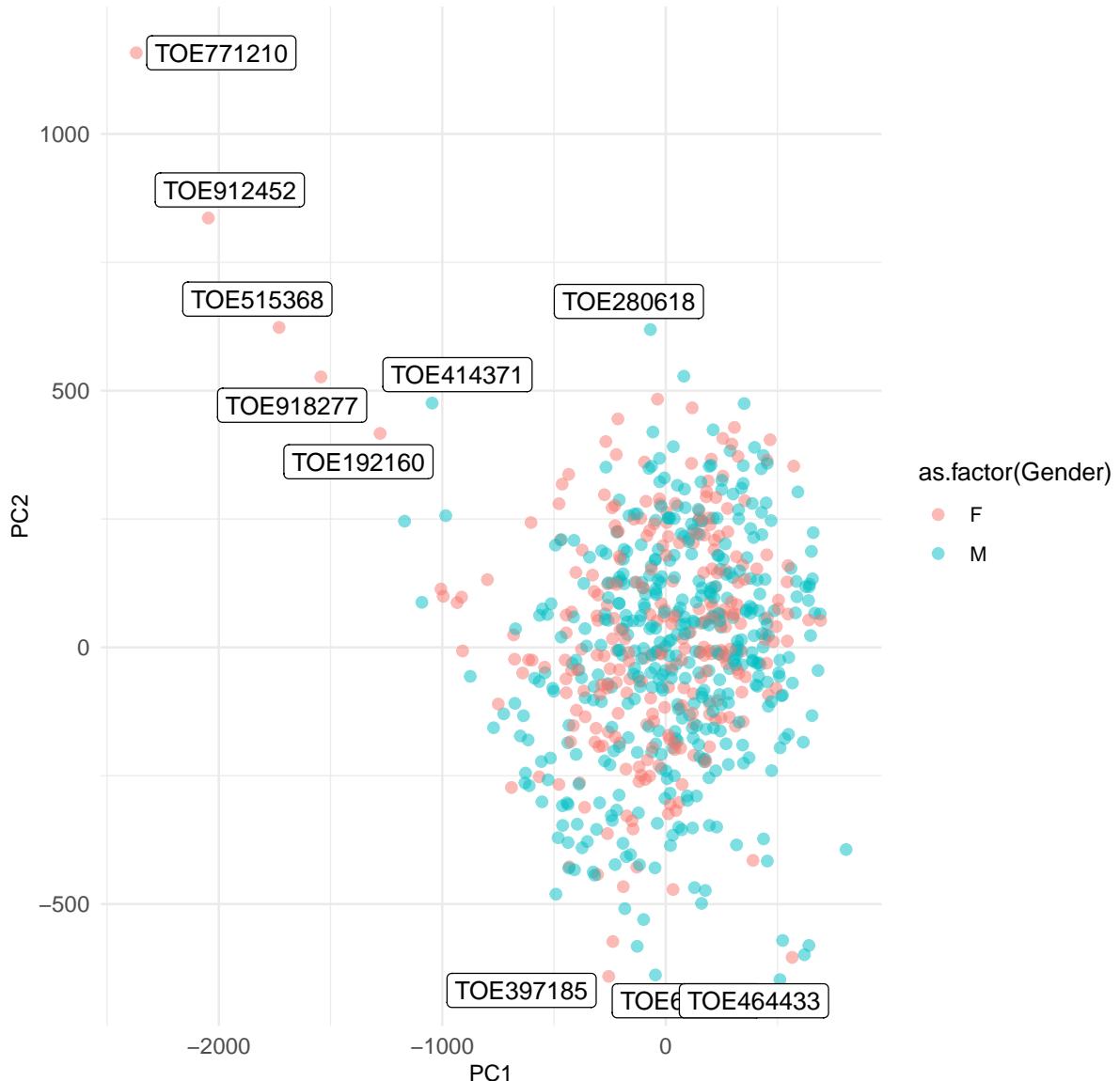
pData.pheno.meth.camp <- merge(pData.camp, camp.pheno.meth, by="S SUBJECTID", sort=F)
# 725 samples with camp.pheno, 703 with camp.pheno.meth at F44, F48, F52
dim(pData.pheno.meth.camp)

```

```
## [1] 703 1085

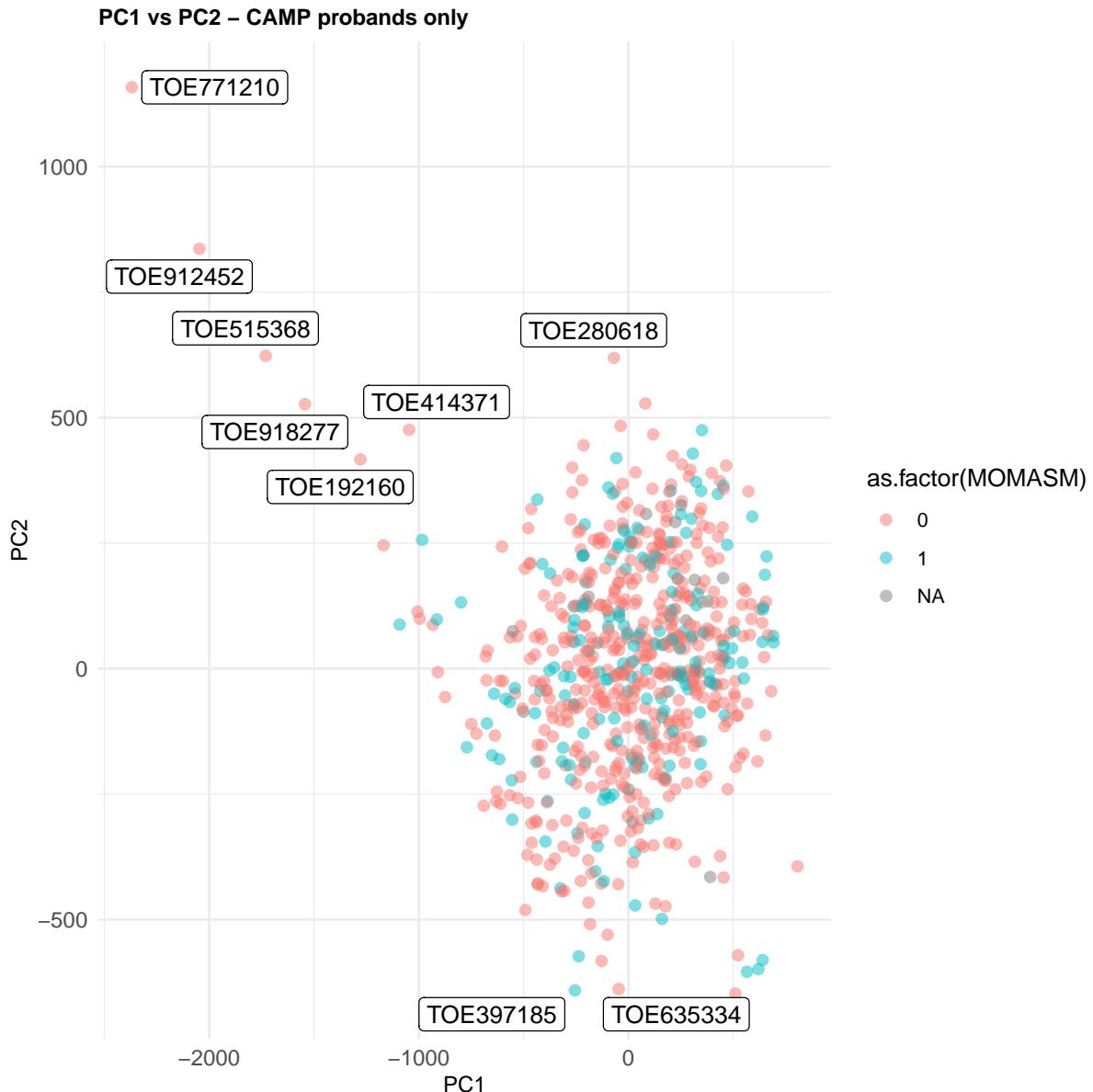
pData.pheno.meth.camp$Sample_Plate <- as.factor(pData.pheno.meth.camp$Sample_Plate)
pData.pheno.meth.camp$Slide <- as.factor(pData.pheno.meth.camp$Slide)
pData.pheno.meth.camp$Array <- as.factor(pData.pheno.meth.camp$Array)

# Check how probands cluster based on PCs
# final sample set clustering
ggplot(pData.pheno.meth.camp, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = as.factor(Gender)), alpha = 0.5, size = 2) +
  labs(title = "PC1 vs PC2 - camp probands only") +
  geom_label_repel(aes(label = 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))
```

**PC1 vs PC2 – camp probands only**

```
ggsave(path=plots.dir, "pc1_pc2_probands_gender_CAMP.png", width = 8, height = 6)
```

```
ggplot(pData.pheno.meth.camp, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = as.factor(MOMASM)), alpha = 0.5, size = 2) +
  labs(title = "PC1 vs PC2 - CAMP probands only") +
  geom_label_repel(aes(label = 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, "pc1_pc2_probands_masthma_CAMP.png", width = 8, height = 6)
```

```
# Because sample plate has too many levels so done slightly differently
p1 <- ggplot(pData.pheno.meth.camp, aes(x = PC1, y = PC2)) +
  scale_color_viridis_d() +
  labs(color = "Sample_Plate", shape = "Sample_Plate",
    title = "PC1 vs PC2 - CAMP probands only") +
  geom_label_repel(aes(label = TOEID),
```

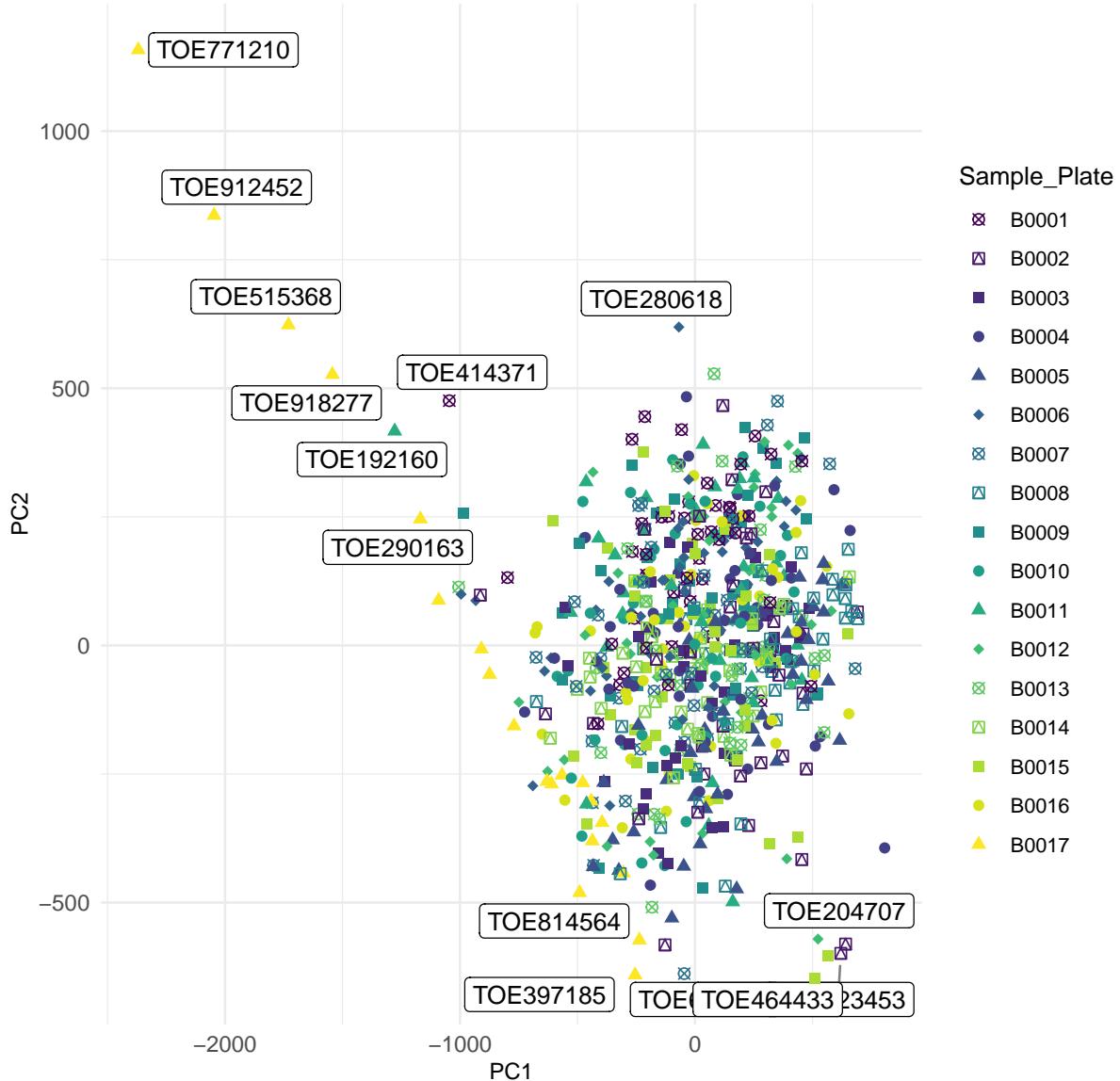
```

    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))

jit <- position_jitter(seed = 123)
p1 + geom_jitter(aes(shape = Sample_Plate,
                      color = as.factor(Sample_Plate)), size = 2, position = jit) +
  scale_shape_manual(values = rep(13:18, len = 17))

```

PC1 vs PC2 – CAMP probands only



```
ggsave(path=plots.dir, "pc1_pc2_probands_batch_CAMP.png", width = 8, height = 6)
```

```

p1 <- ggplot(pData.pheno.meth.camp, aes(x = PC1, y = PC2)) +
  scale_color_viridis_d() +
  labs(color = "Array", shape = "Array",
       title = "PC1 vs PC2 - CAMP probands only") +
  geom_label_repel(aes(label = TOEID),
                  box.padding = 0.25,

```

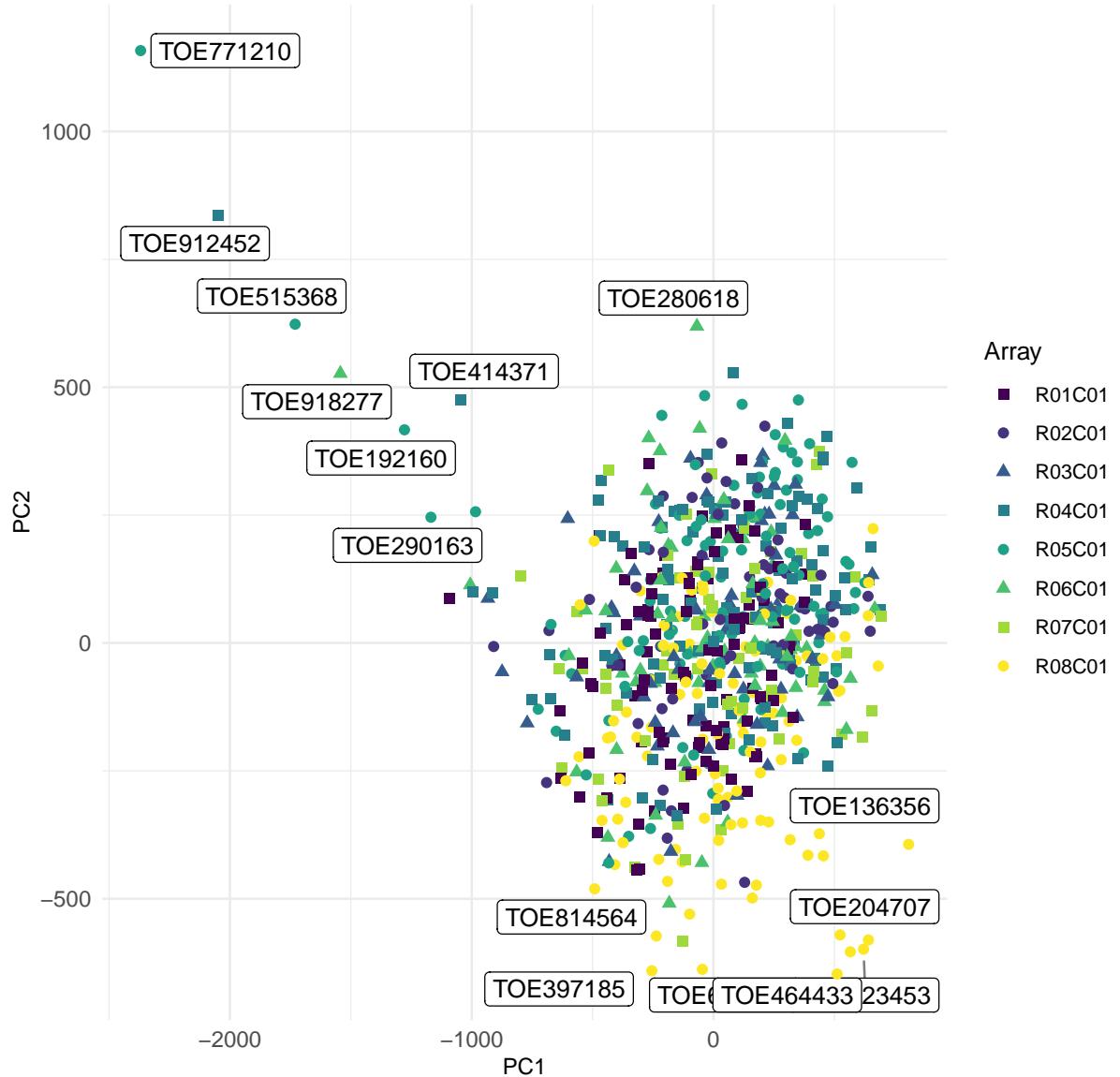
```

    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))

jit <- position_jitter(seed = 123)
p1 + geom_jitter(aes(shape = Array,
                      color = as.factor(Array)), size = 2, position = jit) +
scale_shape_manual(values = rep(15:17, len = 8))

```

PC1 vs PC2 – CAMP probands only



```
ggsave(path=plots.dir, "pc1_pc2_probands_array_CAMP.png", width = 8, height = 6)
```

```

#pData.pheno.meth.camp$age_f48 <- as.numeric(pData.pheno.meth.camp$age_f48)
pData.pheno.meth.camp$age_f48 <- as.numeric(pData.pheno.meth.camp$age_f48.x)
pData.pheno.meth.camp$Gender <- as.factor(pData.pheno.meth.camp$Gender)
pData.pheno.meth.camp$BMI_F48 <- as.numeric(pData.pheno.meth.camp$BMI_F48)
pData.pheno.meth.camp$HTCM_F48 <- as.numeric(pData.pheno.meth.camp$HTCM_F48)
pData.pheno.meth.camp$MOMASM <- as.factor(pData.pheno.meth.camp$MOMASM)
pData.pheno.meth.camp$ETS <- as.factor(pData.pheno.meth.camp$ETS)

```

```

pData.pheno.meth.camp$growth_pattern <- as.factor(pData.pheno.meth.camp$growth_pattern)
pData.pheno.meth.camp$TG <- pData.pheno.meth.camp$TG
pData.pheno.meth.camp$LOG10IGE_iuml_F48 <- as.numeric(pData.pheno.meth.camp$LOG10IGE_iuml_F48)
# # RACE, 1=white, 2= black, 3 = hispanic; 4= other
pData.pheno.meth.camp$RACE <- as.factor(pData.pheno.meth.camp$RACE)

which(is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48)) # 29 samples

## [1] 18 55 56 72 83 125 151 185 203 206 210 266 268 301 337 375 385 464 468
## [20] 469 497 555 560 618 646 667 670 675 698

pData.pheno.meth.camp[is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48),]$toe_ids

## [1] "TOE926201-BIS-v01_R06C01" "TOE457685-BIS-v01_R01C01"
## [3] "TOE629851-BIS-v01_R04C01" "TOE676134-BIS-v01_R06C01"
## [5] "TOE243871-BIS-v01_R01C01" "TOE563668-BIS-v01_R07C01"
## [7] "TOE684478-BIS-v01_R08C01" "TOE586851-BIS-v01_R07C01"
## [9] "TOE195764-BIS-v01_R04C01" "TOE115535-BIS-v01_R05C01"
## [11] "TOE495280-BIS-v01_R05C01" "TOE590448-BIS-v01_R08C01"
## [13] "TOE303413-BIS-v01_R05C01" "TOE317434-BIS-v01_R01C01"
## [15] "TOE222500-BIS-v01_R01C01" "TOE389516-BIS-v01_R05C01"
## [17] "TOE280068-BIS-v01_R06C01" "TOE997567-BIS-v01_R01C01"
## [19] "TOE164255-BIS-v01_R05C01" "TOE917208-BIS-v01_R05C01"
## [21] "TOE941964-BIS-v01_R01C01" "TOE615791-BIS-v01_R04C01"
## [23] "TOE723220-BIS-v01_R08C01" "TOE108743-BIS-v01_R08C01"
## [25] "TOE578417-BIS-v01_R08C01" "TOE180726-BIS-v01_R04C01"
## [27] "TOE507162-BIS-v01_R05C01" "TOE968046-BIS-v01_R03C01"
## [29] "TOE687042-BIS-v01_R04C01"

pData.pheno.meth.camp[is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48),]$S SUBJECTID

## [1] "ST-00063815" "ST-00064911" "ST-00063720" "ST-00065640" "ST-00065282"
## [6] "ST-00061940" "ST-00061823" "ST-00061953" "ST-00065461" "ST-00063739"
## [11] "ST-00063637" "ST-00064782" "ST-00061368" "ST-00061205" "ST-00062076"
## [16] "ST-00063289" "ST-00063966" "ST-00062516" "ST-00062709" "ST-00064231"
## [21] "ST-00062523" "ST-00064904" "ST-00061432" "ST-00065281" "ST-00065478"
## [26] "ST-00063979" "ST-00063709" "ST-00065457" "ST-00060682"

which(is.na(pData.pheno.meth.camp$BMI_F48)) # 18 samples

## [1] 72 83 206 210 218 266 268 275 375 437 464 468 469 497 618 646 667 670

pData.pheno.meth.camp[is.na(pData.pheno.meth.camp$BMI_F48),]$toe_ids

## [1] "TOE676134-BIS-v01_R06C01" "TOE243871-BIS-v01_R01C01"
## [3] "TOE115535-BIS-v01_R05C01" "TOE495280-BIS-v01_R05C01"
## [5] "TOE670023-BIS-v01_R05C01" "TOE590448-BIS-v01_R08C01"
## [7] "TOE303413-BIS-v01_R05C01" "TOE715195-BIS-v01_R08C01"
## [9] "TOE389516-BIS-v01_R05C01" "TOE340633-BIS-v01_R03C01"
## [11] "TOE997567-BIS-v01_R01C01" "TOE164255-BIS-v01_R05C01"
## [13] "TOE917208-BIS-v01_R05C01" "TOE941964-BIS-v01_R01C01"
## [15] "TOE108743-BIS-v01_R08C01" "TOE578417-BIS-v01_R08C01"
## [17] "TOE180726-BIS-v01_R04C01" "TOE507162-BIS-v01_R05C01"

which(is.na(pData.pheno.meth.camp$MOMASM)) # 16 samples

## [1] 95 122 135 175 202 233 254 290 421 424 507 513 530 560 682 683

pData.pheno.meth.camp[is.na(pData.pheno.meth.camp$MOMASM),]$toe_ids

```

```

## [1] "TOE424193-BIS-v01_R01C01" "TOE380009-BIS-v01_R08C01"
## [3] "TOE372699-BIS-v01_R08C01" "TOE790435-BIS-v01_R05C01"
## [5] "TOE601238-BIS-v01_R02C01" "TOE564960-BIS-v01_R08C01"
## [7] "TOE823222-BIS-v01_R01C01" "TOE433587-BIS-v01_R04C01"
## [9] "TOE974671-BIS-v01_R02C01" "TOE903274-BIS-v01_R01C01"
## [11] "TOE856005-BIS-v01_R04C01" "TOE972486-BIS-v01_R05C01"
## [13] "TOE852221-BIS-v01_R04C01" "TOE723220-BIS-v01_R08C01"
## [15] "TOE251257-BIS-v01_R04C01" "TOE213957-BIS-v01_R05C01"

which(is.na(pData.pheno.meth.camp$ETS)) # 3 samples

## [1] 90 137 562

pData.pheno.meth.camp[is.na(pData.pheno.meth.camp$ETS),]$toe_ids

## [1] "TOE488455-BIS-v01_R02C01" "TOE607179-BIS-v01_R05C01"
## [3] "TOE470013-BIS-v01_R01C01"

pData.pheno.meth.camp <- pData.pheno.meth.camp[!is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48),]

# standardizing to make it consistent with CRA as discussed with Dr. Dawn L. DeMeo
summary(pData.pheno.meth.camp$LOG10IGE_iuml_F48)

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -0.08619   1.90725   2.31001   2.24911   2.66893   4.04998

# 2 subjects
length(pData.pheno.meth.camp$LOG10IGE_iuml_F48[pData.pheno.meth.camp$LOG10IGE_iuml_F48<0])

## [1] 2

pData.pheno.meth.camp[pData.pheno.meth.camp$LOG10IGE_iuml_F48<0,]$S SUBJECTID

## [1] "ST-00064884" "ST-00062782"

pData.pheno.meth.camp$LOG10IGE_iuml_F48[pData.pheno.meth.camp$LOG10IGE_iuml_F48<0] <- 0
summary(pData.pheno.meth.camp$LOG10IGE_iuml_F48)

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## 0.000   1.907   2.310   2.249   2.669   4.050

# correction to these values as clarified by Soma
pData.pheno.meth.camp$PRE2575FVC_f48 <- pData.pheno.meth.camp$PRE2575FVC_f48*100
pData.pheno.meth.camp$POS2575FVC_f48 <- pData.pheno.meth.camp$POS2575FVC_f48*100

beta.ewas <- norm.betas.rcp.auto.prob[, colnames(norm.betas.rcp.auto.prob)
                                         %in% pData.pheno.meth.camp$toe_ids]

ewas_var <- pData.pheno.meth.camp$LOG10IGE_iuml_F48

pData.pheno.meth.camp$sex[pData.pheno.meth.camp$Gender=="F"]<-0;
pData.pheno.meth.camp$sex[pData.pheno.meth.camp$Gender=="M"]<-1;
pData.pheno.meth.camp$sex <- as.factor(pData.pheno.meth.camp$sex)
pData.pheno.meth.camp$Plate <- sapply(as.character(pData.pheno.meth.camp$Sample_Plate),
switch, "B0001"=1, "B0002"=2, "B0003"=3, "B0004"=4, "B0005"=5, "B0006"=6, "B0007"=7,
"B0008"=8, "B0009"=9, "B0010"=10, "B0011"=11, "B0012"=12, "B0013"=13, "B0014"=14,
"B0015"=15, "B0016"=16, "B0017"=17, USE.NAMES = F)
pData.pheno.meth.camp$Plate <- as.factor(pData.pheno.meth.camp$Plate)
#pData.pheno.meth.camp$Slide <- as.numeric(pData.pheno.meth.camp$Slide)
pData.pheno.meth.camp$Array <- as.factor(pData.pheno.meth.camp$Array)

covs=pData.pheno.meth.camp[,c("Plate", "Array"), drop=FALSE]
summary(covs)

```

```

##      Plate      Array
## 16    : 45    R04C01 :113
## 1     : 44    R05C01 :106
## 5     : 44    R08C01 :103
## 8     : 43    R01C01 :100
## 4     : 42    R06C01 : 68
## 6     : 42    R02C01 : 65
## (Other):414 (Other):119

# this step not needed anymore as betas only contain autosomes for this IgE EWAS
#autosomal.sites <- meffil.get.autosomal.sites("epic")
#autosomal.sites <- intersect(autosomal.sites, rownames(beta.ewas))
#beta.sva <- beta.ewas[autosomal.sites,]

# should not contain NAs but just as a sanity check
dim(beta.ewas) # 785352    674

## [1] 785352    674

beta.sva <- na.omit(beta.ewas)
dim(beta.sva) # 785352    674

## [1] 785352    674

# using betas to calculate SVs to keep on same scale
#M.ewas <- logit2(beta.sva)
#M.ewas <- as.matrix(beta.sva)

cov.frame <- model.frame(~., data.frame(covs, stringsAsFactors=F), na.action=na.pass)

# null model only with technical batch variables
mod0 <- model.matrix(~., cov.frame)

pheno.sel=pData.pheno.meth.camp[,c("LOG10IGE_iuml_F48","Plate","Array"),
                                 drop=FALSE]
rownames(pheno.sel) <- pData.pheno.meth.camp$toe_ids

# checking all variables are as factor
which(sapply(pheno.sel, function(x) (is.character(x) | is.factor(x)) & length(unique(x))<2))

## named integer(0)

```

### 3 SVA estimation

```

# for reproducibility
# for reproducibility
set.seed(123456)
mod.res <- t(resid(lm(t(beta.sva) ~ ., data=as.data.frame(pheno.sel))))
n.sv <- EstDimRMT(mod.res, FALSE)$dim + 1 # 48
n.sv

## [1] 48

# Full model
mod <- model.matrix( ~ LOG10IGE_iuml_F48 + Plate + Array, data=pheno.sel)

smartsava.ret <- smartsava(beta.sva, mod=mod, mod0=mod0, n.sv=n.sv)
smartsava.sv <- as.data.frame(smartsava.ret$sv)
rownames(smartsava.sv) <- pData.pheno.meth.camp$toe_ids

```

```

save(smartsava.ret, smartsava.sv,
      file=file.path(results.dir,paste0("CAMP_smartsava_results_IgE_",
                                         timeStamp,".RData")))

colnames(smartsava.sv) <- gsub(x = colnames(smartsava.sv),
                                pattern = "\\\V", replacement = "SV")
smartsava.sv <- smartsava.sv[,1:10]
smartsava.sv[1:2,]

##                                     SV1          SV2          SV3          SV4
## TOE654293-BIS-v01_R04C01 -0.002979925 -0.01723315  0.03957017 -0.007675175
## TOE939881-BIS-v01_R05C01 -0.036716840  0.04333561  0.03008778  0.016450772
##                                     SV5          SV6          SV7          SV8
## TOE654293-BIS-v01_R04C01  0.06922376  0.02698026  0.01738379  0.02130087
## TOE939881-BIS-v01_R05C01 -0.02426289  0.01148598  0.01305332  0.026667900
##                                     SV9          SV10
## TOE654293-BIS-v01_R04C01 -0.02611746  0.003484097
## TOE939881-BIS-v01_R05C01 -0.04150205 -0.012291291

pData.pheno.meth.camp <- merge(pData.pheno.meth.camp,
                                 smartsava.sv, by.x="toe_ids", by.y="row.names", sort=F)

save(beta.ewas, pData.pheno.meth.camp,
      file=file.path(results.dir,paste0("CAMP_betas_pheno_forIgE.EWAS_",
                                         timeStamp,".RData")))
beta.sva <- NULL

```

## 4 Correlations PCs and SVs and pc regression plots

```

cor.test(pData.pheno.meth.camp$PC1, pData.pheno.meth.camp$LOG10IGE_iuML_F48) # not sig

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$PC1 and pData.pheno.meth.camp$LOG10IGE_iuML_F48
## t = 0.10293, df = 672, p-value = 0.918
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.07157034 0.07946631
## sample estimates:
##       cor
## 0.003970633

cor.test(pData.pheno.meth.camp$PC2, pData.pheno.meth.camp$LOG10IGE_iuML_F48) # not sig

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$PC2 and pData.pheno.meth.camp$LOG10IGE_iuML_F48
## t = -0.45157, df = 672, p-value = 0.6517
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.09281453 0.05817893
## sample estimates:
##       cor
## -0.0174171

cor.test(pData.pheno.meth.camp$PC3, pData.pheno.meth.camp$LOG10IGE_iuML_F48) # not sig

```

```

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.meth.camp$PC3 and pData.pheno.meth.camp$LOG10IGE_iuml_F48
## t = -0.031042, df = 672, p-value = 0.9752
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.07671004 0.07432877
## sample estimates:
##          cor
## -0.001197464

cor.test(pData.pheno.meth.camp$PC1, as.numeric(pData.pheno.meth.camp$Slide))

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.meth.camp$PC1 and as.numeric(pData.pheno.meth.camp$Slide)
## t = -3.1408, df = 672, p-value = 0.001759
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.1940371 -0.0451709
## sample estimates:
##          cor
## -0.1202801

cor.test(pData.pheno.meth.camp$PC2, as.numeric(pData.pheno.meth.camp$Slide))

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.meth.camp$PC2 and as.numeric(pData.pheno.meth.camp$Slide)
## t = 4.0322, df = 672, p-value = 6.158e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.07909735 0.22658820
## sample estimates:
##          cor
## 0.1536988

cor.test(pData.pheno.meth.camp$PC1, as.numeric(pData.pheno.meth.camp$Array))

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.meth.camp$PC1 and as.numeric(pData.pheno.meth.camp$Array)
## t = 1.8606, df = 672, p-value = 0.06324
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.00395065 0.14631866
## sample estimates:
##          cor
## 0.07159022

cor.test(pData.pheno.meth.camp$PC2, as.numeric(pData.pheno.meth.camp$Array))

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.meth.camp$PC2 and as.numeric(pData.pheno.meth.camp$Array)

```

```

## t = -4.536, df = 672, p-value = 6.79e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.2446951 -0.0981184
## sample estimates:
##      cor
## -0.1723607

cor.test(pData.pheno.meth.camp$PC1, as.numeric(pData.pheno.meth.camp$Plate))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$PC1 and as.numeric(pData.pheno.meth.camp$Plate)
## t = -4.1067, df = 672, p-value = 4.509e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.22927685 -0.08191496
## sample estimates:
##      cor
## -0.1564665

cor.test(pData.pheno.meth.camp$PC2, as.numeric(pData.pheno.meth.camp$Plate))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$PC2 and as.numeric(pData.pheno.meth.camp$Plate)
## t = -1.6064, df = 672, p-value = 0.1087
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.13672976 0.01373477
## sample estimates:
##      cor
## -0.06184889

# we could go upto 6 or 7 SVs in CAMP
cor.test(pData.pheno.meth.camp$SV1, pData.pheno.meth.camp$LOG10IGE_iu碌_F48) # not sig

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV1 and pData.pheno.meth.camp$LOG10IGE_iu碌_F48
## t = -0.19106, df = 672, p-value = 0.8485
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.08284364 0.06818722
## sample estimates:
##      cor
## -0.007370245

cor.test(pData.pheno.meth.camp$SV2, pData.pheno.meth.camp$LOG10IGE_iu碌_F48) # not sig

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV2 and pData.pheno.meth.camp$LOG10IGE_iu碌_F48
## t = -0.17659, df = 672, p-value = 0.8599
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:

```

```

## -0.08228908 0.06874297
## sample estimates:
## cor
## -0.006811901

cor.test(pData.pheno.meth.camp$SV3, pData.pheno.meth.camp$LOG10IGE_iuml_F48)

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV3 and pData.pheno.meth.camp$LOG10IGE_iuml_F48
## t = -0.15279, df = 672, p-value = 0.8786
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.0813773 0.0696565
## sample estimates:
## cor
## -0.00589401

cor.test(pData.pheno.meth.camp$SV1, as.numeric(pData.pheno.meth.camp$Plate))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV1 and as.numeric(pData.pheno.meth.camp$Plate)
## t = -3.5935, df = 672, p-value = 0.0003502
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.21064339 -0.06243605
## sample estimates:
## cor
## -0.1373081

cor.test(pData.pheno.meth.camp$SV2, as.numeric(pData.pheno.meth.camp$Plate))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV2 and as.numeric(pData.pheno.meth.camp$Plate)
## t = -0.21116, df = 672, p-value = 0.8328
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.08361350 0.06741555
## sample estimates:
## cor
## -0.008145427

cor.test(pData.pheno.meth.camp$SV3, as.numeric(pData.pheno.meth.camp$Plate))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV3 and as.numeric(pData.pheno.meth.camp$Plate)
## t = -3.0792, df = 672, p-value = 0.00216
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.19176322 -0.04281373
## sample estimates:
## cor
## -0.1179519

```

```

cor.test(pData.pheno.meth.camp$SV1, as.numeric(pData.pheno.meth.camp$Slide))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV1 and as.numeric(pData.pheno.meth.camp$Slide)
## t = -1.7451, df = 672, p-value = 0.08143
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.141965394 0.008396116
## sample estimates:
## cor
## -0.06716598

cor.test(pData.pheno.meth.camp$SV2, as.numeric(pData.pheno.meth.camp$Slide))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV2 and as.numeric(pData.pheno.meth.camp$Slide)
## t = 1.1204, df = 672, p-value = 0.263
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.03244646 0.11831256
## sample estimates:
## cor
## 0.04317885

cor.test(pData.pheno.meth.camp$SV3, as.numeric(pData.pheno.meth.camp$Slide))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV3 and as.numeric(pData.pheno.meth.camp$Slide)
## t = 4.6003, df = 672, p-value = 5.043e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.1005367 0.2469899
## sample estimates:
## cor
## 0.1747296

cor.test(pData.pheno.meth.camp$SV1, as.numeric(pData.pheno.meth.camp$array))

##
## Pearson's product-moment correlation
##
## data: pData.pheno.meth.camp$SV1 and as.numeric(pData.pheno.meth.camp$array)
## t = 1.0455, df = 672, p-value = 0.2962
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.03532985 0.11546529
## sample estimates:
## cor
## 0.04029717

cor.test(pData.pheno.meth.camp$SV2, as.numeric(pData.pheno.meth.camp$array))

##
## Pearson's product-moment correlation

```

```

## 
## data: pData.pheno.meth.camp$SV2 and as.numeric(pData.pheno.meth.camp$array)
## t = 0.32285, df = 672, p-value = 0.7469
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.06312561 0.08789012
## sample estimates:
## cor
## 0.01245326

cor.test(pData.pheno.meth.camp$SV3, as.numeric(pData.pheno.meth.camp$array))

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.meth.camp$SV3 and as.numeric(pData.pheno.meth.camp$array)
## t = -3.2826, df = 672, p-value = 0.001082
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.19925495 -0.05058622
## sample estimates:
## cor
## -0.1256258

cor.data=pData.pheno.meth.camp[,c("LOG10IGE_iuml_F48","Plate", "Slide",
  "Array", "PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9", "PC10", "SV1", "SV2", "SV3",
  "SV4", "SV5", "SV6", "SV7", "SV8", "SV9", "SV10"), drop=FALSE]
cor.data <- as.data.frame(sapply(cor.data, as.numeric))
M<-cor(cor.data)
head(round(M,2))

##          LOG10IGE_iuml_F48 Plate Slide Array   PC1   PC2   PC3   PC4
## LOG10IGE_iuml_F48        1.00 -0.01 -0.05  0.01  0.00 -0.02  0.00  0.00
## Plate                  -0.01  1.00 -0.54 -0.04 -0.16 -0.06 -0.02 -0.14
## Slide                  -0.05 -0.54  1.00  0.03 -0.12  0.15  0.04  0.02
## Array                  0.01 -0.04  0.03  1.00  0.07 -0.17  0.00  0.00
## PC1                   0.00 -0.16 -0.12  0.07  1.00 -0.01  0.00  0.00
## PC2                   -0.02 -0.06  0.15 -0.17 -0.01  1.00 -0.01 -0.01
##                      PC5   PC6   PC7   PC8   PC9   PC10  SV1   SV2   SV3   SV4
## LOG10IGE_iuml_F48 -0.04 -0.04 -0.03 -0.06 -0.08  0.05 -0.01 -0.01 -0.01  0.02
## Plate                 0.05 -0.12 -0.04  0.07  0.06  0.15 -0.14 -0.01 -0.12 -0.11
## Slide                 0.02 -0.02  0.00 -0.09  0.01 -0.08 -0.07  0.04  0.17 -0.10
## Array                 -0.03 -0.23  0.09 -0.02 -0.08 -0.03  0.04  0.01 -0.13  0.06
## PC1                  0.01  0.00  0.00  0.01  0.01  0.02  0.94  0.00 -0.17  0.24
## PC2                  0.01  0.00  0.01  0.00  0.01  0.00  0.29  0.07  0.80 -0.43
##                      SV5   SV6   SV7   SV8   SV9   SV10
## LOG10IGE_iuml_F48  0.01 -0.02  0.04  0.12  0.07  0.04
## Plate                 -0.07 -0.12  0.09 -0.04 -0.09  0.08
## Slide                 0.03 -0.02 -0.08  0.08  0.05 -0.08
## Array                 -0.05 -0.28 -0.03  0.05  0.03  0.00
## PC1                  -0.06  0.08  0.03  0.08 -0.02 -0.01
## PC2                  0.12  0.05 -0.02 -0.11  0.04  0.02

# matrix of the p-value of the correlation
p.mat <- cor.mtest(cor.data)
head(p.mat[, 1:5])

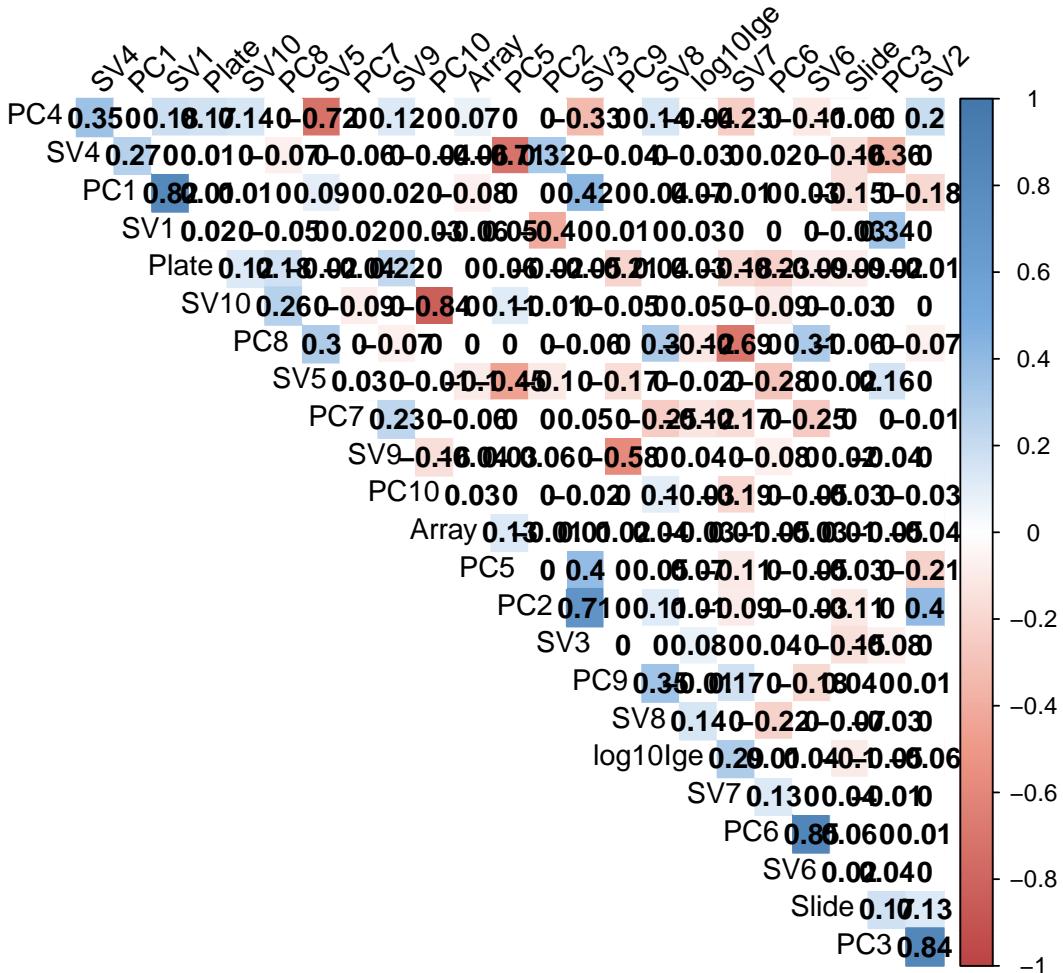
##          LOG10IGE_iuml_F48         Plate         Slide         Array
## LOG10IGE_iuml_F48        0.0000000 8.926535e-01 1.957457e-01 8.052608e-01
## Plate                     0.8926535 0.000000e+00 8.741247e-52 3.103990e-01

```

```
## Slide          0.1957457 8.741247e-52 0.000000e+00 4.188345e-01
## Array          0.8052608 3.103990e-01 4.188345e-01 0.000000e+00
## PC1           0.9180481 4.508975e-05 1.758682e-03 6.323662e-02
## PC2           0.6517234 1.086598e-01 6.157924e-05 6.790218e-06
##               PC1
## LOG10IGE_iuml_F48 9.180481e-01
## Plate          4.508975e-05
## Slide          1.758682e-03
## Array          6.323662e-02
## PC1           0.000000e+00
## PC2           8.641013e-01

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

corrplot(M, method="color", col=col(200),
         type="upper", order="hclust",
         addCoef.col = "black", # Add coefficient of correlation
         tl.col="black", tl.srt=45, #Text label color and rotation
         # Combine with significance
         p.mat = p.mat, sig.level = 0.05, insig = "blank",
         # hide correlation coefficient on the principal diagonal
         diag=FALSE
         )
```



```

pdf(file = file.path(plots.dir, "correlogram_CAMP_pcs_svs_log10Ige.pdf"),
width = 16, height = 15)
corrplot(M, method="color", col=col(200),
type="upper", order="hclust",
addCoef.col = "black", # Add coefficient of correlation
tl.col="black", tl.srt=45, #Text label color and rotation
# Combine with significance
p.mat = p.mat, sig.level = 0.05, insig = "blank",
# hide correlation coefficient on the principal diagonal
diag=FALSE
)
dev.off()

## pdf
## 2

formula <- c(~LOG10IGE_F48+age_f48+Gender+BMI_F48+HTCM_F48+ETS+TG+
growth_pattern+MOMASM+LOG10IGE_iuml_F48+PREFEV_F48+POSFEV_F48+
PREFVC_F48+POSFVC_F48+PREFF_F48+POSFF_F48+PRE2575FVC_f48+POS2575FVC_f48+
Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Slide+Array+PC1+PC2+SV1+SV2")

```

```

remove <- c(list(which(is.na(pData.pheno.meth.camp$age_f48) |
  is.na(pData.pheno.meth.camp$PREFEV_F48) |
  is.na(pData.pheno.meth.camp$POSFEV_F48) |
  is.na(pData.pheno.meth.camp$PREFVC_F48) |
  is.na(pData.pheno.meth.camp$POSFVC_F48) |
  is.na(pData.pheno.meth.camp$PREFF_F48) |
  is.na(pData.pheno.meth.camp$POSFF_F48) |
  is.na(pData.pheno.meth.camp$PRE2575FVC_f48) |
  is.na(pData.pheno.meth.camp$POS2575FVC_f48) |
  is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
  is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
  is.na(pData.pheno.meth.camp$Gender) | is.na(pData.pheno.meth.camp$BMI_F48) |
  is.na(pData.pheno.meth.camp$HTCM_F48)))
length(remove[[1]]) # 34 samples

## [1] 34

pDat.tmp <- pData.pheno.meth.camp[!(rownames(pData.pheno.meth.camp)
  %in% remove[[1]]),]

ids.keep <- intersect(pDat.tmp$toe_ids, colnames(beta.ewas))
betas.pcr=beta.ewas[,which(colnames(beta.ewas) %in% ids.keep)]
dim(beta.ewas) # 785352      674

## [1] 785352      674

dim(betas.pcr) # 785352      640

## [1] 785352      640

pheno.sel=pDat.tmp[,c("age_f48", "Gender", "TG", "LOG10IGE_iuml_F48",
  "BMI_F48", "HTCM_F48", "MOMASM", "RACE", "ETS", "PREFEV_F48", "POSFEV_F48",
  "PREFVC_F48", "POSFVC_F48", "PREFF_F48", "POSFF_F48", "PRE2575FVC_f48",
  "POS2575FVC_f48", "growth_pattern", "Bcell", "CD4T", "CD8T", "Mono",
  "Neu", "NK", "Plate", "Slide", "Array", "PC1", "PC2", "SV1", "SV2"), drop=FALSE];
pheno.sel <- data.frame(pheno.sel);
table(is.na(pheno.sel)) # there shouldn't be any missingness for this plot

##
## FALSE
## 19840

# Top 10 principal components can explain ~37 % of data variation
pcrplot(betas.pcr, pheno.sel, npc=10)

## Analysis is running, please wait...!
## svdscreeplot.jpg was plotted
## Top 10 principal components can explain 36.61516 % of data
##      variation
## pcr_diag.jpg was plotted

file.rename(list.files(pattern="pcr_diag*.jpg"), paste0("pcr_diag_IgE_CAMP.jpg"))

## [1] TRUE

file.copy("pcr_diag_IgE_CAMP.jpg", plots.dir)

## [1] TRUE

rm(betas.pcr)

```

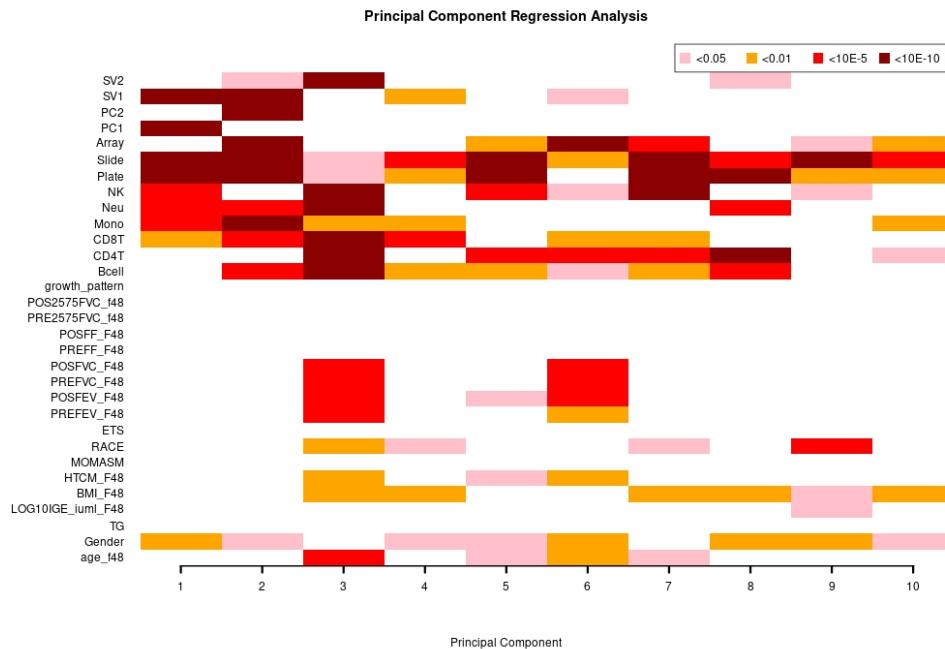


Figure 1: PC regression plot against all phenotypes CAMP

## 5 IgE EWAS

```

# we can use beta.ewas
#ids.keep <- intersect(pData.pheno.meth.camp$toe_ids, colnames(beta.ewas))
#betas.final=beta.ewas[,which(colnames(beta.ewas) %in% ids.keep)]
pData.pheno.meth.camp$Plate <- as.factor(pData.pheno.meth.camp$Plate)
pData.pheno.meth.camp$Slide <- as.factor(pData.pheno.meth.camp$Slide)
pData.pheno.meth.camp$array <- as.factor(pData.pheno.meth.camp$array)

betas.ann850k <- merge(beta.ewas, ann850k, by.x="row.names", by.y="Name", sort=F)
betas.ann850k[1:2,675:678]

## TOE406321-BIS-v01_R08C01 chr pos strand
## 1 0.9694943 chr1 10848 -
## 2 0.9067561 chr1 10850 -

betas.ann850k$CGsite <- betas.ann850k$Row.names
rownames(betas.ann850k) <- betas.ann850k$CGsite
dim(betas.ann850k)

## [1] 785352    721

betas.ann850k <- betas.ann850k[,676:721]
betas.ann850k$Gene <- sub(";.*", "", betas.ann850k$UCSC_RefGene_Name)

# CAMP
# model1: 15,131, 206 overlap with FHS; 198 including CRA
# model2: 15,339, 204 overlap with FHS
# model3: 12,093, 204 overlap with FHS
# model4: 15,391, 207 overlap with FHS; 199 including CRA
# model5: 14,223, 206 overlap with FHS; 193 including CRA
# model6: 14,735, 204 overlap with FHS
# model7: 12,799, 201 overlap with FHS
# model8: 15,075, 206 overlap with FHS; 194 including CRA

formula <- c("`~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+PC1+PC2`",
"~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2",
"~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+PC1+PC2",
"~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+PC1+PC2+PC3",
"~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+SV1+SV2",
"~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+SV1+SV2",
"~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2+SV3",
"~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2")

vars.of.interest <- c("LOG10IGE_iuml_F48", "LOG10IGE_iuml_F48", "LOG10IGE_iuml_F48", "LOG10IGE_iuml_F48",
varNames <- c("LOG10IGE_iuml_F48_ct_2PCs", "LOG10IGE_iuml_F48_ct_plt_arr_2PCs", "LOG10IGE_iuml_F48_noPCs")

remove <- c(list(which(is.na(pData.pheno.meth.camp$age_f48) |
is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
is.na(pData.pheno.meth.camp$Gender))),,
list(which(is.na(pData.pheno.meth.camp$age_f48) |
is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
is.na(pData.pheno.meth.camp$Gender))),,
list(which(is.na(pData.pheno.meth.camp$age_f48) |
is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
is.na(pData.pheno.meth.camp$Gender))),,
list(which(is.na(pData.pheno.meth.camp$age_f48) |
is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
is.na(pData.pheno.meth.camp$Gender))),,
list(which(is.na(pData.pheno.meth.camp$age_f48)) |
is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
is.na(pData.pheno.meth.camp$Gender)))

```

```

        is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
        is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
        is.na(pData.pheno.meth.camp$Gender))),
      list(which(is.na(pData.pheno.meth.camp$age_f48) |
        is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
        is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
        is.na(pData.pheno.meth.camp$Gender))),
      list(which(is.na(pData.pheno.meth.camp$age_f48) |
        is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
        is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
        is.na(pData.pheno.meth.camp$Gender))),
      list(which(is.na(pData.pheno.meth.camp$age_f48) |
        is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
        is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
        is.na(pData.pheno.meth.camp$Gender))),
      list(which(is.na(pData.pheno.meth.camp$age_f48) |
        is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
        is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
        is.na(pData.pheno.meth.camp$Gender)))
    )
}

for (f in 1:length(formula)) {
  # remove subjects if there is a missing value in a variables of interest
  if (length(remove[[f]])!=0) {

    pDat.tmp <- pData.pheno.meth.camp[-remove[[f]],]
    betas.tmp <- beta.ewas[,-remove[[f]]]

  } else {

    pDat.tmp <- pData.pheno.meth.camp
    betas.tmp <- beta.ewas
  }

  print("pDat.tmp dims:")
  print(dim(pDat.tmp))
  print("betas.tmp dims:")
  print(dim(betas.tmp))

  design <- model.matrix(as.formula(formula[f]), data=pDat.tmp)
  print("")
  print(formula[f])
  print("")
  pData.tmp$Gender <- relevel(pData.tmp$Gender, ref='F')

  fit <- limma::lmFit(betas.tmp, design)
  fit <- limma::eBayes(fit)

  topHits <- limma::topTable(fit, coef=2, num=Inf, adjust.method="BH",
                               genelist = betas.ann850k)
  print(summary(decideTests(fit)))
  print("significant probes (bonferroni threshold):")
  print(table(topHits[, 'P.Value']<1.431209e-07))
  print("significant probes (adj.p-value < 0.05):")
  print(table(topHits[, 'adj.P.Val']<0.05))
  print("significant probes (adj.p-value <= 0.10):")
  print(table(topHits[, 'adj.P.Val']<=0.10))
  print("significant probes (adj.p-value <= 0.20):")
}

```

```

print(table(topHits[, 'adj.P.Val']<=0.20))
print("significant probes (p-value < 0.05):")
print(table(topHits[, 'P.Value']<0.05))

tophits.fdr <- topHits[topHits$adj.P.Val<0.05,]

write.table(topHits,file=file.path(results.dir,
  paste0("CAMP_DMPs_topHits_all_IgE_",gsub('\\.','.varNames[f]),_
  "_",timeStamp,".txt")),sep="\t",row.names=F,quote=F)

write.table(tophits.fdr,file=file.path(results.dir,
  paste0("CAMP_DMPs_topHits_fdr_IgE_",gsub('\\.','.varNames[f]),_
  "_",timeStamp,".txt")),sep="\t",row.names=F,quote=F)

# volcano plot
#volcanoplot(fit, coef=2, cex=1, main="volcano plot \n(blue: adj.P.Val<0.05)", pch=20
#  , highlight=sum(limma::topTable(fit, coef=2, adjust.method="BH")$adj.P.Val<0.05)
#  , names=rownames(topHits))

}

## [1] "pDat.tmp dims:"
## [1] 656 1098
## [1] "betas.tmp dims:"
## [1] 785352      656
## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+PC1+PC2"
## [1] ""
##           (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM   RACE2   RACE3   RACE4
## Down          98206             11398    9661   37715   36944    913     513
## NotSig        221137            770221   771832   732971  714944  783781  784457
## Up           466009             3733     3859   14666   33464    658     382
##           ETS1 MOMASM1   Bcell   CD4T   CD8T   Mono   Neu     NK    PC1    PC2
## Down          0       0  81323  70804  85911  78109  97605  70700  221942 295684
## NotSig  785352  785352 423903 447663 411543 481531 419349 479098  93205 192997
## Up            0       0 280126 266885 287898 225712 268398 235554 470205 296671
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 780314 5038
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE  TRUE
## 770221 15131
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE  TRUE
## 767028 18324
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE  TRUE
## 761246 24106
## [1] "significant probes (p-value < 0.05):"
##
## FALSE  TRUE
## 719387 65965
## [1] "pDat.tmp dims:"
## [1] 656 1098
## [1] "betas.tmp dims:"
## [1] 785352      656

```

```

## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1"
## [1] ""
##      (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM RACE2 RACE3 RACE4
## Down      103555           11200    7905  39183  38097   887   464
## NotSig     225027           770013  773866  730115 714991 783963 784593
## Up       456770            4139    3581  16054  32264   502   295
##      ETS1 MOMASM1 Bcell CD4T CD8T Mono Neu NK Plate2 Plate3
## Down      0        2 78227  67745  80249  73436  91556  68044  26036  17019
## NotSig   785352  785350 424168 448913 414103 481671 423255 477198 743749 751345
## Up       0        0 282957 268694 291000 230245 270541 240110 15567 16988
##      Plate4 Plate5 Plate6 Plate7 Plate8 Plate9 Plate10 Plate11 Plate12
## Down     4429  18975  7616 15059  43439  9020  10642  7737  12030
## NotSig  777521 749531 771923 757688 717551 769365 770050 772259 766191
## Up      3402  16846  5813 12605  24362  6967  4660  5356  7131
##      Plate13 Plate14 Plate15 Plate16 Plate17 ArrayR02C01 ArrayR03C01
## Down    15264   6286   8551 14968 141601          1230        4265
## NotSig  754534 766013 767662 750151 575576          782543        776841
## Up      15554  13053   9139 20233  68175          1579        4246
##      ArrayR04C01 ArrayR05C01 ArrayR06C01 ArrayR07C01 ArrayR08C01 PC1
## Down     21718    13364    3162    1441    19702 219929
## NotSig   744458    761336    779175    781503 750788 106634
## Up      19176    10652    3015    2408    14862 458789
##      PC2
## Down    279034
## NotSig  220478
## Up      285840
## [1] "significant probes (bonferroni threshold):"
##
## FALSE TRUE
## 780327 5025
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE TRUE
## 770013 15339
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 766429 18923
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 759705 25647
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 714827 70525
## [1] "pDat.tmp dims:"
## [1] 656 1098
## [1] "betas.tmp dims:"
## [1] 785352 656
## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+PC1+PC2"
## [1] ""
##      (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM RACE2 RACE3 RACE4
## Down      1           9165  48465  31049  40346   971   521
## NotSig    1861         773259  724908  744397 683084 783451 784449
## Up       783490         2928  11979   9906  61922   930   382

```

```

##          ETS1 MOMASM1      PC1      PC2
## Down      0      0 229228 334977
## NotSig  785352 785352 87932 173692
## Up       0      0 468192 276683
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 781664 3688
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE  TRUE
## 773259 12093
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE  TRUE
## 770526 14826
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE  TRUE
## 765785 19567
## [1] "significant probes (p-value < 0.05):"
##
## FALSE  TRUE
## 725541 59811
## [1] "pDat.tmp dims:"
## [1] 656 1098
## [1] "betas.tmp dims:"
## [1] 785352 656
## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+PC1+PC2+PC3"
## [1] ""
##          (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM RACE2 RACE3 RACE4
## Down      102054           11710    9713   38814  40863    943    560
## NotSig    222840           769961   772298  731248  715959  783783  784433
## Up       460458           3681     3341   15290   28530    626    359
##          ETS1 MOMASM1 Bcell CD4T CD8T Mono Neu NK PC1 PC2
## Down      0      0 83051 81739 96189 63249 75930 78595 221516 308694
## NotSig  785352 785352 421515 440074 437960 491399 436518 477586 104556 198781
## Up       0      0 280786 263539 251203 230704 272904 229171 459280 277877
##          PC3
## Down    172154
## NotSig  517088
## Up     96110
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 780247 5105
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE  TRUE
## 769961 15391
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE  TRUE
## 766667 18685
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE  TRUE

```

```

## 760660 24692
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 718285 67067
## [1] "pDat.tmp dims:"
## [1] 656 1098
## [1] "betas.tmp dims:"
## [1] 785352 656
## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+SV1"
## [1] ""
## (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM RACE2 RACE3 RACE4
## Down       69175           11331    4743   28869   48474    806   373
## NotSig     210973          770617  776222  739976  707748  784015 784749
## Up        505204           3404    4387   16507   29130    531   230
## ETS1 MOMASM1 Bcell CD4T CD8T Mono Neu NK Plate2 Plate3
## Down       0      0 116423 116623 140364 62496 58060 107241 75077 44779
## NotSig 785352 785352 353905 409869 359560 432313 421344 427023 574380 656721
## Up        0      0 315024 258860 285428 290543 305948 251088 135895 83852
## Plate4 Plate5 Plate6 Plate7 Plate8 Plate9 Plate10 Plate11 Plate12
## Down     29629 68078 16269 40875 66499 18947 23586 19881 30618
## NotSig 721994 582787 754692 686130 621903 748100 745911 750156 729698
## Up       33729 134487 14391 58347 96950 18305 15855 15315 25036
## Plate13 Plate14 Plate15 Plate16 Plate17 ArrayR02C01 ArrayR03C01
## Down     41812 38712 33378 45234 109806         910      5488
## NotSig 686624 672957 705759 674324 613544         782320    774469
## Up       56916 73683 46215 65794 62002         2122      5395
## ArrayR04C01 ArrayR05C01 ArrayR06C01 ArrayR07C01 ArrayR08C01 SV1
## Down     51660      54486      1960        818      48723 209464
## NotSig 688730      701502      781271      782880 678493 211214
## Up       44962      29364      2121        1654      58136 364674
## SV2
## Down     216742
## NotSig 486943
## Up       81667
## [1] "significant probes (bonferroni threshold):"
##
## FALSE TRUE
## 780603 4749
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE TRUE
## 770617 14735
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 767258 18094
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 760812 24540
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 716820 68532
## [1] "pDat.tmp dims:"
## [1] 656 1098

```

```

## [1] "betas.tmp dims:"
## [1] 785352    656
## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+SV1+SV2"
## [1] ""
##          (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM RACE2 RACE3 RACE4
## Down           2             9697   5523  43324 35357 1117   433
## NotSig        2142         772553 776506 719941 711655 783097 784563
## Up            783208         3102   3323  22087 38340 1138   356
##          ETS1 MOMASM1 SV1 SV2
## Down           0             0 226459 183454
## NotSig 785352 785352 113432 524847
## Up            0             0 445461 77051
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 781267 4085
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE  TRUE
## 772553 12799
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE  TRUE
## 769921 15431
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE  TRUE
## 765468 19884
## [1] "significant probes (p-value < 0.05):"
##
## FALSE  TRUE
## 727625 57727
## [1] "pDat.tmp dims:"
## [1] 656 1098
## [1] "betas.tmp dims:"
## [1] 785352    656
## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2+SV3"
## [1] ""
##          (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM RACE2 RACE3 RACE4
## Down           60539          11612   4404  25107 46230 1328   376
## NotSig        245166         770277 776339 749044 709723 783111 784725
## Up            479647          3463    4609  11201 29399  913   251
##          ETS1 MOMASM1 Bcell CD4T CD8T Mono Neu NK SV1 SV2
## Down           0             0 125876 156693 168504 54770 64424 137833 216720 257938
## NotSig 785352 785352 521284 482079 469196 621168 585975 513417 190766 457402
## Up            0             0 138192 146580 147652 109414 134953 134102 377866 70012
##          SV3
## Down      350055
## NotSig 234658
## Up       200639
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 780389 4963
## [1] "significant probes (adj.p-value < 0.05):"
##

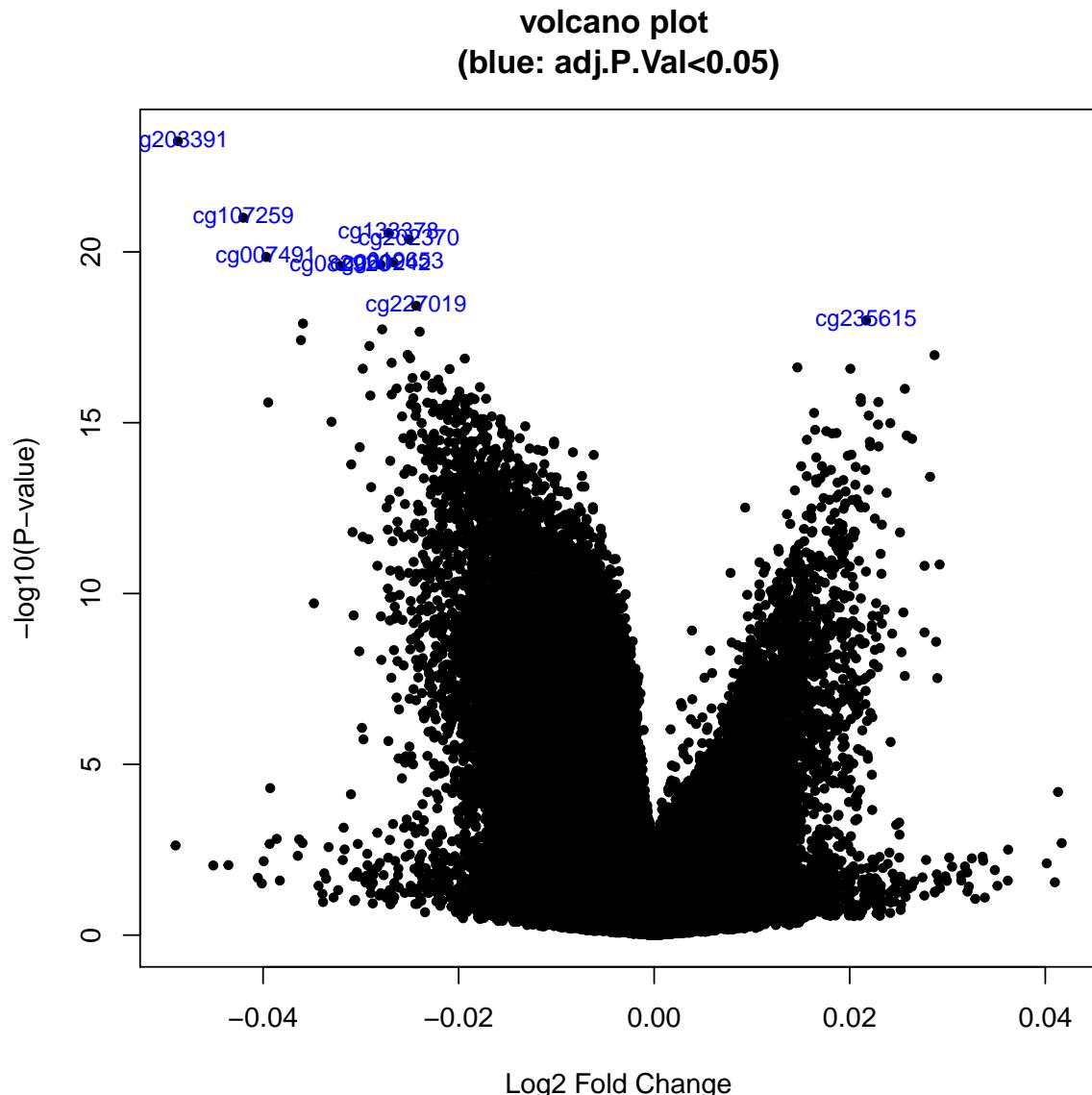
```

```

## FALSE TRUE
## 770277 15075
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 766912 18440
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 760818 24534
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 717906 67446
## [1] "pDat.tmp dims:"
## [1] 656 1098
## [1] "betas.tmp dims:"
## [1] 785352 656
## [1] ""
## [1] "~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2"
## [1] ""
## (Intercept) LOG10IGE_iuml_F48 age_f48 GenderM RACE2 RACE3 RACE4
## Down       69820           10958    4148  32207  45330    944    434
## NotSig     199629          771129  777051  732641  711754  783709  784634
## Up        515903           3265    4153   20504   28268    699    284
## ETS1 MOMASM1 Bcell CD4T CD8T Mono Neu NK SV1 SV2
## Down       0      0 119998 118025 146463  84509  70861 107128 215321 245999
## NotSig   785352 785352 351185 422615 357089 375887 384357 437259 195293 472772
## Up        0      0 314169 244712 281800 324956 330134 240965 374738 66581
## [1] "significant probes (bonferroni threshold):"
##
## FALSE TRUE
## 780667 4685
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE TRUE
## 771129 14223
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 768066 17286
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 762842 22510
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 722196 63156

# Plotting heavy volcano plots for all models may create issues in loading the pdf, therefore just p
volcanoplot(fit, coef=2, cex=1, main="volcano plot \n(blue: adj.P.Val<0.05)", pch=20
            , highlight=sum(limma::topTable(fit, coef=2, adjust.method="BH")$adj.P.Val<0.05)
            , names=rownames(topHits))

```



```
# defined models in a way that my last model will include cell types + 2SVs
# could also use: P_lambda(p) from package QCEWAS, gives same output
P <- topHits$P.Value
chisq <- qchisq(1-P,1)
lambda1 = median(chisq)/qchisq(0.5,1)
lambda1 # 1.125707 for the selected model with ct and 2 SVs
## [1] 1.125707

png(file = file.path(plots.dir, "qqplot_log10Ige_CAMP.png"),
    width = 540, height = 580)
qq(topHits$P.Value, main="q-q plot camp log10Ige")
dev.off()

## pdf
## 2
```

## 6 Applying Bacon to tstats

```
bc <- bacon(topHits$t)
estimates(bc)
```

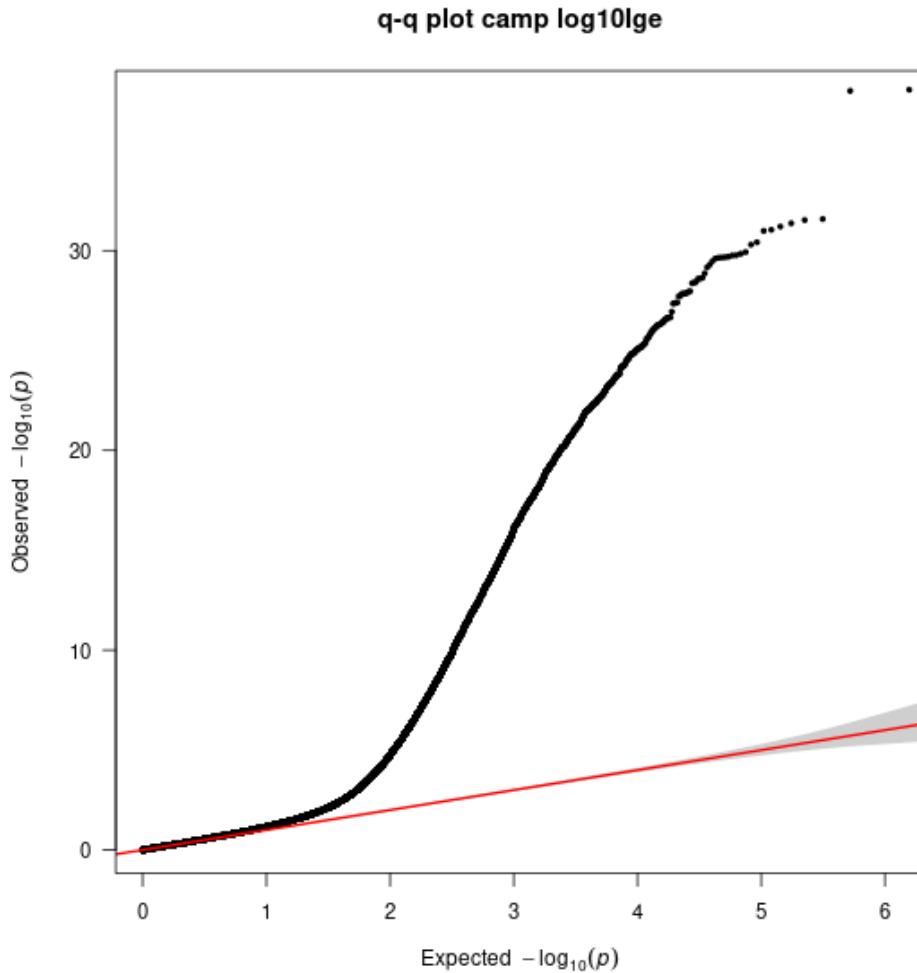


Figure 2: Q-Q CAMP

```

##          p.0      p.1      p.2      mu.0      mu.1      mu.2    sigma.0
## [1,] 0.9548062 0.003287347 0.04190648 -0.002648089 3.723778 -2.224594 1.016955
##      sigma.1 sigma.2
## [1,] 0.7591279 2.33275

str(bc)

## Formal class 'Bacon' [package "bacon"] with 9 slots
##   ..@ teststatistics: num [1:785352, 1] -13.9 -13.9 -12.5 -12.5 -12.5 ...
##   ..@ effectsizes : num [1, 1] 1
##   ..@ standarderrors: num [1, 1] 1
##   ..@ traces       : num [1:5000, 1:9, 1] 0.992 0.99 0.988 0.987 0.985 ...
##   ... .- attr(*, "dimnames")=List of 3
##   ... .$. : NULL
##   ... .$. : chr [1:9] "p.0" "p.1" "p.2" "mu.0" ...
##   ... .$. : NULL
##   ..@ estimates     : num [1, 1:9] 0.95481 0.00329 0.04191 -0.00265 3.72378 ...
##   ... .- attr(*, "dimnames")=List of 2
##   ... .$. : NULL
##   ... .$. : chr [1:9] "p.0" "p.1" "p.2" "mu.0" ...
##   ..@ priors       :List of 3
##   ... $. sigma  :List of 2

```

```

## ... .$. alpha: num 1.28
## ... .$. beta : num 0.36
## ... $. mu      :List of 2
## ... ... $. lambda: num [1:3] 0 3 -3
## ... ... $. tau   : num [1:3] 1000 100 100
## ... ... $. epsilon:List of 1
## ... ... $. gamma: num [1:3] 90 5 5
## ..@ niter       : int 5000
## ..@ nburnin     : int 2000
## ..@ na.exclude   : logi FALSE

bias(bc)

##          mu.0
## -0.002648089

inflation(bc) # using bacon, inflation is 1.017039

## sigma.0
## 1.016955

png(file = file.path(plots.dir, "qqplot_DM_CpGs_CAMP_bacon.png"),
     width = 1024, height = 540)
plot(bc, type="qq")
dev.off()

## pdf
## 2

png(file = file.path(plots.dir, "distribution_zscores_tstats_CAMP.png"),
     width = 540, height = 540)
plot(bc, type="hist")
dev.off()

## pdf
## 2

p <- data.frame(pval(bc))
topHits.bacon <- cbind(topHits, p)
P <- topHits.bacon$pval.bc.

# after bacon using usual method, the inflation: 1.08998
chisq <- qchisq(1-P,1)
lambda1 = median(chisq)/qchisq(0.5,1)
lambda1

## [1] 1.090151

topHits.bacon$fdr.bc. <- p.adjust(topHits.bacon$pval.bc., method="BH")
t <- data.frame(tstat(bc))
topHits.bacon <- cbind(topHits.bacon, t)

write.table(topHits.bacon,file=file.path(results.dir,
                                         paste0("CAMP_DMPs_topHits_all_log10Ige_ct2SVs_bacon_",timeStamp,".txt")),
            sep="\t",row.names=F,quote=F)

save(bc,file=file.path(results.dir,
                      paste0("CAMP_bacon_results_IgE_",
                             timeStamp,".RData")))

```

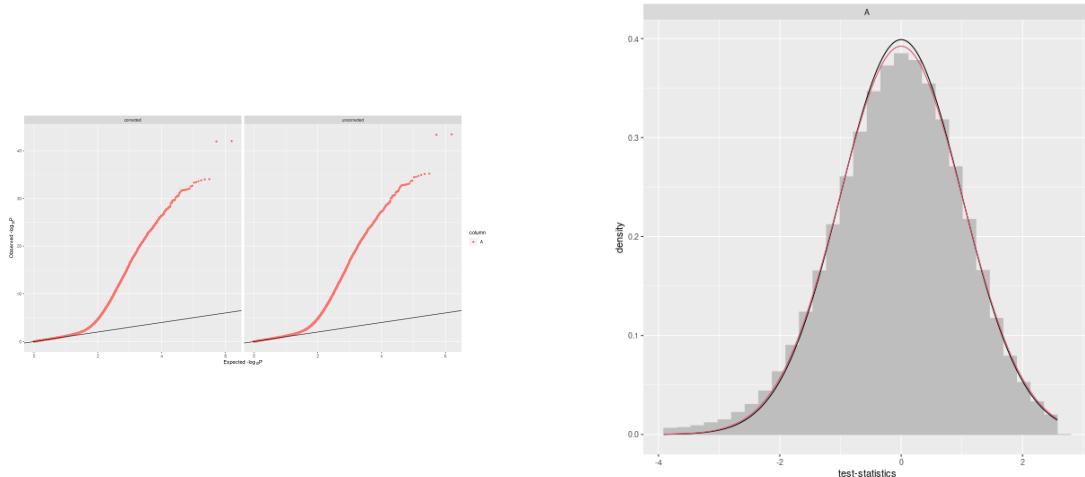


Figure 3: a) Q-Q CAMP bacon b) distribution zscores tstats CAMP

## 7 Regional analysis

```

formula <- c(~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+
            Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2)

vars.of.interest <- c("LOG10IGE_iuml_F48")
varNames <- c("LOG10IGE_iuml_F48")

# 1 missing for log10Ige remove before, others removed 8
remove <- c(list(which(is.na(pData.pheno.meth.camp$age_f48) |
                  is.na(pData.pheno.meth.camp$LOG10IGE_iuml_F48) |
                  is.na(pData.pheno.meth.camp$MOMASM) | is.na(pData.pheno.meth.camp$ETS) |
                  is.na(pData.pheno.meth.camp$Gender))))
length(remove[[1]]) # 18 subjects
## [1] 18

pDat.reg <- pData.pheno.meth.camp[!(rownames(pData.pheno.meth.camp) %in% remove[[1]]),]

ids.keep <- intersect(pDat.reg$toe_ids, colnames(beta.ewas))
betas.reg=beta.ewas[,which(colnames(beta.ewas) %in% ids.keep)]

myMs <- logit2(betas.reg)
myMs <- as.matrix(myMs)

pDat.reg$Gender <- relevel(pDat.reg$Gender, ref='F')
designCAMP = model.matrix(~LOG10IGE_iuml_F48+age_f48+Gender+RACE+ETS+MOMASM+
                           Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2,
                           data=pDat.reg)
myannotationCAMP <- cpq.annotate("array", myMs, what="M",
                                   annotation=c(array = "IlluminaHumanMethylationEPIC",
                                   annotation = "ilmn10b4.hg19"), arraytype = "EPIC",
                                   analysis.type="differential", design=designCAMP,
                                   coef="LOG10IGE_iuml_F48", fdr = 0.05)

# Your contrast returned 18251 individually significant probes.
# We recommend the default setting of pcutoff in dmrcate().

dmrcoutputCAMP <- dmrcate(myannotationCAMP, lambda=1000, C=2, pcutoff = "fdr")
results.rangesCAMP <- extractRanges(dmrcoutputCAMP, genome = "hg19")

```

```

DMRsCAMP <- data.frame(results.rangesCAMP)
DMRsCAMP <- DMRsCAMP[order(DMRsCAMP$Stouffer),]
dim(DMRsCAMP) # 3,251

## [1] 3251 13

DMRsCAMP.split <- sub(".*", "", DMRsCAMP$overlapping.genes)
DMRsCAMP.split.1 <- sub("-.*", "", DMRsCAMP.split)
DMRsCAMP$Gene1st <- DMRsCAMP.split.1

chr_st <- paste(DMRsCAMP$seqnames, DMRsCAMP$start, sep = "_")
chr_st_end <- paste(chr_st, DMRsCAMP$end, sep = "_")
DMRsCAMP$Coordinates <- chr_st_end

DMRsCAMP.sig <- DMRsCAMP[DMRsCAMP$Stouffer<0.05,]

# lowest and highest absolute mean methylation difference: 0.007
summary(DMRsCAMP.sig$meandiff)

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -0.033457 -0.010422 -0.006929 -0.005357 -0.003304  0.029140

# Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
##-0.034407 -0.009555 -0.006183 -0.004803 -0.002677  0.035584
head(DMRsCAMP.sig, n=10)

##   seqnames      start        end width strand no.cpgs min_smoothed_fdr
## 3   chr17 56269170 56270828 1659      *     8 1.652507e-106
## 6   chr14 100610186 100610667 482      *     4 4.904689e-94
## 4   chr20 35503983 35504553 571      *     8 1.242339e-112
## 2   chr1  27240319 27241913 1595      *    11 1.485488e-120
## 14  chr20 24630818 24630982 165      *     3 8.321306e-60
## 7   chr3  3151679  3152916 1238      *     9 1.313876e-102
## 17  chr7  149543136 149543165 30      *     2 3.508565e-63
## 1   chr1  6339954  6342888 2935      *    11 8.954410e-113
## 9   chr5  132008525 132010740 2216      *     9 3.425147e-75
## 31  chr19 5287627  5288631 1005      *     3 5.812943e-53
##           Stouffer       HMFDR       Fisher     maxdiff     meandiff
## 3  9.225520e-81 5.633057e-23 8.460802e-89 -0.03058577 -0.01959608
## 6  4.265284e-76 4.961292e-24 2.704025e-74 -0.04267346 -0.02826872
## 4  4.776919e-76 2.856969e-23 5.803055e-87 -0.02972640 -0.01811723
## 2  1.328849e-57 3.219847e-25 7.053971e-94 -0.02717409 -0.01371400
## 14 1.304158e-50 1.178976e-20 2.290969e-49 -0.02651755 -0.02034006
## 7  6.275455e-50 7.464102e-20 2.901340e-70 -0.02799791 -0.01828434
## 17 4.474517e-49 1.190221e-26 2.521616e-48 -0.05242022 -0.03333019
## 1  5.068944e-49 1.567635e-29 1.156110e-97 -0.05925855 -0.01777123
## 9  1.247820e-47 1.036537e-19 2.588361e-60 -0.02310981 -0.01286142
## 31 4.715780e-44 9.005802e-20 6.062049e-44 -0.02600662 -0.02045843
##   overlapping.genes Gene1st          Coordinates
## 3                  EPX      EPX chr17_56269170_56270828
## 6                  EVL      EVL chr14_100610186_100610667
## 4                  TLDC2    TLDC2 chr20_35503983_35504553
## 2      NUDC, NROB2    NUDC    chr1_27240319_27241913
## 14     SYNDIG1 SYNDIG1    chr20_24630818_24630982
## 7       IL5RA    IL5RA    chr3_3151679_3152916
## 17     ZNF862    ZNF862    chr7_149543136_149543165
## 1       ACOT7    ACOT7    chr1_6339954_6342888
## 9       IL4      IL4    chr5_132008525_132010740
## 31     PTPRS    PTPRS    chr19_5287627_5288631

```

```

dim(DMRsCAMP.sig) # 2,408

## [1] 2408   15

DMRsCAMP.sig$abs_meandiffCAMP <- abs(DMRsCAMP.sig$meandiff)

save(myannotationCAMP, dmrcoutputCAMP,
  file=file.path(results.dir,paste0("CAMPRegional_DMRCs_hg19_results_logIgE_",
  timeStamp, ".RData")))

write.table(DMRsCAMP.sig, file.path(results.dir,
  paste0("CAMP_DMRCs_dmrcate_logIgE_hg19_fdr_", timeStamp, ".txt")),
  sep="\t", quote=F, row.names=F)

```

## 8 Session information

[1] "2021-09-01" [1] "2021-09-01 22:52:27 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, bacon 1.18.0, 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, corrplot 0.84, data.table 1.14.0, dbplyr 2.1.0, DMRcate 2.4.1, DMRcatedata 2.8.2, DNAcopy 1.64.0, doParallel 1.0.16, dplyr 1.0.7, e1071 1.7-6, ellipse 0.4.2, ENmix 1.26.10, ExperimentHub 1.16.1, fastICA 1.2-2, FDB.InfiniumMethylation.hg19 2.2.0,forcats 0.5.1,foreach 1.5.1,gdata 2.18.0,gdsfmt 1.26.1,genefilter 1.72.1,genepotter 1.68.0,GenomeInfoDb 1.26.7,GenomicFeatures 1.42.3,GenomicRanges 1.42.0,GGally 2.1.0,ggplot2 3.3.5,ggpubr 0.4.0,ggrepel 0.9.1,gmodels 2.18.1,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,readxl 1.3.1,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 3.1.2,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): abind 1.4-5,affy 1.68.0,affyio 1.60.0,AnnotationFilter 1.14.0,askpass 1.1,assertthat 0.2.1,backports 1.2.1,base64 2.0,base64enc 0.1-3,beanplot 1.2,BiocManager 1.30.16,BiocVersion 3.12.0,biomaRt 2.46.3,biovizBase 1.38.0,bit 4.0.4,bit64 4.0.5,bitops 1.0-7,blob 1.2.1,boot 1.3-28,broom 0.7.6,BSgenome 1.58.0,

bsseq 1.26.0, cachem 1.0.4, car 3.0-10, carData 3.0-4, caTools 1.18.2, cellranger 1.1.0, checkmate 2.0.0, class 7.3-19, cli 3.0.1, clue 0.3-59, cluster 2.1.2, codetools 0.2-18, colorspace 2.0-2, 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, dichromat 2.0-0, digest 0.6.27, doRNG 1.8.2, DSS 2.38.0, dynamicTreeCut 1.63-1, edgeR 3.32.1, ellipsis 0.3.2, ensemblDb 2.14.1, evaluate 0.14, fansi 0.5.0, farver 2.1.0, fastmap 1.1.0, foreign 0.8-81, Formula 1.2-4, fs 1.5.0, generics 0.1.0, GenomeInfoDbData 1.2.4, GenomicAlignments 1.26.0, GEOquery 2.58.0, ggsignif 0.6.0, glue 1.4.2, grid 4.0.3, gtable 0.3.0, gtools 3.8.2, Gviz 1.34.1, haven 2.4.1, HDF5Array 1.18.1, highr 0.9, Hmisc 4.5-0, hms 1.0.0, htmlTable 2.1.0, htmltools 0.5.1.1, htmlwidgets 1.5.3, httpuv 1.6.0, httr 1.4.2, interactiveDisplayBase 1.28.0, irr 0.84.1, jpeg 0.1-8.1, jsonlite 1.7.2, KernSmooth 2.23-20, labeling 0.4.2, later 1.2.0, latticeExtra 0.6-29, lazyeval 0.2.2, lifecycle 1.0.0, 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, missMethyl 1.24.0, modelr 0.1.8, multtest 2.46.0, munsell 0.5.0, nleqslv 3.3.2, nloptr 1.2.2.2, mnet 7.3-16, nor1mix 1.3-0, openssl 1.4.4, openxlsx 4.2.3, permute 0.9-5, pillar 1.6.2, pkgconfig 2.0.3, png 0.1-7, prettyunits 1.1.1, progress 1.2.2, promises 1.2.0.1, ProtGenerics 1.22.0, proxy 0.4-25, R6 2.5.0, ragg 1.1.2, rappdirs 0.3.3, Rcpp 1.0.6, RCurl 1.98-1.3, reprex 2.0.0, reshape 0.8.8, rhdf5 2.34.0, rhdf5filters 1.2.1, Rhdf5lib 1.12.1, rio 0.5.26, rlang 0.4.11, rngtools 1.5, rpart 4.1-15, RPMM 1.25, rprojroot 2.0.2, Rsamtools 2.6.0, RSQLite 2.2.3, rstatix 0.6.0, 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, systemfonts 1.0.1, textshaping 0.3.3, tidyselect 1.1.1, tools 4.0.3, utf8 1.2.2, VariantAnnotation 1.36.0, vctrs 0.3.8, viridisLite 0.4.0, withr 2.4.2, xfun 0.22, xml2 1.3.2, xtable 1.8-4, yaml 2.2.1, zip 2.1.1, zlibbioc 1.36.0, zoo 1.8-9