

# CRA 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", "tidyR", "dplyr", "gridExtra", "isva", "sva", "base", "DMRcate", "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] "ggnpubr loaded successfully"
## [1] "gdata loaded successfully"
## [1] "gmodels loaded successfully"
## [1] "tidyR 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] "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

```

cra.dir="/proj/regeps/regep00/studies/CRA"
results.dir = file.path(cra.dir, "analysis/reprk/methylation/results/IgE_paper")
plots.dir = file.path(results.dir, "plots")
load(file=file.path(results.dir,
  "../norm.betas.cra_hg19_clean_N0sexchr_probands_1620830664.RData"))
load(file=file.path(results.dir, "../mset.cra.funnorm_hg19_1620501723.RData"))
pData.cra <- pData(mset.cra.funnorm)
pData.cra$toe_ids <- rownames(pData.cra)
ann850k <- getAnnotation(mset.cra.funnorm)

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

load(file=file.path(results.dir,
  "../CRA_EPIC_estimatecellcounts2_result_hg19_1620501723.RData"))

```

```

cra.pheno <- read.csv(file=file.path(cra.dir,
                                      "data/phenotype/CRA_Phenotype_Data/COS_TRIO_pheno_1165.csv"),
                      as.is=TRUE, sep=",", stringsAsFactors=FALSE)

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

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

```

## 2 Data wrangling and prep for SVA

```

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

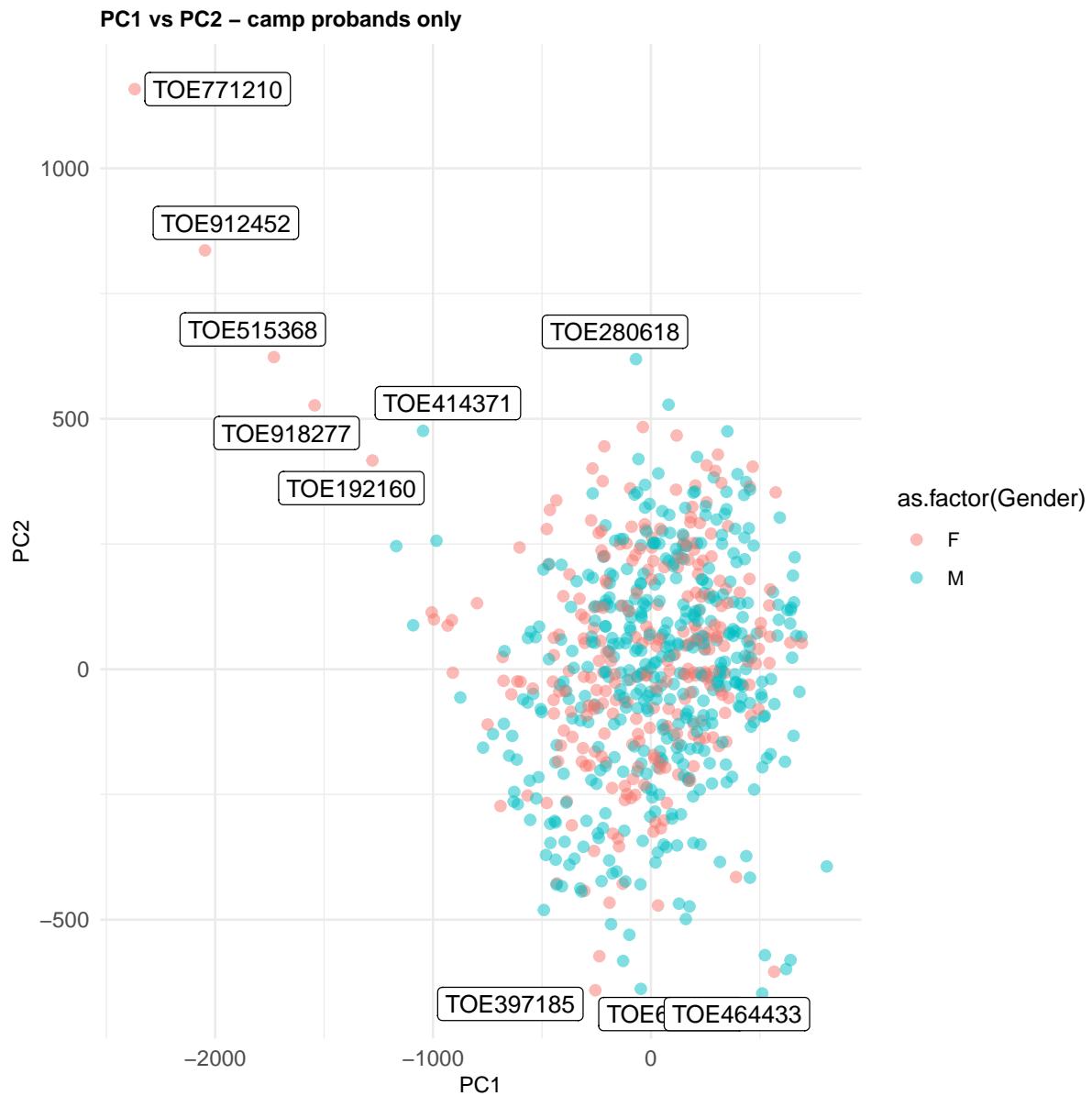
# merge all information necessary for models to phenotype file for CRA
samplesheet.cra=samplesheet.cra[,c("TOEID","Slide","Sample_Plate","Array"), drop=FALSE]
#setnames(pData.cra, "LEVEL1.TOEID", "TOEID")
pData.cra <- merge(pData.cra, samplesheet.cra, by.x="LEVEL1.TOEID", by.y="TOEID", sort=F)
pData.cra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

cra.pheno.ast <- cra.pheno[cra.pheno$Dr_Dx_Asthma==2,]
pData.pheno.cra <- merge(pData.cra, cra.pheno.ast, by="S SUBJECTID", sort=F)
dim(pData.pheno.cra) # 788 samples

## [1] 788 543

# Check how probands cluster based on PCs
# final sample set clustering
ggplot(pData.pheno.cra, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = as.factor(Gender)), alpha = 0.5, size = 2) +
  labs(title = "PC1 vs PC2 - CRA probands only") +
  geom_label_repel(aes(label = LEVEL1.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))

```



```

g1 <- ggplot(pData.pheno.cra, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = as.factor(Gender)), alpha = 0.5, size = 2) +
  labs(title = "PC1 vs PC2 - CRA probands only") +
  geom_label_repel(aes(label = LEVEL1.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))

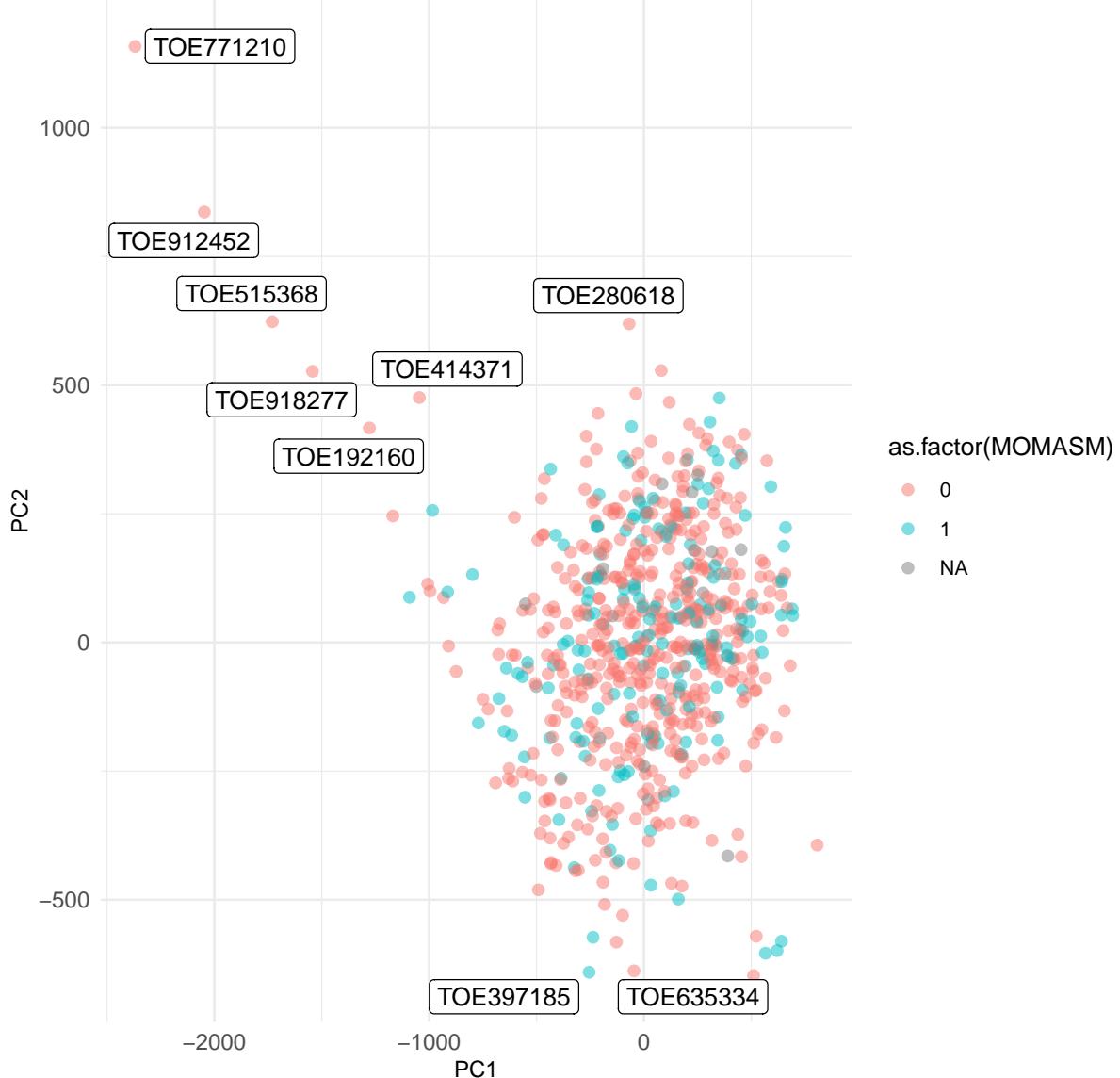
g2 <- ggplot(pData.pheno.cra, aes(x = PC2, y = PC3)) +
  geom_point(aes(color = as.factor(Gender)), alpha = 0.5, size = 2) +
  labs(title = "PC2 vs PC3 - CRA probands only") +
  geom_label_repel(aes(label = LEVEL1.TOEID),
    box.padding = 0.25,
    point.padding = 0.5,
    segment.color = 'grey50') +
  theme_minimal()

```

```

theme(plot.title = element_text(size = 10, face = "bold"),
      axis.text = element_text(size = 10),
      axis.title = element_text(size = 10))
grid.arrange(g1,g2, nrow = 1, ncol = 2)

```

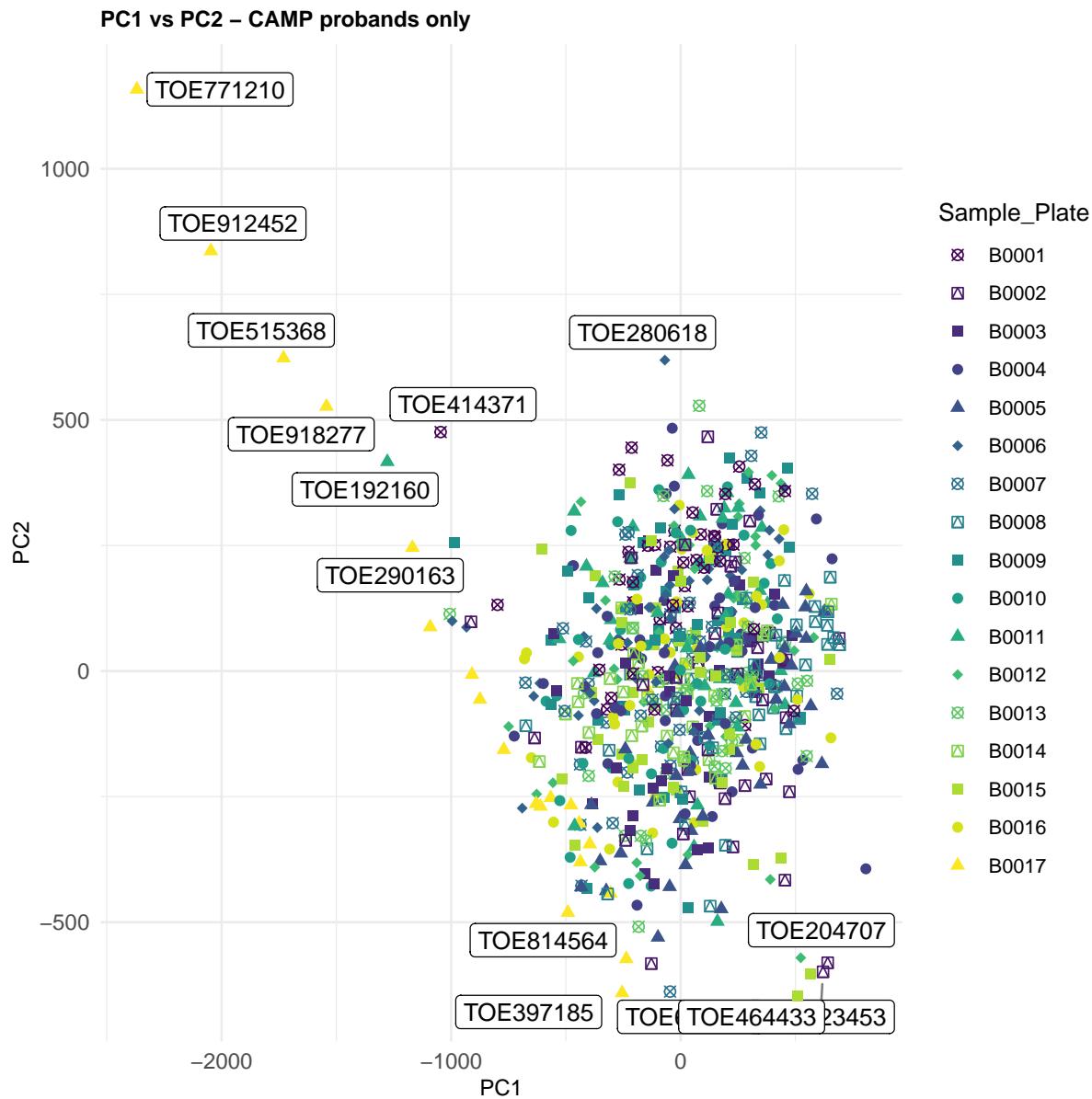
**PC1 vs PC2 – CAMP probands only**

```
ggsave(path=plots.dir, "pc1_pc2_pc3_probands_gender_CRA.png", width = 9, height = 6)
```

```

ggplot(pData.pheno.cra, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = as.factor(Mothers_Asthma_Hx)), alpha = 0.5, size = 2) +
  labs(title = "PC1 vs PC2 - CRA probands only") +
  geom_label_repel(aes(label = LEVEL1.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))

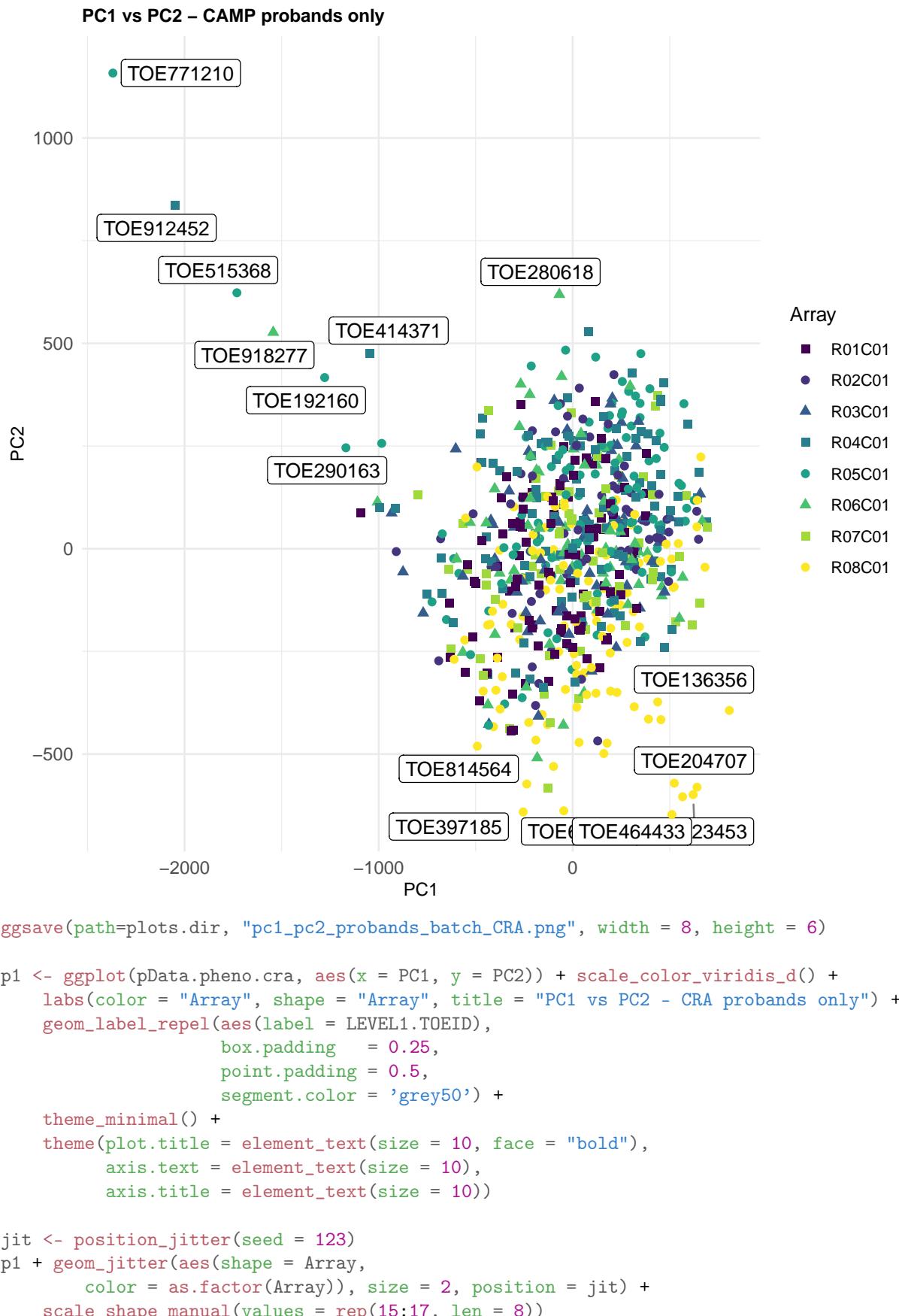
```

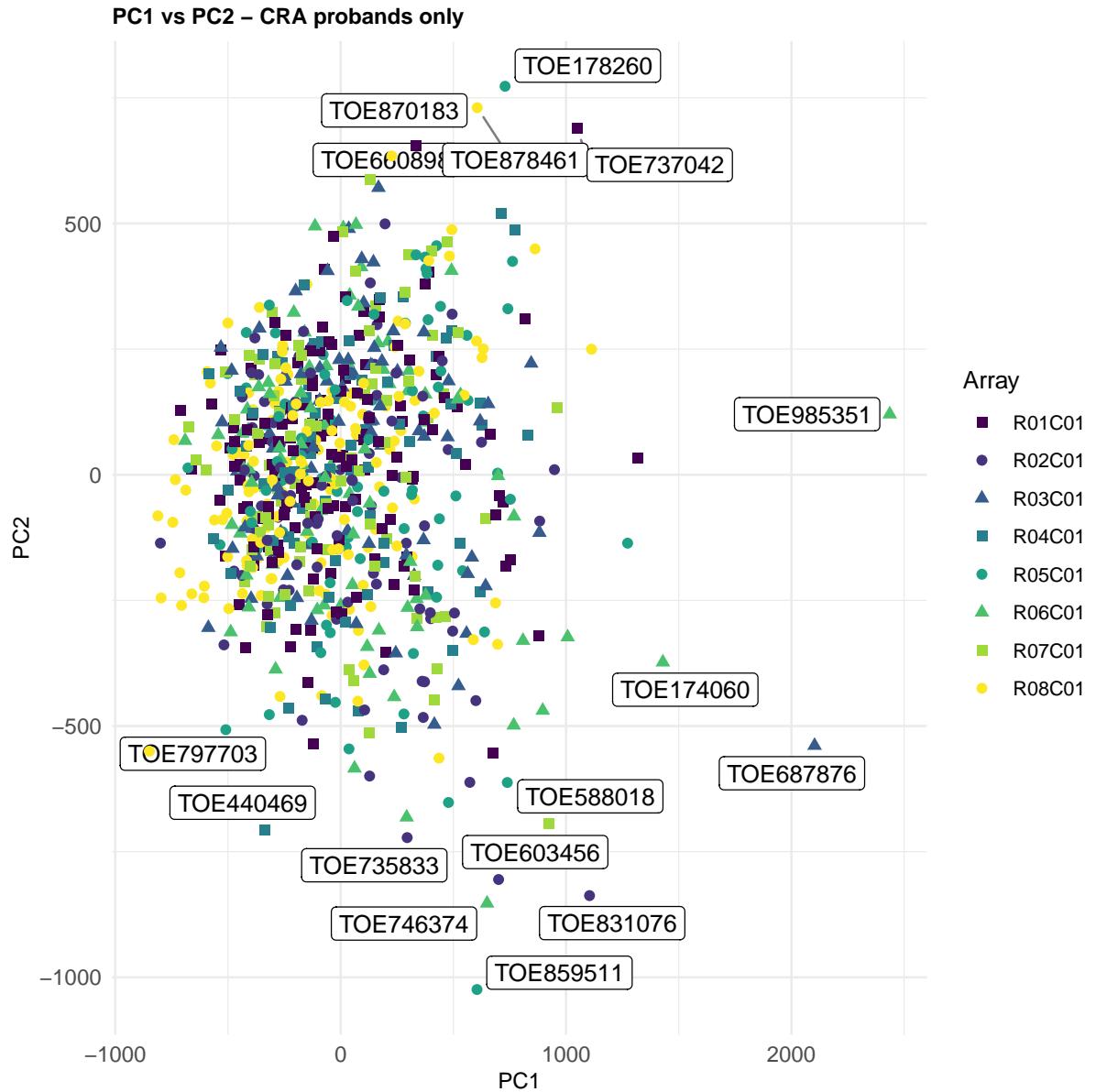


```
ggsave(path=plots.dir, "pc1_pc2_probands_masthma_CRA.png", width = 8, height = 6)
```

```
# Because sample plate has too many levels so done slightly differently
p1 <- ggplot(pData.pheno.cra, aes(x = PC1, y = PC2)) + scale_color_viridis_d() +
  labs(color = "Sample_Plate", shape = "Sample_Plate",
       title = "PC1 vs PC2 - CRA probands only") +
  geom_label_repel(aes(label = LEVEL1.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(15:17, len = 14))
```





```

pData.pheno.cra$age <- as.numeric(pData.pheno.cra$age)
pData.pheno.cra$Gender <- as.factor(pData.pheno.cra$Gender)
pData.pheno.cra$BMI <- as.numeric(pData.pheno.cra$BMI)
pData.pheno.cra$htcm <- as.numeric(pData.pheno.cra$htcm)
pData.pheno.cra$Mothers_Asthma_Hx <- as.factor(pData.pheno.cra$Mothers_Asthma_Hx)
pData.pheno.cra$Sample_Plate <- as.factor(pData.pheno.cra$Sample_Plate)
#samp.cra.pheno.ast$RC <- as.factor(samp.cra.pheno.ast$RC)
#samp.cra.pheno.ast$beadchip <- as.factor(samp.cra.pheno.ast$beadchip)
pData.pheno.cra$smoking_early_life <- as.factor(pData.pheno.cra$smoking_early_life)
pData.pheno.cra$log10Ige <- as.numeric(pData.pheno.cra$log10Ige)

which(is.na(pData.pheno.cra$log10Ige)) # row 273
## [1] 273

pData.pheno.cra[273,]$toe_ids
## [1] "TOE699277-BIS-v01_R03C01"
#[1] "TOE699277-BIS-v01_R03C01"

```

```

pData.pheno.cra[is.na(pData.pheno.cra$log10Ige),]$toe_ids
## [1] "TOE699277-BIS-v01_R03C01"

pData.pheno.cra[is.na(pData.pheno.cra$log10Ige),]$S_SUBJECTID
## [1] "ST-00068461"

which(is.na(pData.pheno.cra$BMI))
## [1] 506 533 608
# [1] 506 533 608
pData.pheno.cra[is.na(pData.pheno.cra$BMI),]$toe_ids
## [1] "TOE471221-BIS-v01_R05C01" "TOE531041-BIS-v01_R01C01"
## [3] "TOE858798-BIS-v01_R02C01"
# TOE471221-BIS-v01_R05C01, TOE531041-BIS-v01_R01C01, TOE858798-BIS-v01_R02C01

which(is.na(pData.pheno.cra$Mothers_Asthma_Hx))
## [1] 77 155 282
#[1] 77 155 282
pData.pheno.cra[is.na(pData.pheno.cra$Mothers_Asthma_Hx),]$toe_ids
## [1] "TOE263702-BIS-v01_R05C01" "TOE539838-BIS-v01_R01C01"
## [3] "TOE809590-BIS-v01_R06C01"
#TOE263702-BIS-v01_R05C01, TOE539838-BIS-v01_R01C01, TOE809590-BIS-v01_R06C01

which(is.na(pData.pheno.cra$smoking_early_life))
## [1] 197 338 368 591 693
# [1] 197 338 368 591 693
pData.pheno.cra[is.na(pData.pheno.cra$smoking_early_life),]$toe_ids
## [1] "TOE280386-BIS-v01_R02C01" "TOE227197-BIS-v01_R01C01"
## [3] "TOE974827-BIS-v01_R08C01" "TOE528343-BIS-v01_R07C01"
## [5] "TOE710489-BIS-v01_R04C01"
# TOE280386-BIS-v01_R02C01, TOE227197-BIS-v01_R01C01, TOE974827-BIS-v01_R08C01,
# TOE528343-BIS-v01_R07C01, TOE710489-BIS-v01_R04C01

pData.pheno.cra <- pData.pheno.cra[!is.na(pData.pheno.cra$log10Ige),]

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

ewas_var <- pData.pheno.cra$log10Ige

pData.pheno.cra$sex[pData.pheno.cra$Gender=="F"]<-0;
pData.pheno.cra$sex[pData.pheno.cra$Gender=="M"]<-1;
pData.pheno.cra$sex <- as.factor(pData.pheno.cra$sex)
pData.pheno.cra$Plate <- sapply(as.character(pData.pheno.cra$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,
                                 USE.NAMES = F)
pData.pheno.cra$Plate <- as.factor(pData.pheno.cra$Plate)
#pData.pheno.cra$Slide <- as.numeric(pData.pheno.cra$Slide)
pData.pheno.cra$array <- as.factor(pData.pheno.cra$array)

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

```

```

##      Plate      Array
## 5      : 65 R01C01 :149
## 13     : 65 R08C01 :147
## 9      : 64 R02C01 : 90
## 10     : 63 R06C01 : 85
## 3      : 62 R05C01 : 84
## 6      : 62 R07C01 : 82
## (Other):406 (Other):150

# 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)

## [1] 790798    787

beta.sva <- na.omit(beta.ewas)
dim(beta.sva)

## [1] 790798    787

#[1] 790798    787

# 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.cra[,c("log10Ige","Plate","Array"), drop=FALSE]
rownames(pheno.sel) <- pData.pheno.cra$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
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
n.sv # 62

## [1] 62

# Full model
mod <- model.matrix( ~ log10Ige + Plate + Array, data=pheno.sel)

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

```

```

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

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

##                                     SV1        SV2        SV3        SV4
## TOE909374-BIS-v01_R08C01 -0.06738999 -0.07542923 -0.003495351 -0.003147781
## TOE239310-BIS-v01_R01C01 -0.03494375  0.03626356 -0.034449350 -0.028382644
##                                     SV5        SV6        SV7        SV8
## TOE909374-BIS-v01_R08C01 -0.007813406 0.02516067 -0.057357175 -0.00136870
## TOE239310-BIS-v01_R01C01 -0.028081234 0.03196173 -0.009819548  0.05528291
##                                     SV9        SV10
## TOE909374-BIS-v01_R08C01  0.01165766 -0.04678761
## TOE239310-BIS-v01_R01C01  0.01677975 -0.03500805

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

save(beta.ewas, pData.pheno.cra,
  file=file.path(results.dir,paste0("CRA_betas_pheno_forIgE.EWAS_",
  timeStamp,".RData")))

beta.sva <- NULL

```

## 4 Correlations PCs and SVs and pc regression plots

```

cor.test(pData.pheno.cra$PC1, pData.pheno.cra$log10Ige) # trend

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$PC1 and pData.pheno.cra$log10Ige
## t = 1.9306, df = 785, p-value = 0.0539
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.001148268 0.137963692
## sample estimates:
##      cor
## 0.06874186

cor.test(pData.pheno.cra$PC2, pData.pheno.cra$log10Ige) # not sig

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$PC2 and pData.pheno.cra$log10Ige
## t = 0.20946, df = 785, p-value = 0.8341
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.06244156 0.07731989
## sample estimates:
##      cor
## 0.007475673

cor.test(pData.pheno.cra$PC3, pData.pheno.cra$log10Ige) # not sig

```

```

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.cra$PC3 and pData.pheno.cra$log10Ige
## t = -1.3045, df = 785, p-value = 0.1924
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.11601675 0.02345161
## sample estimates:
##          cor
## -0.04650922

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

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.cra$PC1 and as.numeric(pData.pheno.cra$Slide)
## t = -4.3887, df = 785, p-value = 1.296e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.22223348 -0.08579553
## sample estimates:
##          cor
## -0.1547523

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

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.cra$PC2 and as.numeric(pData.pheno.cra$Slide)
## t = -2.9932, df = 785, p-value = 0.002847
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.17481579 -0.03661619
## sample estimates:
##          cor
## -0.106229

cor.test(pData.pheno.cra$PC1, as.numeric(pData.pheno.cra$array))

## 
## Pearson's product-moment correlation
## 
## data: pData.pheno.cra$PC1 and as.numeric(pData.pheno.cra$array)
## t = -2.1345, df = 785, p-value = 0.03311
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.145076211 -0.006109569
## sample estimates:
##          cor
## -0.07596175

cor.test(pData.pheno.cra$PC2, as.numeric(pData.pheno.cra$array))

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

```

```

## t = -0.24506, df = 785, p-value = 0.8065
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.07858270  0.06117588
## sample estimates:
##      cor
## -0.008746121

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$PC1 and as.numeric(pData.pheno.cra$Plate)
## t = 0.3819, df = 785, p-value = 0.7026
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.05630883  0.08343455
## sample estimates:
##      cor
## 0.01362941

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$PC2 and as.numeric(pData.pheno.cra$Plate)
## t = -0.48635, df = 785, p-value = 0.6269
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.08713473  0.05259259
## sample estimates:
##      cor
## -0.01735581

# can go upto 2 in CRA
cor.test(pData.pheno.cra$SV1, pData.pheno.cra$log10Ige) # not sig

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV1 and pData.pheno.cra$log10Ige
## t = 0.91617, df = 785, p-value = 0.3599
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.03728766  0.10233300
## sample estimates:
##      cor
## 0.03268211

cor.test(pData.pheno.cra$SV2, pData.pheno.cra$log10Ige) # not sig

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV2 and pData.pheno.cra$log10Ige
## t = -1.6805, df = 785, p-value = 0.09326
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:

```

```
## -0.12921547 0.01005517
## sample estimates:
##       cor
## -0.05987151

cor.test(pData.pheno.cra$SV3, pData.pheno.cra$log10Ige) # significant

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV3 and pData.pheno.cra$log10Ige
## t = 2.179, df = 785, p-value = 0.02963
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.007693802 0.146626818
## sample estimates:
##       cor
## 0.07753672

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV1 and as.numeric(pData.pheno.cra$Slide)
## t = -0.97346, df = 785, p-value = 0.3306
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.10435481 0.03524671
## sample estimates:
##       cor
## -0.03472343

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV2 and as.numeric(pData.pheno.cra$Slide)
## t = 3.5447, df = 785, p-value = 0.0004164
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.05612191 0.19369982
## sample estimates:
##       cor
## 0.1255142

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV3 and as.numeric(pData.pheno.cra$Slide)
## t = -4.3952, df = 785, p-value = 1.259e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.22245064 -0.08602231
## sample estimates:
##       cor
## -0.1549753
```

```

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV1 and as.numeric(pData.pheno.cra$array)
## t = -1.5616, df = 785, p-value = 0.1188
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.12504850 0.01428997
## sample estimates:
## cor
## -0.05565022

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV2 and as.numeric(pData.pheno.cra$array)
## t = -1.2157, df = 785, p-value = 0.2244
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.1128936 0.0266142
## sample estimates:
## cor
## -0.04335104

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV3 and as.numeric(pData.pheno.cra$array)
## t = 0.26237, df = 785, p-value = 0.7931
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.06056043 0.07919659
## sample estimates:
## cor
## 0.009363807

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

##
## Pearson's product-moment correlation
##
## data: pData.pheno.cra$SV1 and as.numeric(pData.pheno.cra$Plate)
## t = 0.65412, df = 785, p-value = 0.5132
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.04662034 0.09307311
## sample estimates:
## cor
## 0.02334032

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

##
## Pearson's product-moment correlation

```

```

## 
## data: pData.pheno.cra$SV2 and as.numeric(pData.pheno.cra$Plate)
## t = -0.30234, df = 785, p-value = 0.7625
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.08061408 0.05913895
## sample estimates:
## cor
## -0.01079025

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

## 
## Pearson's product-moment correlation
## 

## data: pData.pheno.cra$SV3 and as.numeric(pData.pheno.cra$Plate)
## t = -1.5371, df = 785, p-value = 0.1247
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.12418681 0.01516506
## sample estimates:
## cor
## -0.0547776

cor.data=pData.pheno.cra[,c("log10Ige","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 Plate Slide Array   PC1   PC2   PC3   PC4   PC5   PC6   PC7
## log10Ige      1.00  0.03 -0.10 -0.03  0.07  0.01 -0.05 -0.04  0.07  0.01 -0.12
## Plate        0.03  1.00 -0.09  0.00  0.01 -0.02 -0.02  0.17  0.06 -0.23 -0.04
## Slide       -0.10 -0.09  1.00  0.01 -0.15 -0.11  0.17 -0.06 -0.03  0.06  0.00
## Array        -0.03  0.00  0.01  1.00 -0.08 -0.01 -0.05  0.07  0.13 -0.05 -0.06
## PC1         0.07  0.01 -0.15 -0.08  1.00  0.00  0.00  0.00  0.00  0.00  0.00
## PC2         0.01 -0.02 -0.11 -0.01  0.00  1.00  0.00  0.00  0.00  0.00  0.00
##          PC8   PC9  PC10   SV1   SV2   SV3   SV4   SV5   SV6   SV7   SV8
## log10Ige   -0.12 -0.01 -0.03  0.03 -0.06  0.08 -0.03 -0.02  0.04  0.29  0.14
## Plate       0.18 -0.21  0.00  0.02 -0.01 -0.05  0.01 -0.02 -0.09 -0.18  0.04
## Slide       -0.06  0.04 -0.03 -0.03  0.13 -0.15 -0.16  0.02  0.02  0.04 -0.07
## Array        0.00  0.02  0.03 -0.06 -0.04  0.01 -0.06 -0.10 -0.03  0.01  0.04
## PC1         0.00  0.00  0.00  0.82 -0.18  0.42  0.27  0.09  0.03 -0.01  0.04
## PC2         0.00  0.00  0.00 -0.40  0.40  0.71  0.32 -0.10 -0.03 -0.09  0.11
##          SV9   SV10
## log10Ige   0.04  0.05
## Plate      0.22  0.12
## Slide      0.02 -0.03
## Array     -0.04  0.00
## PC1        0.02  0.01
## PC2        0.06 -0.01

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

##          log10Ige      Plate       Slide       Array       PC1
## log10Ige 0.000000000 0.47501049 6.884339e-03 0.41560882 5.389661e-02

```

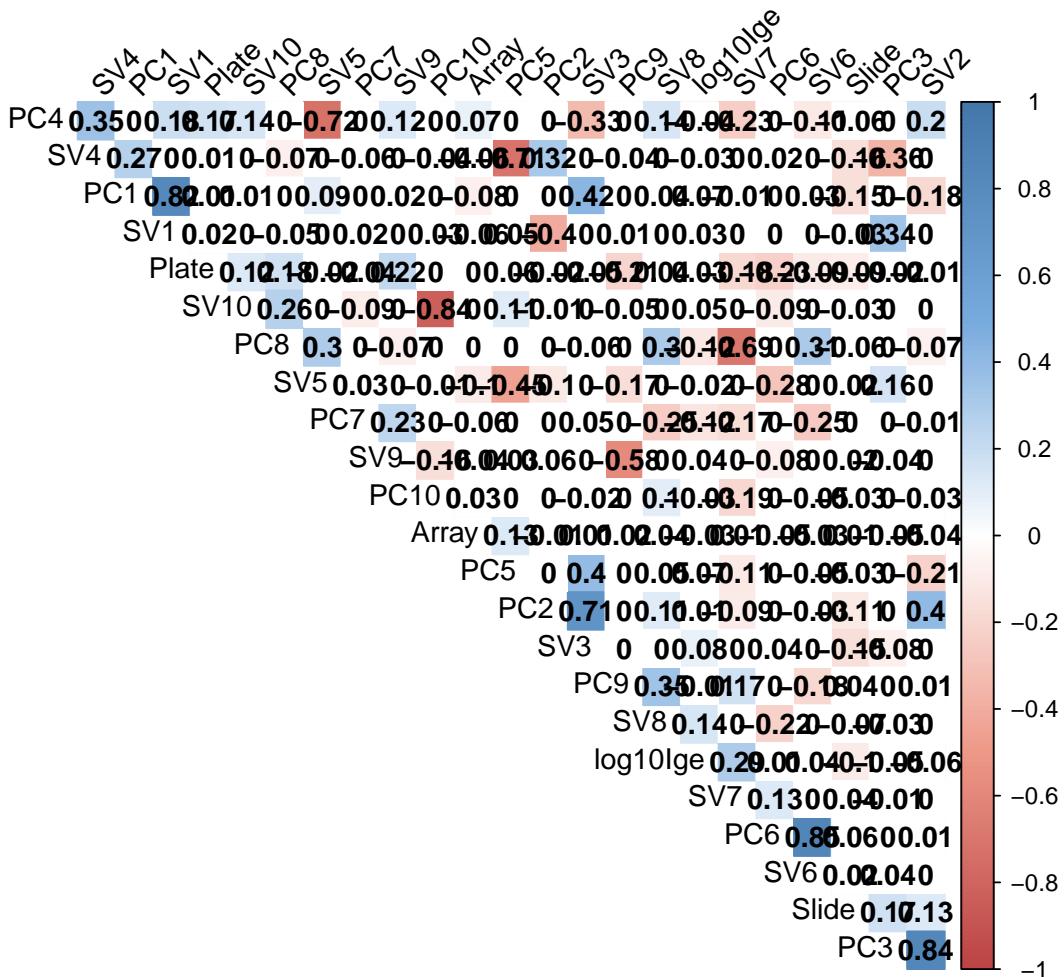
```

## Plate      0.475010492 0.00000000 1.434934e-02 0.98177979 7.026371e-01
## Slide      0.006884339 0.01434934 0.000000e+00 0.80976802 1.296225e-05
## Array      0.415608815 0.98177979 8.097680e-01 0.00000000 3.311499e-02
## PC1       0.053896613 0.70263714 1.296225e-05 0.03311499 0.000000e+00
## PC2       0.834144957 0.62685781 2.846869e-03 0.80647642 9.989602e-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_CRA_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+age+Gender+BMI+htcm+Mothers_Asthma_Hx+smoking_early_life+
fevbd0+fevbd1+fvcbd0+fvcbd1+fev1_fvc_bd0+fev1_fvc_bd1+f2575_fvc_bd0+
f2575_fvc_bd1+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Slide+Array+PC1+PC2+SV1+SV2)

remove <- c(list(which(is.na(pData.pheno.cra$age)) | is.na(pData.pheno.cra$fevbd0) |
is.na(pData.pheno.cra$fevbd1) | is.na(pData.pheno.cra$fvcbd0) |
is.na(pData.pheno.cra$fvcbd1) | is.na(pData.pheno.cra$fev1_fvc_bd0) |
is.na(pData.pheno.cra$fev1_fvc_bd1) | is.na(pData.pheno.cra$f2575_fvc_bd0) |
is.na(pData.pheno.cra$f2575_fvc_bd1) | is.na(pData.pheno.cra$log10Ige) |
is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_early_life) |
is.na(pData.pheno.cra$Gender) | is.na(pData.pheno.cra$BMI) |
is.na(pData.pheno.cra$htcm))))
length(remove[[1]]) # 30 samples

## [1] 30

pDat.tmp <- pData.pheno.cra[!(rownames(pData.pheno.cra) %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) # 790798    787

## [1] 790798    787

dim(betas.pcr) # 790798    757

## [1] 790798    757

pheno.sel=pDat.tmp[,c("age", "Gender", "BMI", "htcm", "Mothers_Asthma_Hx",
"smoking_early_life", "fevbd0", "fevbd1", "fvcbd0",
"fvcbd1", "fev1_fvc_bd0", "fev1_fvc_bd1", "f2575_fvc_bd0",
"f2575_fvc_bd1", "log10Ige", "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
## 21196

#Top 10 principal components can explain ~43 % 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 43.38627 % of data
##      variation
## pcr_diag.jpg was plotted

```

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

## [1] TRUE

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

## [1] TRUE

rm(betas.pcr)
```

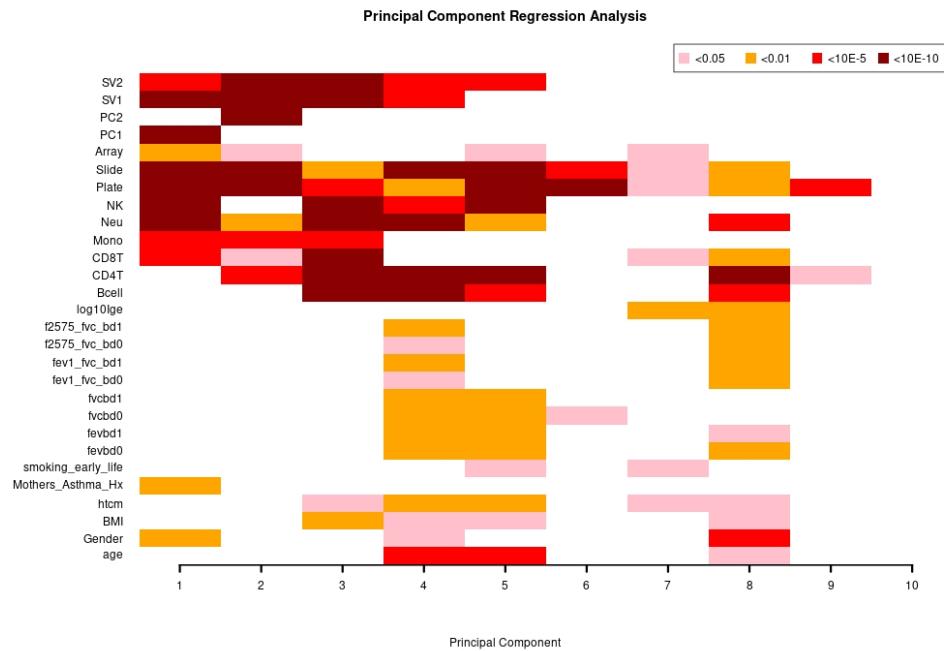


Figure 1: PC regression plot against all phenotypes CRA

## 5 IgE EWAS

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

betas.ann850k <- merge(beta.ewas, ann850k, by.x="row.names", by.y="Name", sort=F)
betas.ann850k[1:2,786:790]

##    TOE860944-BIS-v01_R01C01 TOE930479-BIS-v01_R02C01 TOE294448-BIS-v01_R06C01
## 1          0.934889         0.9641686        0.9271225
## 2          0.697392         0.8770760        0.6075057
##     chr   pos
## 1 chr1 10848
## 2 chr1 10850

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

## [1] 790798     834

betas.ann850k <- betas.ann850k[,789:834]
betas.ann850k$Gene <- sub(".*", "", betas.ann850k$UCSC_RefGene_Name)

# CRA
# model1: 22,097, 212 overlap with FHS
# model2: 22,931, 213 overlap with FHS
# model3: 18,507, 205 overlap with FHS
# model4: 23,060, 211 overlap with FHS
# model5: 21,801, 200 overlap with FHS
# model6: 21,791, 208 overlap with FHS
# model7: 27,651, 221 overlap with FHS
# model8: 19,117, 202 overlap with FHS

formula <- c(~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2+PC3+SV1+SV2+SV3,
  ~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2+PC3+SV1+SV2+SV3,
  ~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2+PC3+SV1+SV2+SV3,
  ~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2+PC3+SV1+SV2+SV3,
  ~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2+PC3+SV1+SV2+SV3,
  ~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2+PC3+SV1+SV2+SV3,
  ~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+Array+PC1+PC2+PC3+SV1+SV2+SV3)

vars.of.interest <- c("log10Ige", "log10Ige", "log10Ige", "log10Ige", "log10Ige", "log10Ige", "log10Ige",
  varNames <- c("log10Ige_ct_2PCs", "log10Ige_ct_plt_arr_2PCs", "log10Ige_noct_2PCs", "log10Ige_ct_3PCs")

# 1 missing for log10Ige remove before, others removed 8
remove <- c(list(which(is.na(pData.pheno.cra$age) | is.na(pData.pheno.cra$log10Ige) |
  is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_early_life) |
  is.na(pData.pheno.cra$Gender))), list(which(is.na(pData.pheno.cra$age) | is.na(pData.pheno.cra$log10Ige) |
  is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_early_life) |
  is.na(pData.pheno.cra$Gender))), list(which(is.na(pData.pheno.cra$age) | is.na(pData.pheno.cra$log10Ige) |
  is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_early_life) |
  is.na(pData.pheno.cra$Gender))),
```

```

list(which(is.na(pData.pheno.cra$age) | is.na(pData.pheno.cra$log10Ige) |
       is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_ean))
list(which(is.na(pData.pheno.cra$age) | is.na(pData.pheno.cra$log10Ige) |
       is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_ean)
       is.na(pData.pheno.cra$Gender))),
list(which(is.na(pData.pheno.cra$age) | is.na(pData.pheno.cra$log10Ige) |
       is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_ean)
       is.na(pData.pheno.cra$Gender)))
)

length(remove[[1]])

## [1] 8

sig.probes <- list()

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.cra[-remove[[f]],]
    betas.tmp <- beta.ewas[,-remove[[f]]]

  } else {

    pDat.tmp <- pData.pheno.cra
    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("")
  pDat.tmp$Gender <- relevel(pDat.tmp$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("CRA_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("CRA_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] 779 555
## [1] "betas.tmp dims:"
## [1] 790798    779
## [1] ""
## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+PC1+PC2"
## [1] ""
##          (Intercept) log10Ige      age GenderM Mothers_Asthma_Hx2
## Down        30415     14738   59486   41891                  0
## NotSig      214067    768701  705004   723322                 790798
## Up         546316     7359   26308   25585                  0
##          smoking_early_life2 Bcell   CD4T   CD8T   Mono   Neu   NK   PC1
## Down            0 43555  44065  40707  45321  54142 37540 456785
## NotSig        790798 547249 496610 534770 609594 538683 609313 926777
## Up             0 199994 250123 215321 135883 197973 143945 241336
##          PC2
## Down     351629
## NotSig   134410
## Up       304759
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 784129  6669
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE  TRUE
## 768701  22097
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE  TRUE
## 763706  27092
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE  TRUE
## 754325  36473
## [1] "significant probes (p-value < 0.05):"

```

```

##  

## FALSE TRUE  

## 711305 79493  

## [1] "pDat.tmp dims:"  

## [1] 779 555  

## [1] "betas.tmp dims:"  

## [1] 790798 779  

## [1] ""  

## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+  

## [1] ""  

## (Intercept) log10Ige age GenderM Mothers_Asthma_Hx2  

## Down 34422 15952 56871 42576 0  

## NotSig 229669 767867 708053 720628 790798  

## Up 526707 6979 25874 27594 0  

## smoking_early_life2 Bcell CD4T CD8T Mono Neu NK Plate2  

## Down 0 35220 35597 40657 36364 43806 34037 20  

## NotSig 790798 567322 521900 528514 631760 564473 603905 790658  

## Up 0 188256 233301 221627 122674 182519 152856 120  

## Plate3 Plate4 Plate5 Plate6 Plate7 Plate8 Plate9 Plate10 Plate11 Plate12  

## Down 70009 58619 28290 37368 52058 21643 43084 38719 57955 63712  

## NotSig 608621 597846 709004 689263 641291 746587 716029 670681 626168 688657  

## Up 112168 134333 53504 64167 97449 22568 31685 81398 106675 38429  

## Plate13 Plate14 ArrayR02C01 ArrayR03C01 ArrayR04C01 ArrayR05C01  

## Down 47928 1792 2835 13016 10318 41393  

## NotSig 714174 787743 784937 762295 762161 726290  

## Up 28696 1263 3026 15487 18319 23115  

## ArrayR06C01 ArrayR07C01 ArrayR08C01 PC1 PC2  

## Down 43260 9484 28023 450277 345052  

## NotSig 718189 770371 743148 125832 155302  

## Up 29349 10943 19627 214689 290444  

## [1] "significant probes (bonferroni threshold):"  

##  

## FALSE TRUE  

## 783959 6839  

## [1] "significant probes (adj.p-value < 0.05):"  

##  

## FALSE TRUE  

## 767867 22931  

## [1] "significant probes (adj.p-value <= 0.10):"  

##  

## FALSE TRUE  

## 762623 28175  

## [1] "significant probes (adj.p-value <= 0.20):"  

##  

## FALSE TRUE  

## 752467 38331  

## [1] "significant probes (p-value < 0.05):"  

##  

## FALSE TRUE  

## 708600 82198  

## [1] "pDat.tmp dims:"  

## [1] 779 555  

## [1] "betas.tmp dims:"  

## [1] 790798 779  

## [1] ""  

## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+PC1+PC2"  

## [1] ""  

## (Intercept) log10Ige age GenderM Mothers_Asthma_Hx2

```

```

## Down 0 16817 65174 37247 0
## NotSig 356 772291 705985 738616 790798
## Up 790442 1690 19639 14935 0
## smoking_early_life2 PC1 PC2
## Down 0 469989 333197
## NotSig 790798 75084 141700
## Up 0 245725 315901
## [1] "significant probes (bonferroni threshold):"
##
## FALSE TRUE
## 785230 5568
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE TRUE
## 772291 18507
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 766993 23805
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 756484 34314
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 707666 83132
## [1] "pDat.tmp dims:"
## [1] 779 555
## [1] "betas.tmp dims:"
## [1] 790798 779
## [1] ""
## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+PC1+PC2"
## [1] ""
## (Intercept) log10Ige age GenderM Mothers_Asthma_Hx2
## Down 33179 15067 25099 37221 0
## NotSig 215156 767738 754918 731361 790798
## Up 542463 7993 10781 22216 0
## smoking_early_life2 Bcell CD4T CD8T Mono Neu NK PC1
## Down 0 44794 44731 36954 63383 76659 31842 479360
## NotSig 790798 540513 491448 546227 564866 497475 627483 87307
## Up 0 205491 254619 207617 162549 216664 131473 224131
## PC2 PC3
## Down 336121 233240
## NotSig 156584 311819
## Up 298093 245739
## [1] "significant probes (bonferroni threshold):"
##
## FALSE TRUE
## 784075 6723
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE TRUE
## 767738 23060
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 761792 29006

```

```

## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 750458 40340
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 705325 85473
## [1] "pDat.tmp dims:"
## [1] 779 555
## [1] "betas.tmp dims:"
## [1] 790798 779
## [1] ""
## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+Plate+A"
## [1] ""
## (Intercept) log10Ige age GenderM Mothers_Asthma_Hx2
## Down 24025 17786 52023 39831 0
## NotSig 177580 769007 713347 720770 790798
## Up 589193 4005 25428 30197 0
## smoking_early_life2 Bcell CD4T CD8T Mono Neu NK Plate2
## Down 0 81064 86404 85890 53692 61412 72465 6
## NotSig 790798 407031 438076 490156 385599 339322 516770 790780
## Up 0 302703 266318 214752 351507 390064 201563 12
## Plate3 Plate4 Plate5 Plate6 Plate7 Plate8 Plate9 Plate10 Plate11 Plate12
## Down 210741 288361 20981 45363 92604 17385 74792 160243 220866 154051
## NotSig 490760 372671 746234 707311 628548 757156 690486 555097 470060 589893
## Up 89297 129766 23583 38124 69646 16257 25520 75458 99872 46854
## Plate13 Plate14 ArrayR02C01 ArrayR03C01 ArrayR04C01 ArrayR05C01
## Down 47529 14850 1611 7311 4899 17008
## NotSig 723642 771530 779397 767384 771669 750859
## Up 19627 4418 9790 16103 14230 22931
## ArrayR06C01 ArrayR07C01 ArrayR08C01 SV1 SV2
## Down 14515 3392 25084 261793 48882
## NotSig 746925 776153 728131 292335 503420
## Up 29358 11253 37583 236670 238496
## [1] "significant probes (bonferroni threshold):"
##
## FALSE TRUE
## 785130 5668
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE TRUE
## 769007 21791
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 763433 27365
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 752599 38199
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 707667 83131
## [1] "pDat.tmp dims:"
## [1] 779 555
## [1] "betas.tmp dims:"

```

```

## [1] 790798    779
## [1] ""
## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+SV1+SV2"
## [1] ""
##          (Intercept) log10Ige      age GenderM Mothers_Asthma_Hx2
## Down            0   22021   63154   37105             0
## NotSig         453   763147  704620  729920           790798
## Up            790345   5630   23024   23773             0
##          smoking_early_life2     SV1     SV2
## Down            0  446997  171644
## NotSig         790798   92394  216054
## Up            0  251407  403100
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 783025  7773
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE  TRUE
## 763147  27651
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE  TRUE
## 755204  35594
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE  TRUE
## 738490  52308
## [1] "significant probes (p-value < 0.05):"
##
## FALSE  TRUE
## 688795 102003
## [1] "pDat.tmp dims:"
## [1] 779 555
## [1] "betas.tmp dims:"
## [1] 790798    779
## [1] ""
## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2"
## [1] ""
##          (Intercept) log10Ige      age GenderM Mothers_Asthma_Hx2
## Down        21942   14061   35635   42130             0
## NotSig      184589   771681  710480  713215           790798
## Up          584267   5056   44683   35453             0
##          smoking_early_life2 Bcell   CD4T   CD8T   Mono   Neu   NK   SV1
## Down            0  59378  81831  107045  61367  82888  87043 366727
## NotSig         790798  576949  530158  464593  577035  521917  515268 177506
## Up            0 154471  178809  219160  152396  185993  188487 246565
##          SV2     SV3
## Down       203873 448202
## NotSig     334680 153729
## Up        252245 188867
## [1] "significant probes (bonferroni threshold):"
##
## FALSE  TRUE
## 786340  4458
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE  TRUE

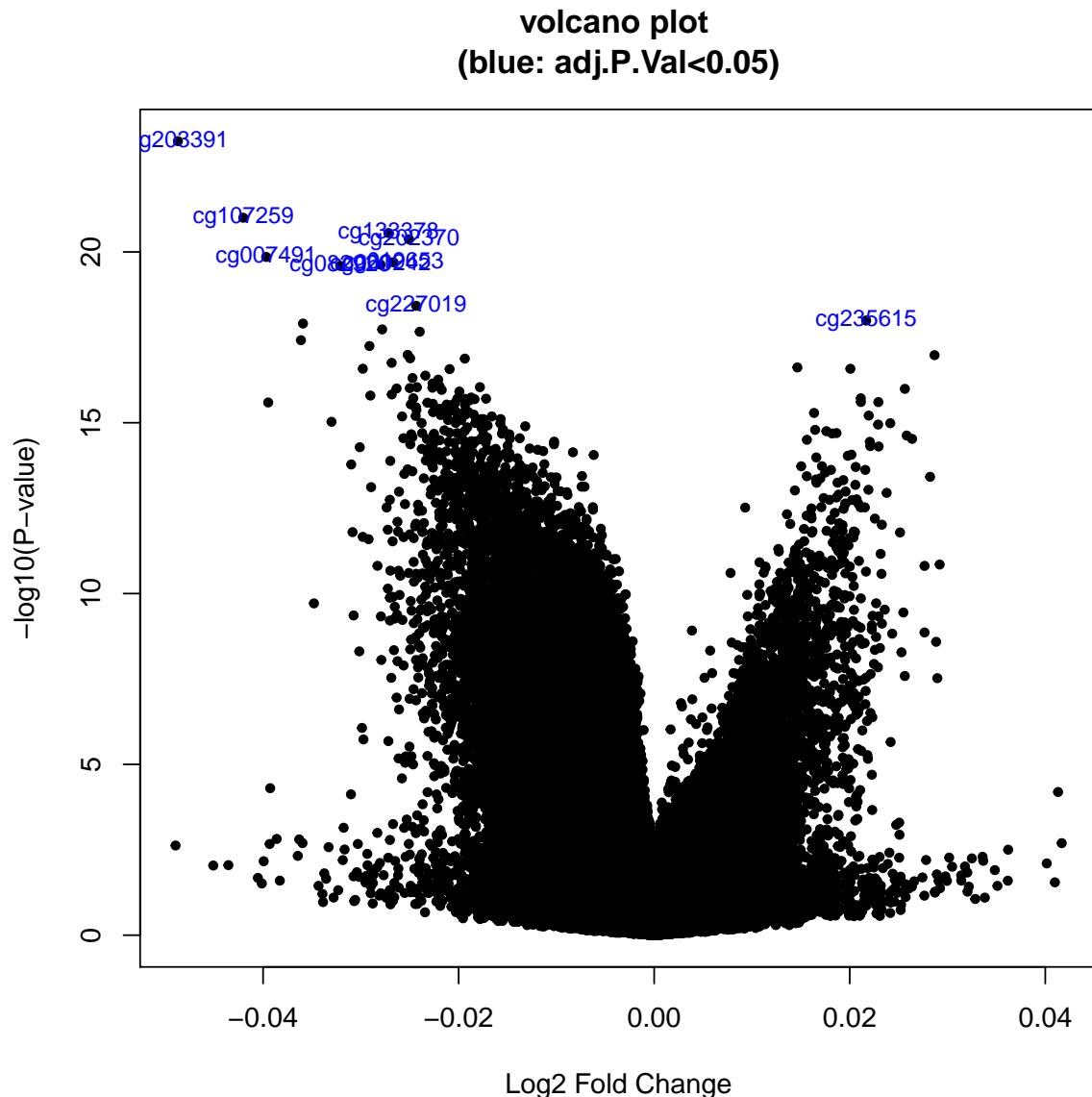
```

```

## 771681 19117
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 765115 25683
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 751192 39606
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 696841 93957
## [1] "pDat.tmp dims:"
## [1] 779 555
## [1] "betas.tmp dims:"
## [1] 790798 779
## [1] ""
## [1] "~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2"
## [1] ""
##             (Intercept) log10Ige     age GenderM Mothers_Asthma_Hx2
## Down          18163    17978  55115   36826                  0
## NotSig        157720   768997 710976  726052                 790798
## Up            614915    3823  24707   27920                  0
##             smoking_early_life2 Bcell   CD4T   CD8T   Mono    Neu    NK    SV1
## Down           0 105978 110763 112236  63008  72182 101813 310517
## NotSig        790798 351737 379119 424133 378110 324446 444163 228485
## Up            0 333083 300916 254429 349680 394170 244822 251796
##             SV2
## Down         89058
## NotSig       442801
## Up          258939
## [1] "significant probes (bonferroni threshold):"
##
## FALSE TRUE
## 785435 5363
## [1] "significant probes (adj.p-value < 0.05):"
##
## FALSE TRUE
## 768997 21801
## [1] "significant probes (adj.p-value <= 0.10):"
##
## FALSE TRUE
## 762802 27996
## [1] "significant probes (adj.p-value <= 0.20):"
##
## FALSE TRUE
## 750395 40403
## [1] "significant probes (p-value < 0.05):"
##
## FALSE TRUE
## 703463 87335

# 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 # 1.336503 for the selected model with ct and 2 SVs
## [1] 1.336503

png(file = file.path(plots.dir, "qqplot_log10Ige_CRA.png"),
    width = 540, height = 580)
qq(topHits$P.Value, main="q-q plot CRA log10Ige")
dev.off()

## pdf
## 2
```

## 6 Applying Bacon to tstats

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

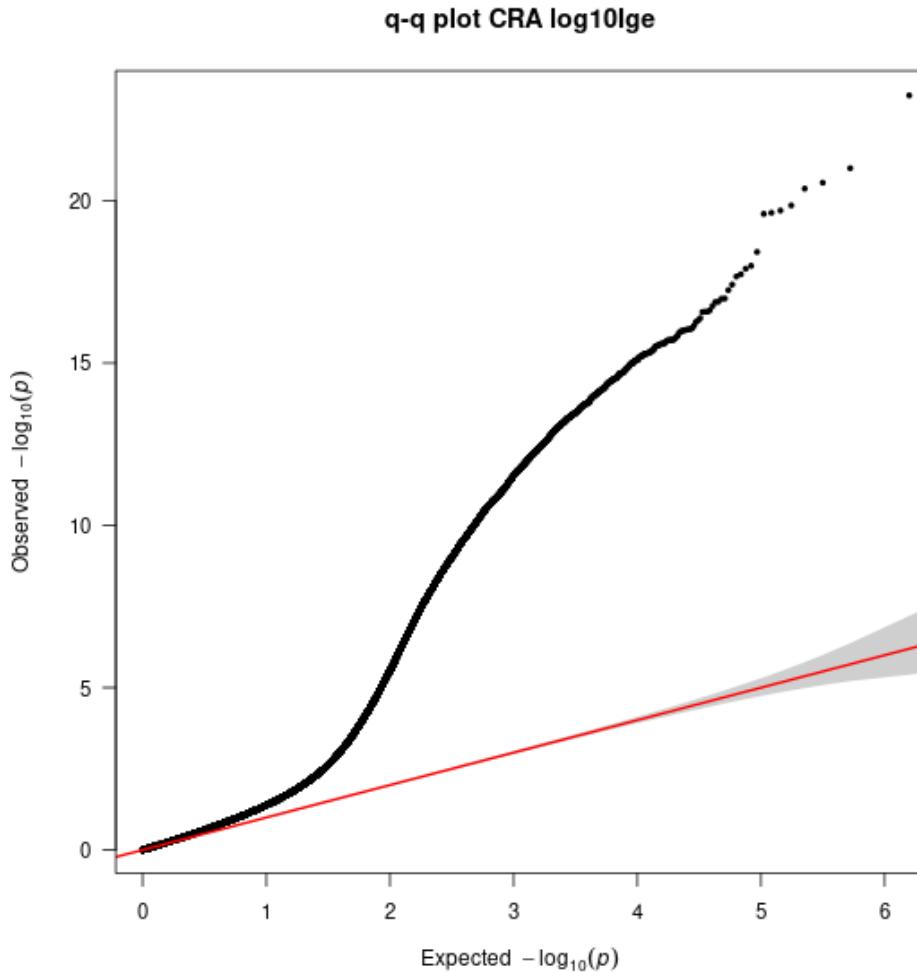


Figure 2: Q-Q CRA

```

##          p.0          p.1          p.2        mu.0        mu.1        mu.2    sigma.0
## [1,] 0.9564738 0.008918905 0.03460729 -0.3062579 2.932686 -3.395995 1.062413
##      sigma.1 sigma.2
## [1,] 1.048049 1.641935

str(bc)

## Formal class 'Bacon' [package "bacon"] with 9 slots
##   ..@ teststatistics: num [1:790798, 1] -10.45 -9.87 -9.76 -9.71 -9.57 ...
##   ..@ effectsizes : num [1, 1] 1
##   ..@ standarderrors: num [1, 1] 1
##   ..@ traces       : num [1:5000, 1:9, 1] 0.992 0.99 0.988 0.986 0.984 ...
##   ... .- attr(*, "dimnames")=List of 3
##   ... .$. : NULL
##   ... .$. : chr [1:9] "p.0" "p.1" "p.2" "mu.0" ...
##   ... .$. : NULL
##   ..@ estimates     : num [1, 1:9] 0.95647 0.00892 0.03461 -0.30626 2.93269 ...
##   ... .- attr(*, "dimnames")=List of 2
##   ... .$. : 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
## -0.3062579

inflation(bc) # using bacon, inflation is 1.062275

## sigma.0
## 1.062413

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

## pdf
## 2

png(file = file.path(plots.dir, "distribution_zscores_tstats_CRA.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: 1.085221
chisq <- qchisq(1-P,1)
lambda1 = median(chisq)/qchisq(0.5,1)
lambda1

## [1] 1.084922

topHits.bacon$fdp.bc. <- p.adjust(topHits.bacon$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("CRA_DMPs_topHits_all_log10Ige_ct2SVs_bacon_",timeStamp,".txt")),
            sep="\t",row.names=F,quote=F)

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

```

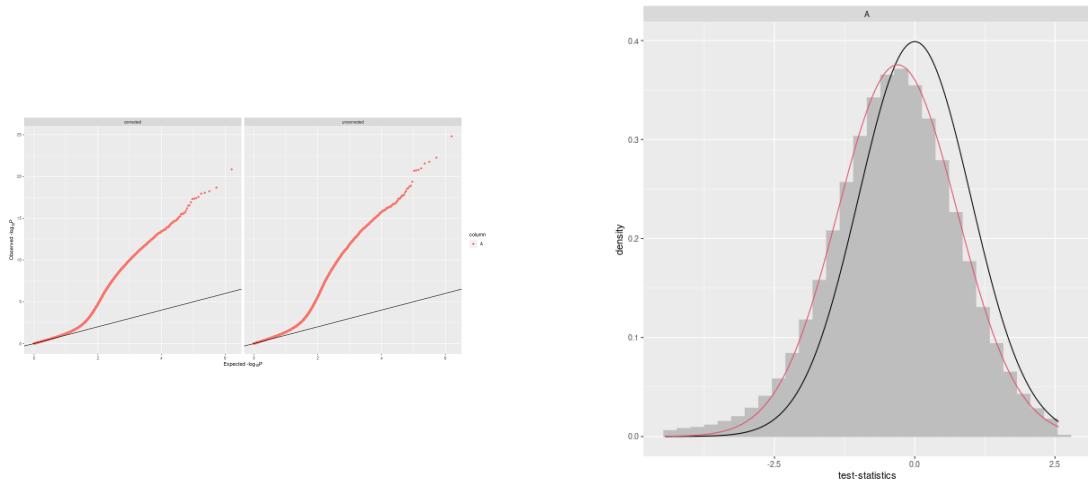


Figure 3: a) Q-Q CRA bacon b) distribution zscores tstats CRA

## 7 Regional analysis

```
# selected model
formula <- c(~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+
            Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2")
vars.of.interest <- c("log10Ige")
varNames <- c("log10Ige")

# 1 missing for log10Ige remove before, others removed 8
remove <- c(list(which(is.na(pData.pheno.cra$age) | is.na(pData.pheno.cra$log10Ige) |
           is.na(pData.pheno.cra$Mothers_Asthma_Hx) | is.na(pData.pheno.cra$smoking_early_life) |
           is.na(pData.pheno.cra$Gender))))
length(remove[[1]])

## [1] 8

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

ids.keep <- intersect(pData.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)

pData.reg$Gender <- relevel(pData.reg$Gender, ref='F')
designCRA = model.matrix(~log10Ige+age+Gender+Mothers_Asthma_Hx+smoking_early_life+
            Bcell+CD4T+CD8T+Mono+Neu+NK+SV1+SV2, data=pData.reg)
myannotationCRA <- cpg.annotate("array", myMs, what="M",
        annotation=c(array = "IlluminaHumanMethylationEPIC",
        annotation = "ilmn10b4.hg19"), arraytype = "EPIC",
        analysis.type="differential", design=designCRA,
        coef="log10Ige", fdr = 0.05)
# Your contrast returned 23240 individually significant probes.
# We recommend the default setting of pcutoff in dmrcrate().
dmrcoutputCRA <- dmrcrate(myannotationCRA, lambda=1000, C=2, pcutoff = "fdr")
results.rangesCRA <- extractRanges(dmrcoutputCRA, genome = "hg19")
DMRsCRA <- data.frame(results.rangesCRA)
DMRsCRA <- DMRsCRA[order(DMRsCRA$Stouffer),]
dim(DMRsCRA) # 4,647

## [1] 4647 13
```

```

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

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

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

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

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -0.026960 -0.007809 -0.005069 -0.004203 -0.002559  0.023338

head(DMRsCRA.sig, n=10)

##      seqnames      start        end width strand no.cpgs min_smoothed_fdr
## 1    chr20 35503983 35504553    571     *       8 6.106768e-91
## 4    chr17 56269170 56270828   1659     *       8 8.308882e-68
## 3    chr3 3151679 3152916   1238     *       9 1.580564e-85
## 8    chr14 100610186 100610667   482     *       4 2.301473e-59
## 7    chr5 132008525 132010740   2216     *       9 3.612939e-53
## 2    chr1 27240319 27241913   1595     *      11 1.210829e-86
## 6    chr10 114437411 114438633  1223     *       7 3.497963e-60
## 15   chr12 109568238 109569180   943     *       6 1.482069e-39
## 11   chr22 50983415 50986051  2637     *      12 3.694647e-63
## 21   chr15 99443213 99443666   454     *       4 7.599770e-43
##          Stouffer        HMFDR        Fisher      maxdiff      meandiff
## 1 4.417295e-60 1.355871e-16 2.049241e-66 -0.02507541 -0.01573926
## 4 6.750150e-54 2.375937e-13 8.898446e-56 -0.02267981 -0.01493122
## 3 4.022091e-53 9.796774e-13 1.275866e-62 -0.02609584 -0.01776738
## 8 5.141221e-47 9.800322e-15 2.983353e-45 -0.02980263 -0.01992335
## 7 6.394615e-43 2.422358e-12 3.255699e-46 -0.02421982 -0.01103724
## 2 1.147791e-42 8.284038e-14 2.403202e-63 -0.02107842 -0.01086252
## 6 5.890393e-38 2.753732e-13 1.253698e-46 -0.02212060 -0.01174225
## 15 2.936356e-34 8.338758e-13 5.781702e-34 -0.02319301 -0.01521277
## 11 6.217772e-33 1.239660e-12 3.793708e-43 -0.02208861 -0.01236503
## 21 1.742301e-32 1.490692e-13 9.510525e-32 -0.02702223 -0.01831677
##      overlapping.genes Gene1st           Coordinates
## 1                  TLDC2    TLDC2 chr20_35503983_35504553
## 4                  EPX      EPX chr17_56269170_56270828
## 3                  IL5RA    IL5RA chr3_3151679_3152916
## 8                  EVL      EVL chr14_100610186_100610667
## 7                  IL4      IL4 chr5_132008525_132010740
## 2          NUDC, NROB2    NUDC    chr1_27240319_27241913
## 6 VTI1A, RP11-25C19.3 VTI1A    chr10_114437411_114438633
## 15          ACACB    ACACB chr12_109568238_109569180
## 11          CTA-384D8.31    CTA    chr22_50983415_50986051
## 21          IGF1R    IGF1R chr15_99443213_99443666

dim(DMRsCRA.sig) # 3,685

## [1] 3685 15

DMRsCRA.sig$abs_meandiffCRA <- abs(DMRsCRA.sig$meandiff)

save(myannotationCRA, dmrcoutputCRA,
  file=file.path(results.dir, paste0("CRARegional_DMRS_hg19_results_logIgE_",

```

```

        timeStamp, ".RData")))

write.table(DMRsCRA.sig, file.path(results.dir,
  paste0("CRA_DMRs_dmrcate_logIgE_hg19_fdr_", timeStamp, ".txt")),
  sep="\t", quote=F, row.names=F)

```

## 8 Session information

[1] "2021-09-01" [1] "2021-09-01 23:36:42 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, geneplotter 1.68.0, GenomeInfoDb 1.26.7, GenomicFeatures 1.42.3, GenomicRanges 1.42.0, GGally 2.1.0, ggplot2 3.3.5, ggpibr 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, 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, 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, nmet 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, readxl 1.3.1, 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, scirme 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