

# CENTRAL analysis

Stephan H. Bernhart

15/Jul/2020

## Load necessary packages

```
library(minfi)
library(FlowSorted.Blood.EPIC) # for cell counting
library(ggplot2)
library(coda.base) # for aitchinson distance computation
library(CMplot)
library(dendextend)
library(plotROC)
library(sva)
library(missMethyl)
```

## Generate correlation function

```
cor_fun <- function(x, y, method){
  tmp <- cor.test(x, y, method= method)
  cbind(r=tmp$estimate, p=tmp$p.value) }
```

## Input data

Read metharray sheet from “Image\_Data\_103343” directory and check result

```
targets <- read.metharray.sheet("Image_Data_103343/")
```

```
head(targets)
```

```
##      Sample_Name Sample_Well Sample_Plate Sample_Group Pool_ID
## 1 103343-001-001          A01           1        NA    <NA>
## 2 103343-001-002          B01           1        NA    <NA>
## 3 103343-001-003          C01           1        NA    <NA>
## 4 103343-001-004          D01           1        NA    <NA>
## 5 103343-001-005          E01           1        NA    <NA>
## 6 103343-001-006          F01           1        NA    <NA>
##      Customer.ID   Array      Slide
## 1 SXS-Control_Plate1 R01C01 202060330159
## 2          20_CENTRAL_T0 R02C01 202060330159
## 3          20_CENTRAL_T18 R03C01 202060330159
## 4          338_CENTRAL_T0 R04C01 202060330159
## 5          338_CENTRAL_T18 R05C01 202060330159
## 6          124_CENTRAL_T0 R06C01 202060330159
```

```

##                                Basename
## 1 Image_Data_103343/202060330159/202060330159_R01C01
## 2 Image_Data_103343/202060330159/202060330159_R02C01
## 3 Image_Data_103343/202060330159/202060330159_R03C01
## 4 Image_Data_103343/202060330159/202060330159_R04C01
## 5 Image_Data_103343/202060330159/202060330159_R05C01
## 6 Image_Data_103343/202060330159/202060330159_R06C01

Read in Data
RGset <- read.metharray.exp(targets = targets)

Check input data
RGset

## class: RGChannelSet
## dim: 1051815 244
## metadata(0):
## assays(2): Green Red
## rownames(1051815): 1600101 1600111 ... 99810990 99810992
## rowData names(0):
## colnames(244): SXS-Control_Plate1 20_CENTRAL_T0 ... 136_CENTRAL_T18
##   SXS-Control Plate4
## colData names(10): Sample_Name Sample_Well ... Basename filenames
## Annotation
##   array: IlluminaHumanMethylationEPIC
##   annotation: ilm10b4.hg19

phenotypes<-read.csv("Pheno_all_CENTRAL_IDs.csv",header=T,row.names=1)

Load phenotype data

qcReport(RGset)

Perform quality control

Estimate Cellcounts

Cellcounts<-estimateCellCounts2(RGset,compositeCellType = "Blood",
processMethod = "preprocessNoob",
referencePlatform=c("IlluminaHumanMethylationEPIC"),
IDOLOptimizedCpGs = IDOLOptimizedCpGs,
cellTypes = c("CD8T", "CD4T", "NK","Bcell", "Mono", "Neu"))
rownames(Cellcounts$counts)<-RGset$Customer.ID

Put out Cellcounts table
write.csv(Cellcounts$counts,file="Cellcounts2_names.csv")

```

## Analyse Cell counts

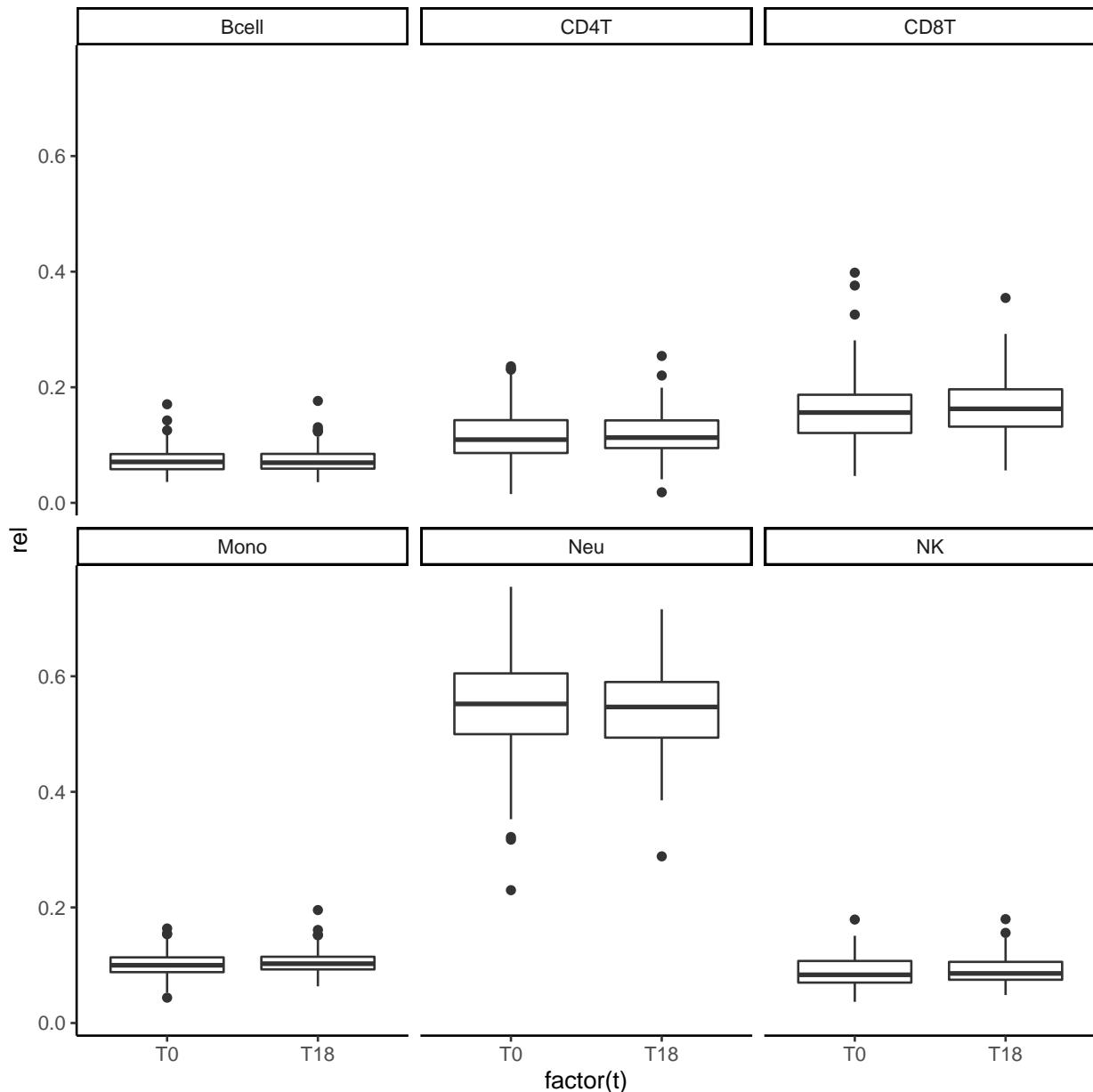
```
Use perl to generate input for ggplot cat Cellcounts2_names.csv | perl -F, -wlane 'next
if (/SXS/); if(/Neu/){for($i=1; $i<=$#F; $i++) {$F[$i]=~s/\//g;$t[$i]=$F[$i];}next;}
$F[0]=~s/\//g;@x=split('/',$F[0]); for($i=1;$i<=$#F; $i++) { print "$x[0]\t$x[-1]\t$F[$i]\t$t[$i]";}
' > Cellcounts2_names.Rin2
```

Read input for ggplot

```
cellcounts<-read.table("Cellcounts2_names.Rin2",sep="\t",header=F,
col.names=c("id","t","rel","celltype"))
```

Create plot of time dependency of blood cell distribution

```
ggplot(cellcounts,aes(x=factor(t), y=rel)) + geom_boxplot() +
facet_wrap(~celltype) + theme_classic()
```

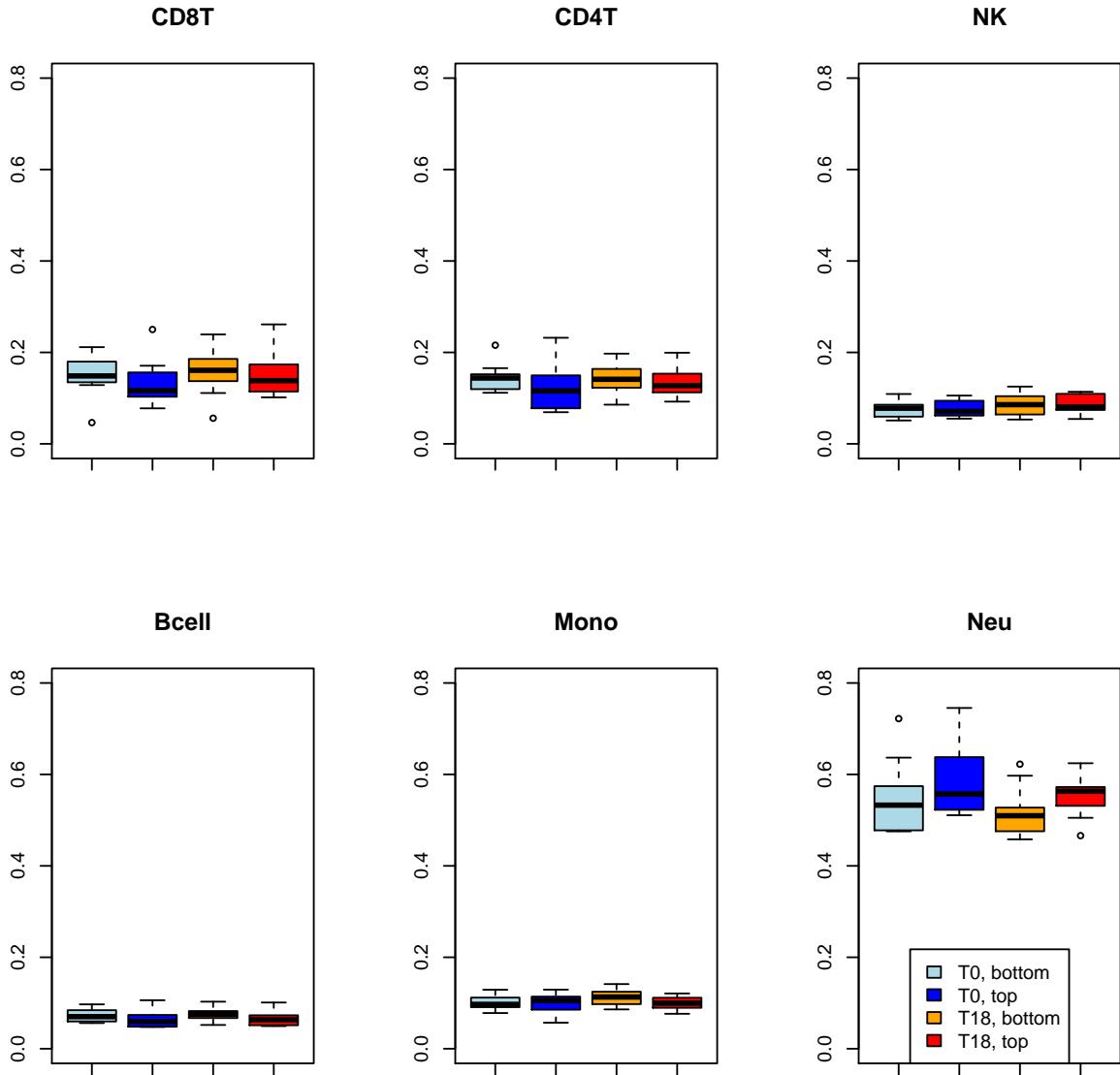


Create clustering of Cellcounts

```
Cellc<-Cellcounts$counts[-c(1,98,195,244),]
```

Check for intervention influence on Cellcounts

```
T0cellc<-Cellc[grep("T0",rownames(Cellc)),]
T18cellc<-Cellc[grep("T18",rownames(Cellc)),]
boxcol<-c("lightblue","blue","orange","red")
par(mfrow=c(2,3))
boxplot(T0cellc[phenotypes[rownames(T0cellc), 2] == "2", 1],
T0cellc[phenotypes[rownames(T0cellc), 2] == "1", 1],
T18cellc[phenotypes[rownames(T18cellc), 2] == "2", 1],
T18cellc[phenotypes[rownames(T18cellc), 2] == "1", 1],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[1])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 2] == "2", 2],
T0cellc[phenotypes[rownames(T0cellc), 2] == "1", 2],
T18cellc[phenotypes[rownames(T18cellc), 2] == "2", 2],
T18cellc[phenotypes[rownames(T18cellc), 2] == "1", 2],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[2])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 2] == "2", 3],
T0cellc[phenotypes[rownames(T0cellc), 2] == "1", 3],
T18cellc[phenotypes[rownames(T18cellc), 2] == "2", 3],
T18cellc[phenotypes[rownames(T18cellc), 2] == "1", 3],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[3])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 2] == "2", 4],
T0cellc[phenotypes[rownames(T0cellc), 2] == "1", 4],
T18cellc[phenotypes[rownames(T18cellc), 2] == "2", 4],
T18cellc[phenotypes[rownames(T18cellc), 2] == "1", 4],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[4])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 2] == "2", 5],
T0cellc[phenotypes[rownames(T0cellc), 2] == "1", 5],
T18cellc[phenotypes[rownames(T18cellc), 2] == "2", 5],
T18cellc[phenotypes[rownames(T18cellc), 2] == "1", 5],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[5])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 2] == "2", 6],
T0cellc[phenotypes[rownames(T0cellc), 2] == "1", 6],
T18cellc[phenotypes[rownames(T18cellc), 2] == "2", 6],
T18cellc[phenotypes[rownames(T18cellc), 2] == "1", 6],
col=boxcol,ylim=c(0,0.8),main=colnames(T18cellc)[6])
legend("bottom",c("T0, bottom","T0, top","T18, bottom","T18, top"),
fill=boxcol,col=boxcol)
```



```

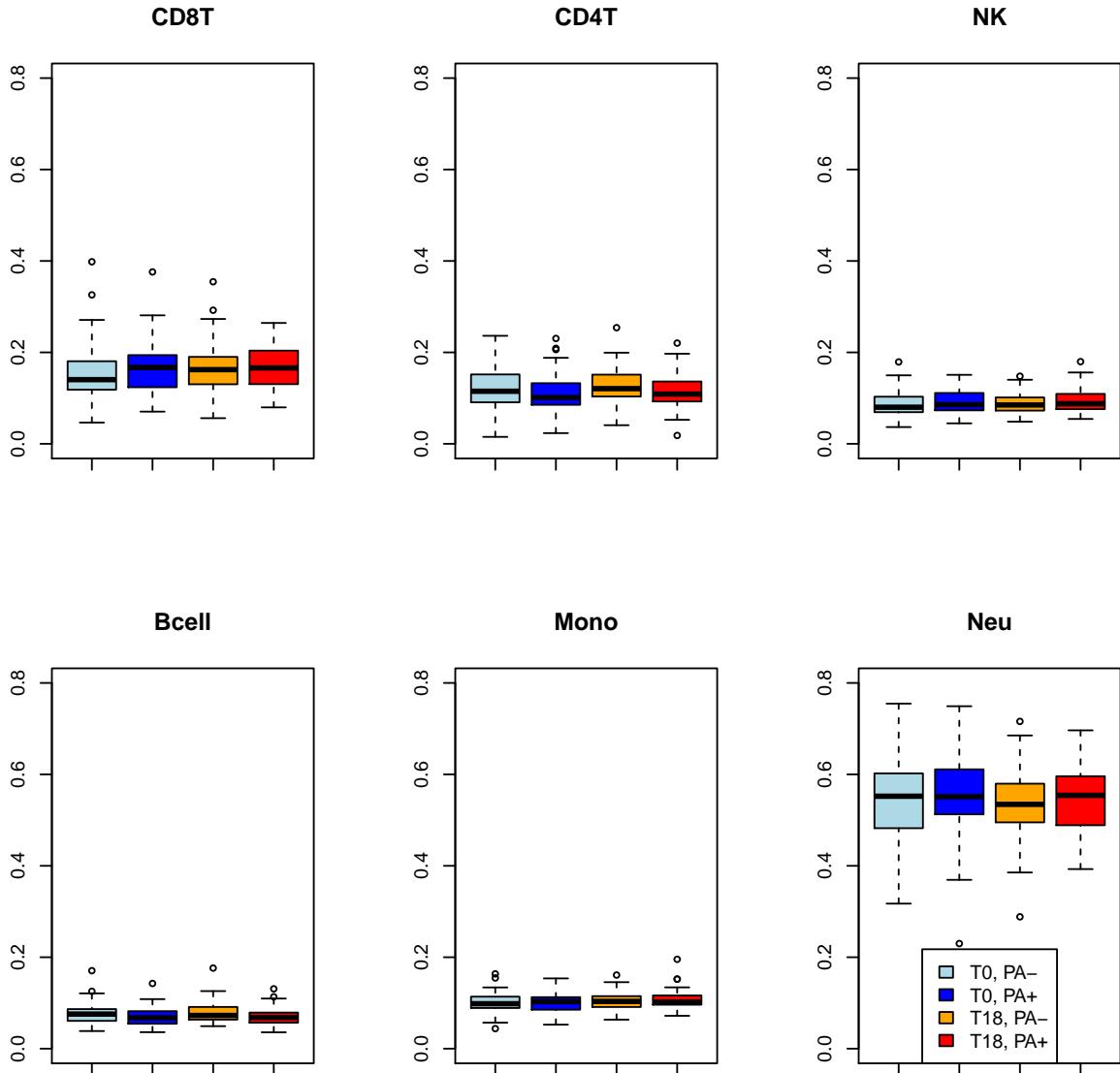
par(mfrow=c(2,3))
boxplot(T0cell1c[phenotypes[rownames(T0cell1c), 7] == "PA-", 1],
T0cell1c[phenotypes[rownames(T0cell1c), 7] == "PA+", 1],
T18cell1c[phenotypes[rownames(T18cell1c), 7] == "PA-", 1],
T18cell1c[phenotypes[rownames(T18cell1c), 7] == "PA+", 1],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cell1c)[1])
boxplot(T0cell1c[phenotypes[rownames(T0cell1c), 7] == "PA-", 2],
T0cell1c[phenotypes[rownames(T0cell1c), 7] == "PA+", 2],
T18cell1c[phenotypes[rownames(T18cell1c), 7] == "PA-", 2],
T18cell1c[phenotypes[rownames(T18cell1c), 7] == "PA+", 2],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cell1c)[2])
boxplot(T0cell1c[phenotypes[rownames(T0cell1c), 7] == "PA-", 3],
T0cell1c[phenotypes[rownames(T0cell1c), 7] == "PA+", 3],
T18cell1c[phenotypes[rownames(T18cell1c), 7] == "PA-", 3],

```

```

T18cellc[phenotypes[rownames(T18cellc), 7] == "PA+", 3],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[3])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 7] == "PA-", 4],
T0cellc[phenotypes[rownames(T0cellc), 7] == "PA+", 4],
T18cellc[phenotypes[rownames(T18cellc), 7] == "PA-", 4],
T18cellc[phenotypes[rownames(T18cellc), 7] == "PA+", 4],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[4])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 7] == "PA-", 5],
T0cellc[phenotypes[rownames(T0cellc), 7] == "PA+", 5],
T18cellc[phenotypes[rownames(T18cellc), 7] == "PA-", 5],
T18cellc[phenotypes[rownames(T18cellc), 7] == "PA+", 5],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[5])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 7] == "PA-", 6],
T0cellc[phenotypes[rownames(T0cellc), 7] == "PA+", 6],
T18cellc[phenotypes[rownames(T18cellc), 7] == "PA-", 6],
T18cellc[phenotypes[rownames(T18cellc), 7] == "PA+", 6],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[6])
legend("bottom",c("T0, PA-","T0, PA+","T18, PA-","T18, PA+"),
fill=boxcol,col=boxcol)

```

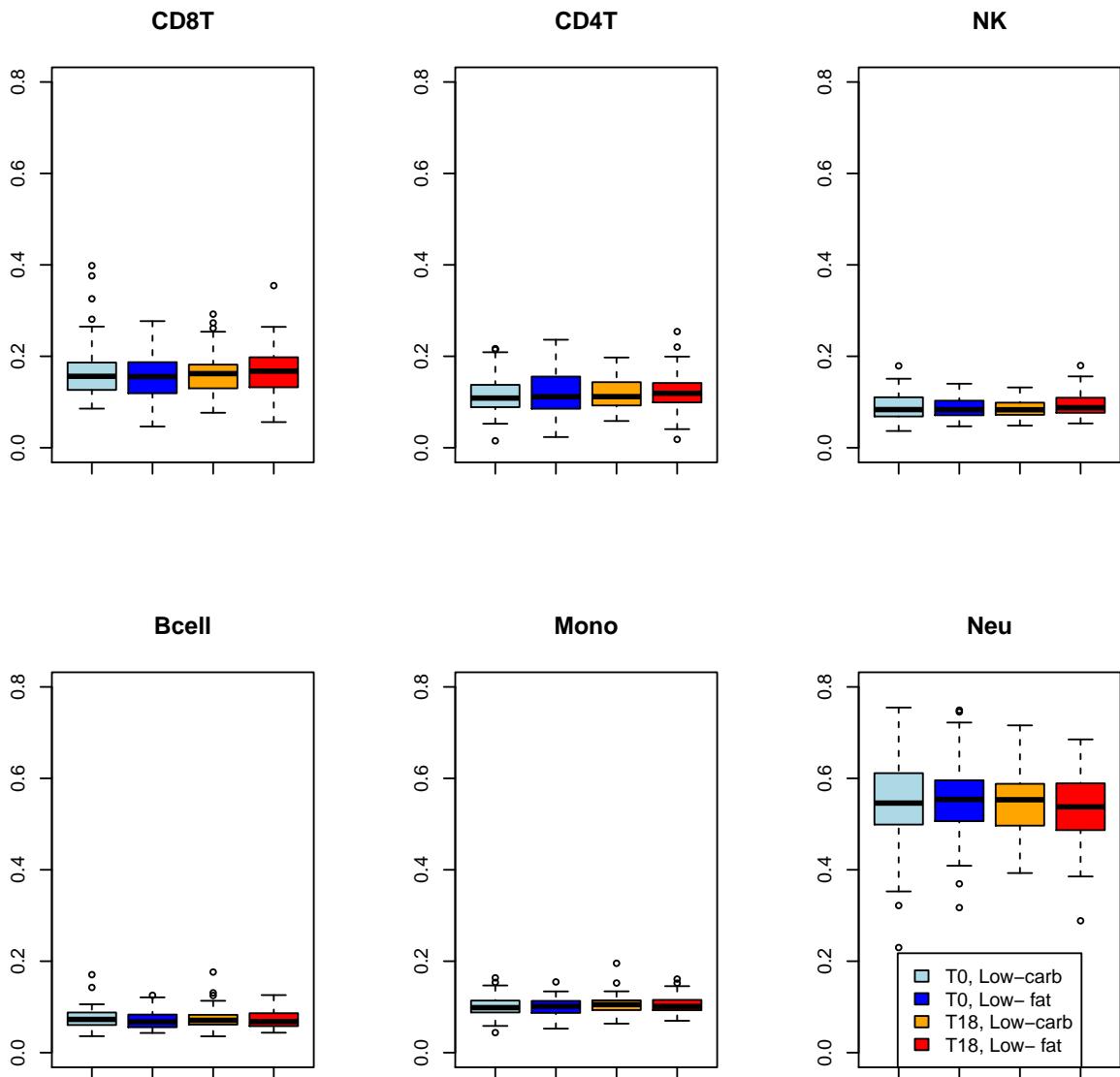


```
par(mfrow=c(2,3))
boxplot(T0cell1c[phenotypes[rownames(T0cell1c), 5] == "Low-carb", 1],
T0cell1c[phenotypes[rownames(T0cell1c), 5] == "Low- fat", 1],
T18cell1c[phenotypes[rownames(T18cell1c), 5] == "Low-carb", 1],
T18cell1c[phenotypes[rownames(T18cell1c), 5] == "Low- fat", 1],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cell1c)[1])
boxplot(T0cell1c[phenotypes[rownames(T0cell1c), 5] == "Low-carb", 2],
T0cell1c[phenotypes[rownames(T0cell1c), 5] == "Low- fat", 2],
T18cell1c[phenotypes[rownames(T18cell1c), 5] == "Low-carb", 2],
T18cell1c[phenotypes[rownames(T18cell1c), 5] == "Low- fat", 2],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cell1c)[2])
boxplot(T0cell1c[phenotypes[rownames(T0cell1c), 5] == "Low-carb", 3],
T0cell1c[phenotypes[rownames(T0cell1c), 5] == "Low- fat", 3],
T18cell1c[phenotypes[rownames(T18cell1c), 5] == "Low-carb", 3],
```

```

T18cellc[phenotypes[rownames(T18cellc), 5] == "Low- fat", 3],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[3])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 5] == "Low-carb", 4],
T0cellc[phenotypes[rownames(T0cellc), 5] == "Low- fat", 4],
T18cellc[phenotypes[rownames(T18cellc), 5] == "Low-carb", 4],
T18cellc[phenotypes[rownames(T18cellc), 5] == "Low- fat", 4],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[4])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 5] == "Low-carb", 5],
T0cellc[phenotypes[rownames(T0cellc), 5] == "Low- fat", 5],
T18cellc[phenotypes[rownames(T18cellc), 5] == "Low-carb", 5],
T18cellc[phenotypes[rownames(T18cellc), 5] == "Low- fat", 5],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[5])
boxplot(T0cellc[phenotypes[rownames(T0cellc), 5] == "Low-carb", 6],
T0cellc[phenotypes[rownames(T0cellc), 5] == "Low- fat", 6],
T18cellc[phenotypes[rownames(T18cellc), 5] == "Low-carb", 6],
T18cellc[phenotypes[rownames(T18cellc), 5] == "Low- fat", 6],
ylim=c(0,0.8),col=boxcol,main=colnames(T18cellc)[6])
legend("bottom",c("T0, Low-carb","T0, Low- fat","T18, Low-carb","T18, Low- fat"),
fill=boxcol,col=boxcol)

```



## Create Beta

Do quantile normalization and create GRset

```
GRset.quantile <- preprocessQuantile(RGset, fixOutliers = TRUE,
removeBadSamples = TRUE, badSampleCutoff = 10.5,
quantileNormalize = TRUE, stratified = TRUE, mergeManifest = FALSE)
```

Add annotations to GRset

```
annotation <- getAnnotation(GRset.quantile)
```

rename columns of GRset

```

colnames(GRset.quantile)<-targets$Customer.ID

remove technical control data sets from GRset
onlyTGR<-GRset.quantile[,c(-1,-98,-195,-244)]
```

Create beta values

```
beta <- getBeta(onlyTGR)
```

generate batches for batch correction

```
batch<-targets$Slide[c(-1,-98,-195,-244)]
```

filter probes with bad p-values in more than two data-sets

```

detP <- detectionP(RGset)
keep <- rowSums(detP < 0.01) >= 0.99*ncol(RGset)
tokeep<-which(keep)
filteredbeta<-beta[rownames(beta) %in% names(tokeep),]
pheno<-as.data.frame(phenotypes[,6])
```

Correct for batch effects

```

levels(pheno[,1])[5]<-"T0"
rownames(pheno)<-rownames(phenotypes)
pheno[grep("T0", rownames(phenotypes)),1]<-"T0"
colnames(pheno)<-"intervention"
modcombat = model.matrix(~1, data=pheno)
combat_filteredbeta = ComBat(dat=filteredbeta, batch=batch, mod=modcombat, par.prior=TRUE,
prior.plots=FALSE)
```

Correct for cell type composition

```

beta.lm<-apply(combat_filteredbeta, 1,
function(x){Cellc[colnames(combat_filteredbeta),]->blood
lm(x~CD8T+CD4T+NK+Bcell+Mono+Neu,data=as.data.frame(blood))})
residuals<-t(sapply(beta.lm,function(x)residuals(summary(x))))
colnames(residuals)<-colnames(combat_filteredbeta)
adj.betas<-residuals+matrix(apply(combat_filteredbeta, 1, mean),nrow=nrow(residuals),
ncol=ncol(residuals))
```

append location to beta values

```

Gadj.beta<-cbind(annotation[rownames(annotation) %in% names(tokeep),1:2],adj.betas)
Gcombat_filteredbeta<-cbind(annotation[rownames(annotation) %in% names(tokeep),1:2
],combat_filteredbeta)
```

## Consistency check

Look at the correlation of beta

```

Corr<-cor(adj.betas)
maximumcor<-rep("NA",length(Corr[1,]))
names(maximumcor)<-colnames(Corr)
for (i in 1:length(Corr[1,]))
maximumcor[i]<-names(which.max(Corr[i,-i]))
```

check for inconsistencies: are same patient pairs best correlators?

```

for (i in 1:length(Corr[,])) {
  a<-gsub("_T[01][8]?", "", maximumcor[i])
  b<-gsub("_T[01][8]?", "", names(maximumcor)[i])
  if(a!=b) {
    print(paste(i,maximumcor[i],names(maximumcor)[i],sep=" "))
  }
}

```

## metilene analysis

### generate metilene output

Generate colnames for metilene input

```

colnames(Gadj.beta)[1]<-"chr"
colnames(Gadj.beta)[2]<-"pos"

```

put out beta in metilene input format

```
write.table(Gadj.beta[,],file="Betavalues_metilenein",sep="\t",quote=F,row.names = FALSE)
```

Annotate Top Responders in file “Betavalues\_metilenein” by appending Top to the colnames of the following columns (e.g. T0\_Bottom or T18\_Bottom)::

```
rownames(phenotypes)[phenotypes[,2]==1]
```

```

## [1] "44_CENTRAL_TO"   "44_CENTRAL_T18"   "301_CENTRAL_TO"   "301_CENTRAL_T18"
## [5] "189_CENTRAL_TO"   "189_CENTRAL_T18"   "276_CENTRAL_TO"   "276_CENTRAL_T18"
## [9] "193_CENTRAL_TO"   "193_CENTRAL_T18"   "102_CENTRAL_TO"   "102_CENTRAL_T18"
## [13] "43_CENTRAL_TO"   "43_CENTRAL_T18"   "292_CENTRAL_TO"   "292_CENTRAL_T18"
## [17] "290_CENTRAL_TO"   "290_CENTRAL_T18"   "55_CENTRAL_TO"   "55_CENTRAL_T18"

```

Annotate Bottom Responders in file “Betavalues\_metilenein” by appending Bottom to the colnames of the following columns (eg T0\_Bottom or T18\_Bottom):

```
rownames(phenotypes)[phenotypes[,2]==2]
```

```

## [1] "296_CENTRAL_TO"   "296_CENTRAL_T18"   "225_CENTRAL_TO"   "225_CENTRAL_T18"
## [5] "15_CENTRAL_TO"   "15_CENTRAL_T18"   "174_CENTRAL_TO"   "174_CENTRAL_T18"
## [9] "159_CENTRAL_TO"   "159_CENTRAL_T18"   "70_CENTRAL_TO"   "70_CENTRAL_T18"
## [13] "336_CENTRAL_TO"   "336_CENTRAL_T18"   "72_CENTRAL_TO"   "72_CENTRAL_T18"
## [17] "297_CENTRAL_TO"   "297_CENTRAL_T18"   "20_CENTRAL_TO"   "20_CENTRAL_T18"

```

run methilene

```

methilene_linux64 -M 1000 -d 0.03 -m 3 -t 8 -a Top -b Bottom Betavalues_metilenein >
Betavalues_metilene_Top_metout
filter for adj.p < 0.05
cat Betavalues_metilene_Top_metout | perl -wlane
'print "$F[0]\t$F[1]\t$F[2]\t$F[3]\t$F[4]\t$F[7]" if ($F[3]<0.05);' >
Betavalues_metilene_Top_metout.bed
intersecting with a gencode 19 annotation bed file (including genes and 1500nt upstream) using bedtools
intersect, filtering output
bedtools intersect -wao -a Betavalues_metilene_Top_metout.bed -b
gencode.v19.TSS1500plusgenes.bed | perl -wlane
'$a=".;" if($F[-4]=-/-/){@x=split(/-/,$F[-4]); $a=$x[2];}'
print "$F[0]\t$F[1]\t$F[2]\t$F[3]\t$F[4]\t$F[5]\t$a" if($F[3]<0.05);'| sort -gk 4 >

```

```
Betavalues_metilene_Top_metout.tsv
read in DMRs

TopBottomDMRs<-read.table("Betavalues_metilene_Top_metout.bed",header=F,
col.names=c("chr","start","stop", "adjp","diff","2DKS"))
```

read in metilene results for Manhattan plotting

```
metileneresultsTop<-read.table("Betavalues_metilene_Top_metout",header=F)
colnames(metileneresultsTop)<-c("chr","start","stop","adjp","diff",
"NumberCpGs","p_wilcox","p_2D_KS","meanTop", "meanBottom")
```

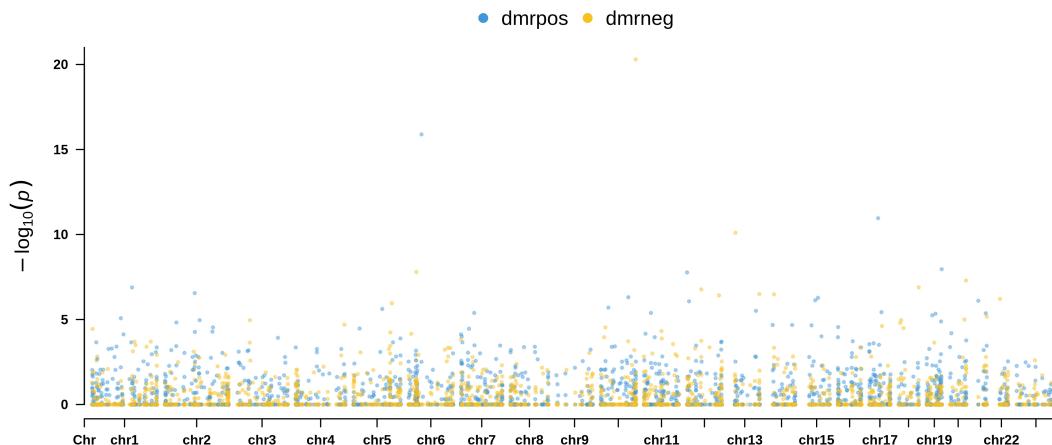
generate one entry for start of DMRs

```
dmrpos<-rep(1,length(metileneresultsTop[,1]))
dmrneg<-rep(1,length(metileneresultsTop[,1]))

for (i in 1:length(metileneresultsTop[,1])){
  if(metileneresultsTop[i,5]<0) {
    dmrneg[i]<-metileneresultsTop[i,8] # use 2D KS p value
  }
  else {
    dmnpos[i]<-metileneresultsTop[i,8] # use 2D KS p value
  }
}
```

Manhattan plotting

```
CMplot(cbind(as.character(metileneresultsTop[,3]),as.character(metileneresultsTop[,1]),
as.integer(metileneresultsTop[,2]),dmrpos,dmrneg), cex=c(0.2,0.5,0.5),
multitracks=TRUE,plot.type="m",file="jpg",file.output=TRUE)
```



## Correlation of methylation T0 and perz. weightloss

```
T0pearsonweightloss<-apply(adj.betas[,grep("T0",colnames(adj.betas))],1,cor_fun,  
as.vector(phenotypes[colnames(adj.betas)][grep("T0",colnames(adj.betas))],284)),method=  
"pearson")  
T0spearmanweightloss<-apply(adj.betas[,grep("T0",colnames(adj.betas))],1,cor_fun,  
as.vector(phenotypes[colnames(adj.betas)][grep("T0",colnames(adj.betas))],284)),method=  
"spearman")
```

adjust p-values

```
T0pearsonad<-p.adjust(T0pearsonweightloss[2,])  
T0spearmanad<-p.adjust(T0spearmanweightloss[2,])
```

create rownames

```
rownames(T0pearsonweightloss)<-c("pearson_corr","pearson_pvalue")  
rownames(T0spearmanweightloss)<-c("spearman_corr","spearman_pvalue")
```

generate weightloss Table wrt to T0 methylation, use geometric means

```
T0sqrtweightloss<-t(rbind(T0pearsonweightloss,T0pearsonad,  
T0spearmanweightloss,T0spearmanad,sqrt(T0pearsonweightloss[2,]*T0spearmanweightloss[2,]),  
sqrt(T0pearsonad*T0spearmanad),(T0pearsonweightloss[1,]+T0spearmanweightloss[1,])/2))
```

add colnames

```
colnames(T0sqrtweightloss)[7]<-"combined_p"  
colnames(T0sqrtweightloss)[8]<-"combined_p_adj"  
colnames(T0sqrtweightloss)[9]<-"combined_corr"
```

sort and write table

```
sortedT0sqrtweightloss<-T0sqrtweightloss[order(T0sqrtweightloss[,7]),]  
write.table(sortedT0sqrtweightloss,file="adjsqrtsortedT0weigthoss",sep="\t",quote=FALSE)
```

## Generate Manhattan plot

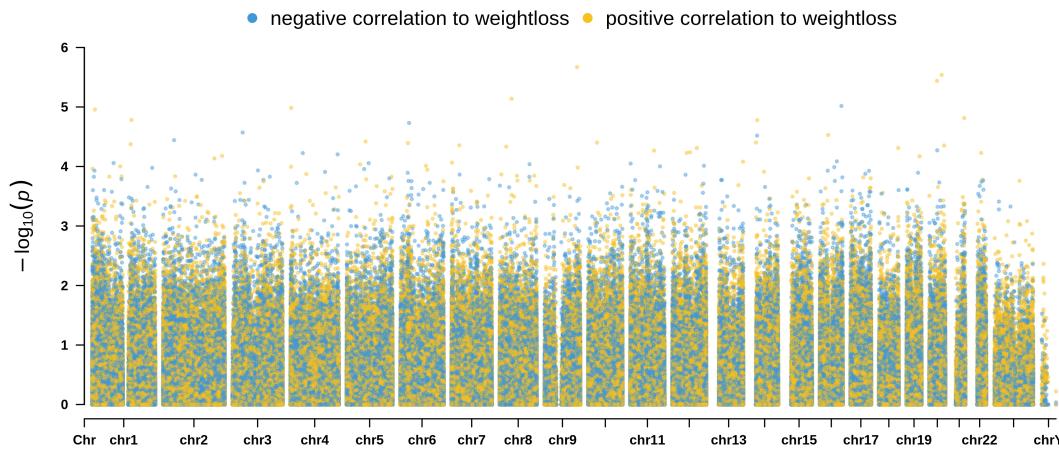
preparation

```
noone<-rep(1,length(T0sqrtweightloss[,7]))  
one<-rep(1,length(T0sqrtweightloss[,7]))  
for (i in 1:length(T0sqrtweightloss[,9])){  
  if(T0sqrtweightloss[i,9]<0) {  
    one[i]<-T0sqrtweightloss[i,7]  
  }  
  else {  
    noone[i]<-T0sqrtweightloss[i,7]  
  }  
}
```

generation

```
cminput<-cbind(rownames(annotation)[rownames(annotation) %in% names(tokeep)],  
annotation[rownames(annotation) %in% names(tokeep),1:2],one,noone)  
colnames(cminput)<-c("", "", "", "negative correlation to weightloss",  
"positive correlation to weightloss")
```

```
CMplot(cminput, cex=c(0.2,0.5,0.5),multitracks=TRUE,plot.type="m",
file="jpg",file.output=TRUE,highlight=dmrcpgs)
```

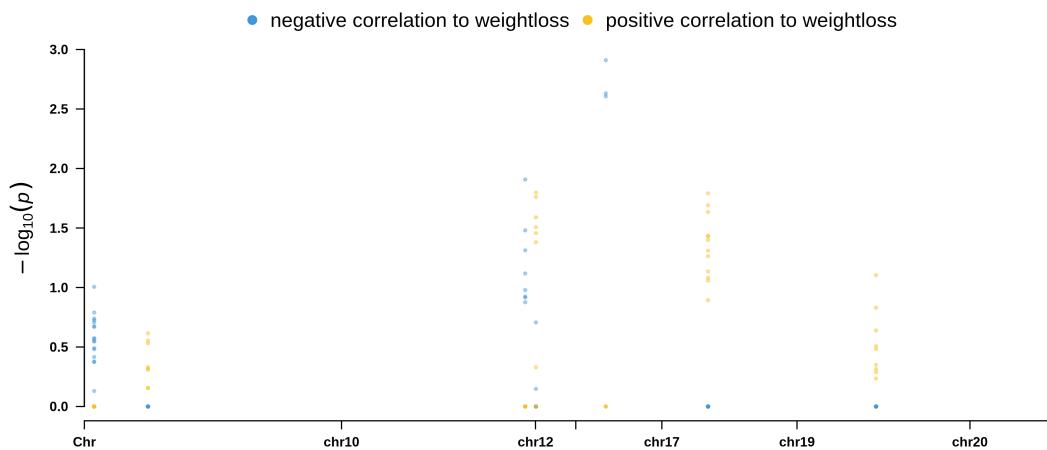


annotate DMRCpgs

```
dmrcpgs<-cminput[cminput[,2]==TopBottomDMRs[1,1] &
cminput[,3] > TopBottomDMRs[1,2] & cminput[,3]<TopBottomDMRs[1,3] ,1]
for (i in 2:8)
dmrcpgs<-c(dmrcpgs,cminput[cminput[,2]==TopBottomDMRs[i,1] &
cminput[,3] >= TopBottomDMRs[i,2] & cminput[,3]<=TopBottomDMRs[i,3] ,1])
```

plot for DMRs only

```
CMplot(cminput[cminput[,1] %in% dmrcpgs,], cex=c(0.2,0.5,0.5),multitracks=TRUE,plot.type="m",
file="jpg",file.output=TRUE)
```



## Compute ROC

Select for negative correlation and p(correlation)<0.01

```
T0beta<-adj.betas[T0sqrtweightloss[,9]<0 & T0sqrtweightloss[,7]<0.01,
grep("T0",colnames(adj.betas))]
```

```

TObeta001<-adj.betas[T0sqrtweightloss[,9]<0 & T0sqrtweightloss[,7]<0.001,
grep("T0", colnames(adj.betas))]
TObetapos001<-adj.betas[T0sqrtweightloss[,9]>0 & T0sqrtweightloss[,7]<0.001,
grep("T0", colnames(adj.betas))]
TObeta0001<-adj.betas[T0sqrtweightloss[,9]<0 & T0sqrtweightloss[,7]<0.0001,
grep("T0", colnames(adj.betas))]
forauc<-colMeans(TObeta[, phenotypes[colnames(TObeta), 4]=="male"])
forauc001<-colMeans(TObeta001[, phenotypes[colnames(TObeta001), 4]=="male"])
foraucpos001<-colMeans(TObetapos001[, phenotypes[colnames(TObetapos001), 4]=="male"])
forauc0001<-colMeans(TObeta0001[, phenotypes[colnames(TObeta0001), 4]=="male"])
TObetapos0001<-adj.betas[T0sqrtweightloss[,9]>0 & T0sqrtweightloss[,7]<0.0001,
grep("T0", colnames(adj.betas))]
foraucpos0001<-colMeans(TObetapos0001[, phenotypes[colnames(TObetapos0001), 4]=="male"])

```

How many CpGs do contribute?

```

length(TObeta001[,1])
length(TObeta0001[,1])
length(TObetapos0001[,1])
length(TObetapos001[,1])

```

define successfull (>5% weightloss) and unsuccessful weightloss (<=0 weightloss)

```

chosen<-rep(0,110)
for (i in 1:110) {
  if (phenotypes[names(forauc)][i], 284] <= 0) {
    chosen[i]=2
  }
  if (phenotypes[names(forauc)][i], 284] > 5) {
    chosen[i]=1
  }
}
Weightloss<-forauc[chosen==1]
Noweightloss<-forauc[chosen==2]
Weightloss001<-forauc001[chosen==1]
Noweightloss001<-forauc001[chosen==2]
Weightlosspos001<-foraucpos001[chosen==1]
Noweightlosspos001<-foraucpos001[chosen==2]
Weightloss0001<-forauc0001[chosen==1]
Noweightloss0001<-forauc0001[chosen==2]
Weightlosspos0001<-foraucpos0001[chosen==1]
Noweightlosspos0001<-foraucpos0001[chosen==2]

```

Generate input for AUC plotting and computation

```

AUCinput<-rbind(cbind(Noweightloss,rep(1,length(Noweightloss))),
  cbind(Weightloss,rep(0,length(Weightloss))))
AUCinput001<-rbind(cbind(Noweightloss001,rep(1,length(Noweightloss001))),
  cbind(Weightloss001,rep(0,length(Weightloss001))))
AUCinput0001<-rbind(cbind(Noweightloss0001,rep(1,length(Noweightloss0001))),
  cbind(Weightloss0001,rep(0,length(Weightloss0001))))
AUCinputpos001<-rbind(cbind(Noweightlosspos001,rep(1,length(Noweightlosspos001))),
  cbind(Weightlosspos001,rep(0,length(Weightlosspos001))))
AUCinputpos0001<-rbind(cbind(Noweightlosspos0001,rep(1,length(Noweightlosspos0001))),
  cbind(Weightlosspos0001,rep(0,length(Weightlosspos0001))))

```

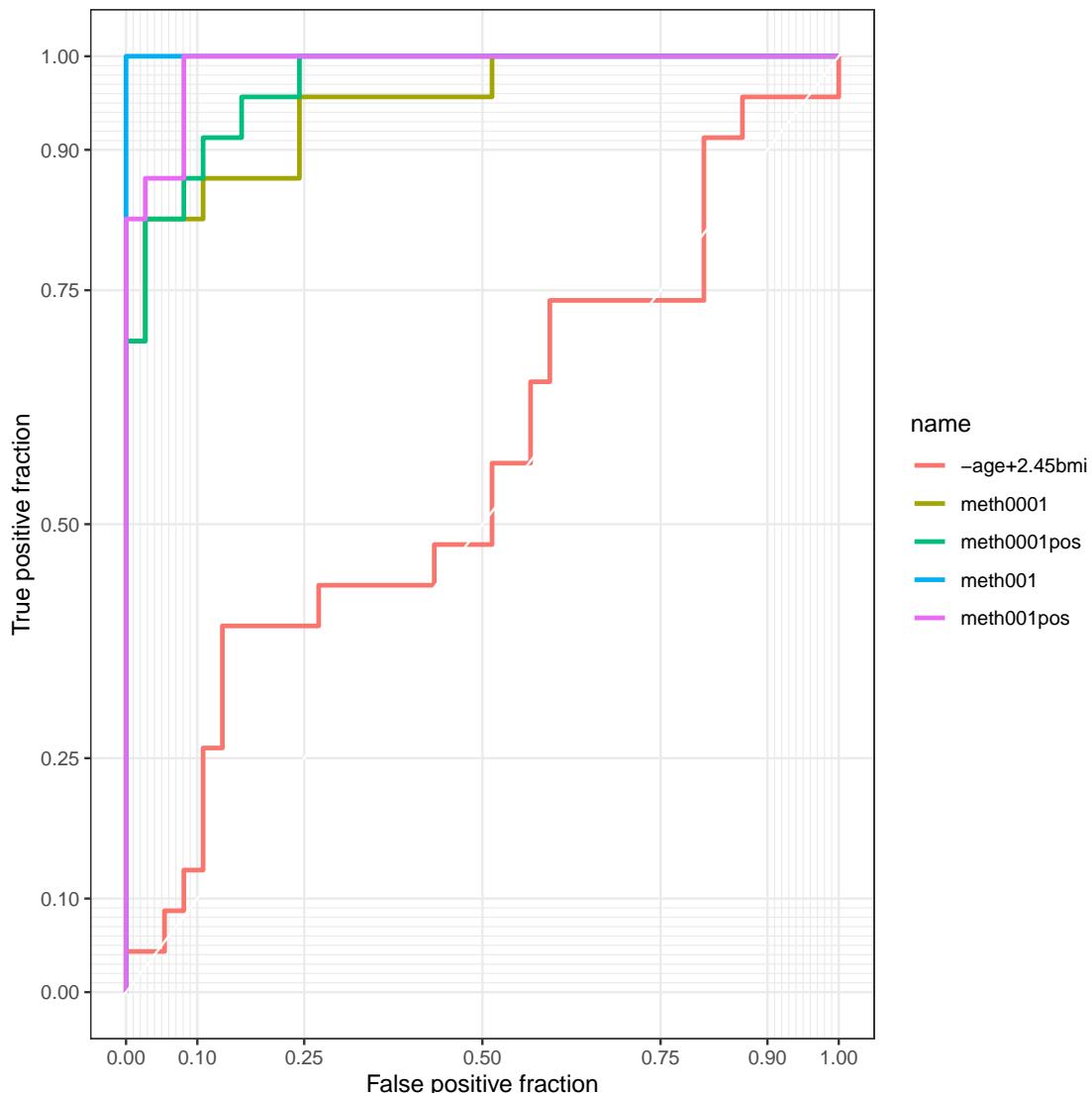
```
AUCinput3<-cbind(AUCinput001,
-1*phenotypes[rownames(AUCinput001),8]+2.45*phenotypes[rownames(AUCinput001),40],
AUCinput0001[,1],-1*AUCinputpos001[,1],-1*AUCinputpos0001[,1])
```

Add colnames and generate data frame for ROC plotting

```
colnames(AUCinput3)<-c("meth001","resp","-age+2.45bmi","meth0001",
"meth001pos","meth0001pos")
longtest <- melt_roc(as.data.frame(AUCinput3), "resp", c("meth001","meth0001",
"meth001pos","meth0001pos","-age+2.45bmi"))
```

Plot ROC curve

```
moreauc<-ggplot(longtest, aes(d = D, m = M, color = name)) + geom_roc(n.cuts=0)
moreauc<-moreauc+ style_roc()
moreauc
```



compute methylation and combined age and bmi AUC

```
calc_auc(moreauc)
```

```
##   PANEL group      AUC
## 1      1    1 0.5640423
## 2      1    2 0.9482961
## 3      1    3 0.9706228
## 4      1    4 1.0000000
## 5      1    5 0.9882491
```

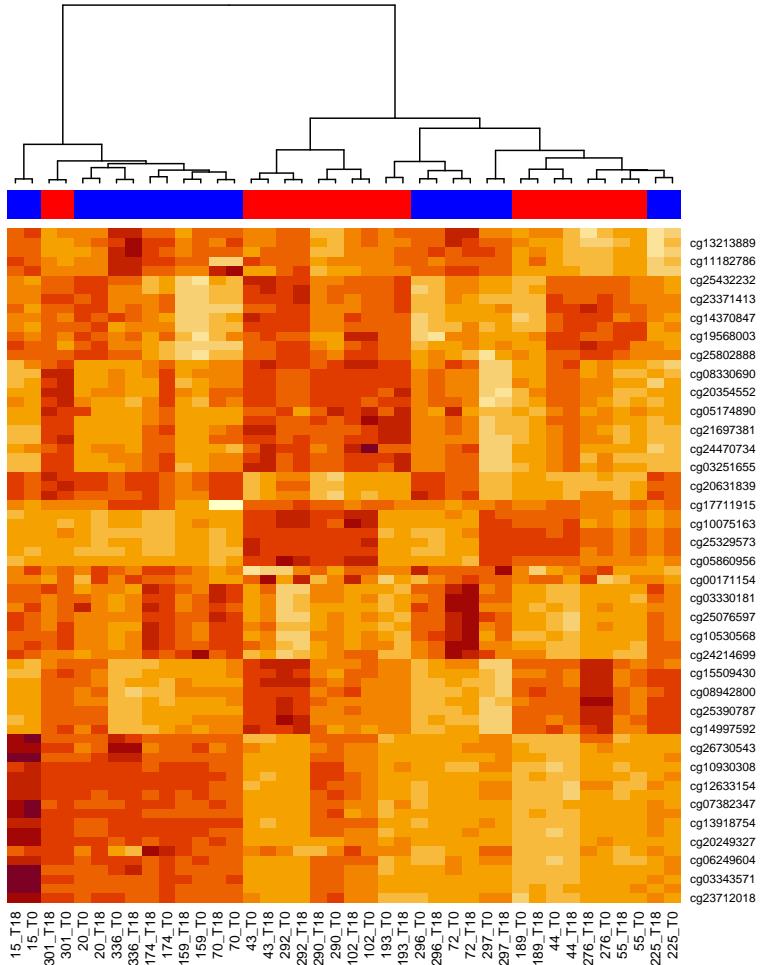
## Generate heatmaps

Heatmap from DMRs Read cg ids from DMRs

```
Metids<-read.table("NeueresBetavalues_metilene.cg",header=F)
```

Generate heatmap from DMR cgs

```
Bottombetas<-adj.betas[which(rownames(adj.betas) %in% Metids[,1]),
rownames(phenotypes)[phenotypes[,2]==2]]
Topbetas<-adj.betas[which(rownames(adj.betas) %in% Metids[,1]),
rownames(phenotypes)[phenotypes[,2]==1]]
colnames(Topbetas)<-gsub("_CENTRAL","",colnames(Topbetas))
colnames(Bottombetas)<-gsub("_CENTRAL","",colnames(Bottombetas))
heatmap(cbind(Topbetas,Bottombetas),Rowv=NA,ColSideColors=c(rep("red",20),rep("blue",20)),
hclustfun = function(x) hclust(x, method="ward.D"))
```



## Look at the differences between T0 and T18

Find dmps

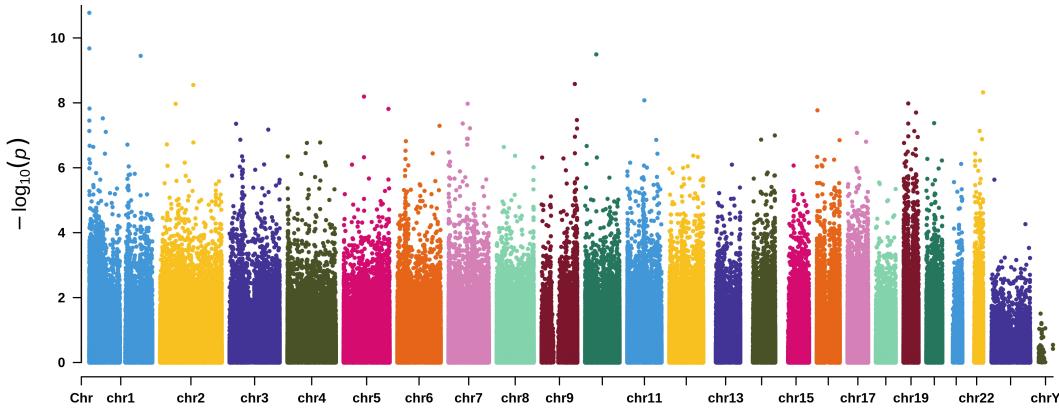
```
newcol<-colnames(onlyTGR)
newcol[grep("T18", colnames(onlyTGR))]<-"T18"
newcol[grep("T0", colnames(onlyTGR))]<-"T0"
t18dmp<-dmpFinder(adj.betas,newcol,type="categorical")
```

put out data

```
write.table(t18dmp,file="adjt0_t18dmp.tsv",sep="\t",quote=FALSE)
head(t18dmp, n=6L)
```

Do Manhattan plot

```
CMplot(cbind(rownames(annotation),annotation[,1:2],t18dmp[rownames(annotation),3]),
cex=c(0.2,0.5,0.5),plot.type="m",file="jpg",file.output=FALSE)
```

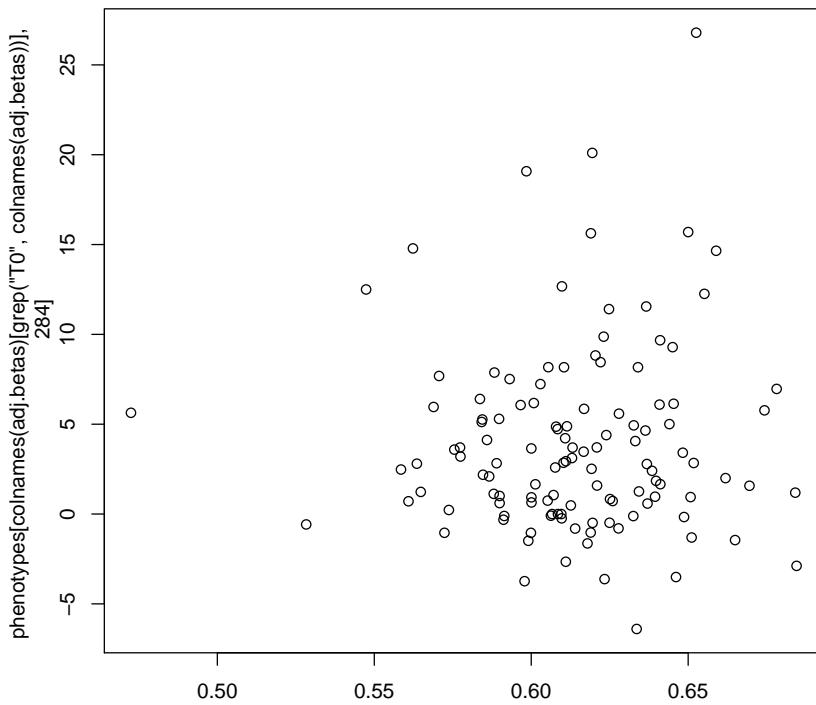


## Single CpG analysis with DMPfinder

dmpFinder beta values vs weightloss

```
dmpsweightloss<-dmpFinder(adj.betas[,grep("T0",colnames(adj.betas))],  
phenotypes[colnames(adj.betas)[grep("T0",colnames(adj.betas))],284],type="continuous")
```

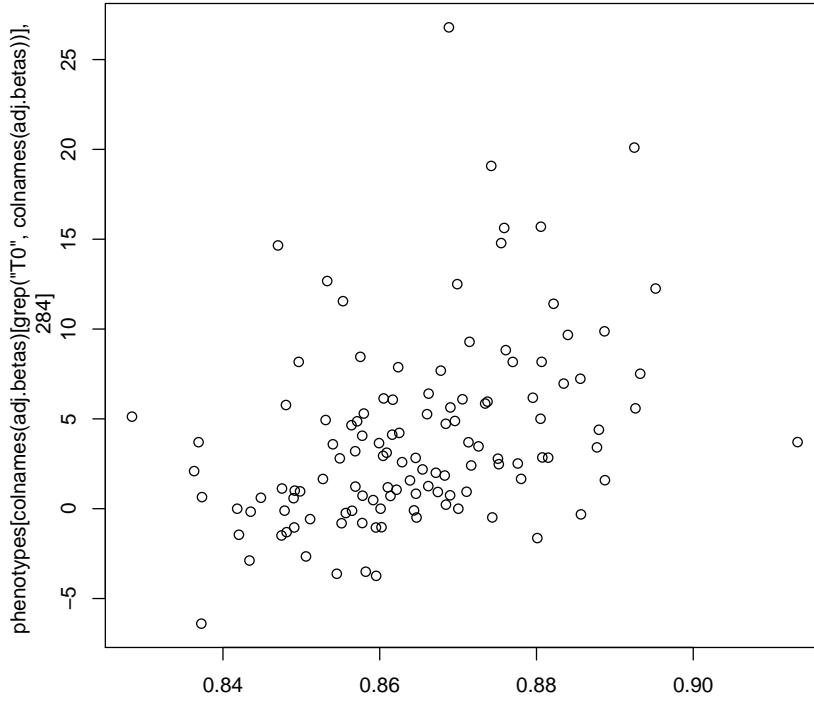
```
plot(beta[rownames(adj.betas)==rownames(dmpsweightloss)[1],grep("T0",colnames(adj.betas))],  
phenotypes[colnames(adj.betas)[grep("T0",colnames(adj.betas))],284])
```



```
beta[rownames(adj.betas) == rownames(dmpsweightloss)[1], grep("T0",  
colnames(adj.betas))]
```

comparison to correlation

```
plot(adj.betas[rownames(adj.betas)==rownames(sortedT0sqrtweightloss)[1],  
grep("T0", colnames(adj.betas))], phenotypes[colnames(adj.betas)[grep("T0",  
colnames(adj.betas))], 284])
```



```
adj.betas[rownames(adj.betas) == rownames(sortedT0sqrtweightloss)[1],  
grep("T0", colnames(adj.betas))]
```

## dmpFinder of subtypes

Intervention types Generate beta matrices for subtypes PA or diet type

```
onlyPA<-adj.betas[,rownames(phenotypes)[which(phenotypes[,7]=="PA+")]]  
onlyno<-adj.betas[,rownames(phenotypes)[which(phenotypes[,7]=="PA-")]]  
onlylowfat<-adj.betas[,rownames(phenotypes)[which(phenotypes[,5]=="Low- fat")]]  
onlylowcarb<-adj.betas[,rownames(phenotypes)[which(phenotypes[,5]=="Low-carb")]]
```

generate categorical names

```
newcolonlyno<-colnames(onlyno)  
newcolonlyPA<-colnames(onlyPA)  
newcolonlylowfat<-colnames(onlylowfat)  
newcolonlylowcarb<-colnames(onlylowcarb)  
newcolonlyno[grep("T18", colnames(onlyno))]<-"T18"  
newcolonlyno[grep("T0", colnames(onlyno))]<-"T0"  
newcolonlyPA[grep("T18", colnames(onlyPA))]<-"T18"
```

```

newcolonlyPA[grep("T0", colnames(onlyPA))] <- "T0"
newcolonlylowfat[grep("T18", colnames(onlylowfat))] <- "T18"
newcolonlylowfat[grep("T0", colnames(onlylowfat))] <- "T0"
newcolonlylowcarb[grep("T18", colnames(onlylowcarb))] <- "T18"
newcolonlylowcarb[grep("T0", colnames(onlylowcarb))] <- "T0"

```

DMPfinder

```

t18dmponlylowcarb <- dmpFinder(onlylowcarb, newcolonlylowcarb, type = "categorical")
t18dmponlylowfat <- dmpFinder(onlylowfat, newcolonlylowfat, type = "categorical")
t18dmponlyno <- dmpFinder(onlyno, newcolonlyno, type = "categorical")
t18dmponlyPA <- dmpFinder(onlyPA, newcolonlyPA, type = "categorical")

```

write and inspect data

```

write.table(t18dmponlylowfat, file = "adjt0_t18dmponlylowfat.tsv", sep = "\t", quote = FALSE)
write.table(t18dmponlylowcarb, file = "adjt0_t18dmponlylowcarb.tsv", sep = "\t", quote = FALSE)
write.table(t18dmponlyno, file = "adjt0_t18dmponlyno.tsv", sep = "\t", quote = FALSE)
write.table(t18dmponlyPA, file = "adjt0_t18dmponlyPA.tsv", sep = "\t", quote = FALSE)

```

diet combined with PA+

```

lowcarbno <- adj.betas[, rownames(phenotypes)[which(phenotypes[, 6] == "Low-carb")]]
lowcarbPA <- adj.betas[, rownames(phenotypes)[which(phenotypes[, 6] == "Low-carb PA")]]
lowfatno <- adj.betas[, rownames(phenotypes)[which(phenotypes[, 6] == "Low-fat")]]
lowfatPA <- adj.betas[, rownames(phenotypes)[which(phenotypes[, 6] == "Low-fat PA")]]

```

generate categorical names

```

newcollowcarbno <- colnames(lowcarbno)
newcollowcarbPA <- colnames(lowcarbPA)
newcollowfatno <- colnames(lowfatno)
newcollowfatPA <- colnames(lowfatPA)
newcollowcarbno[grep("T18", colnames(lowcarbno))] <- "T18"
newcollowcarbno[grep("T0", colnames(lowcarbno))] <- "T0"
newcollowfatno[grep("T18", colnames(lowfatno))] <- "T18"
newcollowfatno[grep("T0", colnames(lowfatno))] <- "T0"
newcollowcarbPA[grep("T18", colnames(lowcarbPA))] <- "T18"
newcollowcarbPA[grep("T0", colnames(lowcarbPA))] <- "T0"
newcollowfatPA[grep("T18", colnames(lowfatPA))] <- "T18"
newcollowfatPA[grep("T0", colnames(lowfatPA))] <- "T0"

```

DmpFinder

```

t18dmplowcarbno <- dmpFinder(lowcarbno, newcollowcarbno, type = "categorical")
t18dmplowcarbPA <- dmpFinder(lowcarbPA, newcollowcarbPA, type = "categorical")
t18dmplowfatno <- dmpFinder(lowfatno, newcollowfatno, type = "categorical")
t18dmplowfatPA <- dmpFinder(lowfatPA, newcollowfatPA, type = "categorical")

```

write and inspect data

```

write.table(t18dmplowfatPA, file = "adjt0_t18dmplowfatPA.tsv", sep = "\t", quote = FALSE)
write.table(t18dmplowfatno, file = "adjt0_t18dmplowfatno.tsv", sep = "\t", quote = FALSE)
write.table(t18dmplowcarbPA, file = "adjt0_t18dmplowcarbPA.tsv", sep = "\t", quote = FALSE)
write.table(t18dmplowcarbno, file = "adjt0_t18dmplowcarbno.tsv", sep = "\t", quote = FALSE)

```

## Compute correlations of betas to phenotype information

```

colstocomp<-c(31,33,35,40,43,44,46,47,49,50,52,56,57,58,94,96,98,104,106,
108,174,176,178,194,196,198,214,216,218)
colsT0<-c(31,40,44,47,50,94,104,174,194,214)
colsT18<-c(33,43,46,49,52,96,106,176,196,216)
colsdel<-c(35,56,57,58,98,108,178,198,218,284)

colnames(phenotypes)[colsT0]

## [1] "wc0"          "BMI_basline"   "vat_0"        "dsc_0"
## [5] "ssc_0"         "GLU_F_o_C"     "HOMAO"       "CRPHS_S_0"
## [9] "Leptin_0"      "Adiponectin_0"

colnames(phenotypes)[colsT18]

## [1] "wc18"          "BMI_T18"       "vat_18"       "dsc_18"
## [5] "ssc_18"         "GLU_F_18_C"    "HOMA18"      "CRPHS_S_18"
## [9] "Leptin_18"      "Adiponectin_18"

colnames(phenotypes)[colsdel]

## [1] "wc_del18"      "vat_del18"     "dsc_del18"    "ssc_del18"
## [5] "GLU_del18"     "HOMA_del18"   "CRP_del18"    "leptin_del18"
## [9] "adiponectin_del18" "WeightLoss.."

```

## Compute and combine and put out correlations

T0

```

for (j in colsT0) {
  pearson<-apply(adj.betas[,grep("T0",colnames(adj.betas))],1,cor_fun,
  as.vector(phenotypes[colnames(adj.betas)[grep("T0",colnames(adj.betas))],j]),
  method="pearson")
  spearman<-apply(adj.betas[,grep("T0",colnames(adj.betas))],1,cor_fun,
  as.vector(phenotypes[colnames(adj.betas)[grep("T0",colnames(adj.betas))],j]),
  method="spearman")
  pearsonad<-p.adjust(pearson[2,])
  spearmanad<-p.adjust(spearman[2,])
  rownames(pearson)<-c("pearson_corr","pearson_pvalue")
  rownames(spearman)<-c("spearman_corr","spearman_pvalue")
  sqrts<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
  sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
  colnames(sqrts)[7]<-"combined_p"
  colnames(sqrts)[8]<-"combined_p_adj"
  colnames(sqrts)[9]<-"combined_corr"
  write.table(sqrts[order(sqrts[,7]),],file=paste(colnames(phenotypes)[j],"adj_T0.tsv"),
  sep="\t",quote=FALSE)
}

```

```

T0matrix<-adj.betas[,grep("T0",colnames(adj.betas))]
T18matrix<-adj.betas[,grep("T18",colnames(adj.betas))]
wcvec0<-as.vector(phenotypes[colnames(adj.betas)[grep("T18",colnames(adj.betas))],31])
wcvec18<-as.vector(phenotypes[colnames(adj.betas)[grep("T18",colnames(adj.betas))],33])
pearson<-apply(cbind(wc0matrix,wc18matrix),1,cor_fun,as.vector(c(wcvec0,wcvec18)),

```

```

method="pearson")
spearman<-apply(cbind(wc0matrix,wc18matrix),1,cor_fun,as.vector(c(wcvec0,wcvec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtts<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtts)[7]<-"combined_p"
colnames(sqrtts)[8]<-"combined_p_adj"
colnames(sqrtts)[9]<-"combined_corr"
write.table(sqrtts[order(sqrtts[,7]),],file=paste("WC","adj_T0andT18.tsv",sep=""),sep="\t",
quote=FALSE)

bmivec0<-as.vector(phenotypes[colnames(adj.betas)][grep("T0",colnames(adj.betas))],40])
bmivec18<-as.vector(phenotypes[colnames(adj.betas)][grep("T18",colnames(adj.betas))],43))
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(bmivec0,bmivec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(bmivec0,bmivec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtts<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtts)[7]<-"combined_p"
colnames(sqrtts)[8]<-"combined_p_adj"
colnames(sqrtts)[9]<-"combined_corr"
write.table(sqrtts[order(sqrtts[,7]),],file=paste("BMI","adj_T0andT18.tsv",sep=""),sep="\t",
quote=FALSE)

vec0<-as.vector(phenotypes[colnames(adj.betas)][grep("T0",colnames(adj.betas))],44)
vec18<-as.vector(phenotypes[colnames(adj.betas)][grep("T18",colnames(adj.betas))],46))
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtts<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtts)[7]<-"combined_p"
colnames(sqrtts)[8]<-"combined_p_adj"
colnames(sqrtts)[9]<-"combined_corr"
write.table(sqrtts[order(sqrtts[,7]),],file=paste("vat","adj_T0andT18.tsv",sep=""),sep="\t",
quote=FALSE)

vec0<-as.vector(phenotypes[colnames(adj.betas)][grep("T0",colnames(adj.betas))],47)
vec18<-as.vector(phenotypes[colnames(adj.betas)][grep("T18",colnames(adj.betas))],49))

```

```

pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtst<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtst)[7]<-"combined_p"
colnames(sqrtst)[8]<-"combined_p_adj"
colnames(sqrtst)[9]<-"combined_corr"
write.table(sqrtst[order(sqrtst[,7]),],file=paste("dsc","adj_T0andT18.tsv",sep=""),sep="\t",
quote=FALSE)

vec0<-as.vector(phenotypes[,colnames(adj.betas)][grep("T0",colnames(adj.betas))],50)
vec18<-as.vector(phenotypes[,colnames(adj.betas)][grep("T18",colnames(adj.betas))],52)
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtst<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtst)[7]<-"combined_p"
colnames(sqrtst)[8]<-"combined_p_adj"
colnames(sqrtst)[9]<-"combined_corr"
write.table(sqrtst[order(sqrtst[,7]),],file=paste("ssc","adj_T0andT18.tsv",sep=""),sep="\t",
quote=FALSE)

vec0<-as.vector(phenotypes[,colnames(adj.betas)][grep("T0",colnames(adj.betas))],94)
vec18<-as.vector(phenotypes[,colnames(adj.betas)][grep("T18",colnames(adj.betas))],96)
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtst<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtst)[7]<-"combined_p"
colnames(sqrtst)[8]<-"combined_p_adj"
colnames(sqrtst)[9]<-"combined_corr"
write.table(sqrtst[order(sqrtst[,7]),],file=paste("GLU","adj_T0andT18.tsv"),sep="\t",
quote=FALSE)

vec0<-as.vector(phenotypes[,colnames(adj.betas)][grep("T0",colnames(adj.betas))],104)

```

```

vec18<-as.vector(phenotypes[,colnames(adj.betas)][grep("T18",colnames(adj.betas))],106)
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtst<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtst)[7]<-"combined_p"
colnames(sqrtst)[8]<-"combined_p_adj"
colnames(sqrtst)[9]<-"combined_corr"
write.table(sqrtst[order(sqrtst[,7]),],file=paste("HOMA","adj_T0andT18.tsv"),sep="\t",
quote=FALSE)

vec0<-as.vector(phenotypes[,colnames(adj.betas)][grep("T0",colnames(adj.betas))],174)
vec18<-as.vector(phenotypes[,colnames(adj.betas)][grep("T18",colnames(adj.betas))],176)
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtst<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtst)[7]<-"combined_p"
colnames(sqrtst)[8]<-"combined_p_adj"
colnames(sqrtst)[9]<-"combined_corr"
write.table(sqrtst[order(sqrtst[,7]),],file=paste("CRPHS","adj_T0andT18.tsv"),sep="\t",
quote=FALSE)

vec0<-as.vector(phenotypes[,colnames(adj.betas)][grep("T0",colnames(adj.betas))],194)
vec18<-as.vector(phenotypes[,colnames(adj.betas)][grep("T18",colnames(adj.betas))],196)
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrtst<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrtst)[7]<-"combined_p"
colnames(sqrtst)[8]<-"combined_p_adj"
colnames(sqrtst)[9]<-"combined_corr"
write.table(sqrtst[order(sqrtst[,7]),],file=paste("Leptin","adj_T0andT18.tsv"),sep="\t",
quote=FALSE)

```

```

vec0<-as.vector(phenotypes[,colnames(adj.betas)[grep("T0", colnames(adj.betas))],214])
vec18<-as.vector(phenotypes[,colnames(adj.betas)[grep("T18", colnames(adj.betas))],216])
pearson<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="pearson")
spearman<-apply(cbind(T0matrix,T18matrix),1,cor_fun,as.vector(c(vec0,vec18)),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrts<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrts)[7]<-"combined_p"
colnames(sqrts)[8]<-"combined_p_adj"
colnames(sqrts)[9]<-"combined_corr"
write.table(sqrts[order(sqrts[,7]),],file=paste("Adiponectin","adj_T0andT18.tsv"),sep="\t",
quote=FALSE)

#c(33,43,46,49,52,96,106,176,196,216)

```

T18

```

for (j in colsT18) {
pearson<-apply(adj.betas[,grep("T18", colnames(adj.betas))],1,cor_fun,
as.vector(phenotypes[,colnames(adj.betas)[grep("T18", colnames(adj.betas))],j]),
method="pearson")
spearman<-apply(adj.betas[,grep("T18", colnames(adj.betas))],1,cor_fun,
as.vector(phenotypes[,colnames(adj.betas)[grep("T18", colnames(adj.betas))],j]),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")
rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrts<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrts)[7]<-"combined_p"
colnames(sqrts)[8]<-"combined_p_adj"
colnames(sqrts)[9]<-"combined_corr"
write.table(sqrts[order(sqrts[,7]),],file=paste(colnames(phenotypes)[j],"adj_T18.tsv"),
sep="\t",
quote=FALSE)
}

```

Delta T0 T18

```

for (j in colsdel) {
pearson<-apply(adj.betas[,grep("T0", colnames(adj.betas))],1,cor_fun,
as.vector(phenotypes[,colnames(adj.betas)[grep("T0", colnames(adj.betas))],j]),
method="pearson")
spearman<-apply(adj.betas[,grep("T0", colnames(adj.betas))],1,cor_fun,
as.vector(phenotypes[,colnames(adj.betas)[grep("T0", colnames(adj.betas))],j]),
method="spearman")
pearsonad<-p.adjust(pearson[2,])
spearmanad<-p.adjust(spearman[2,])
rownames(pearson)<-c("pearson_corr","pearson_pvalue")

```

```

rownames(spearman)<-c("spearman_corr","spearman_pvalue")
sqrts<-t(rbind(pearson,pearsonad,spearman,spearmanad,sqrt(pearson[2,]*spearman[2,]),
sqrt(pearsonad*spearmanad),(pearson[1,]+spearman[1,])/2))
colnames(sqrts)[7]<-"combined_p"
colnames(sqrts)[8]<-"combined_p_adj"
colnames(sqrts)[9]<-"combined_corr"
write.table(sqrts[order(sqrts[,7]),],file=paste(colnames(phenotypes)[j],"T0_all.tsv",
sep=""),sep="\t",quote=FALSE)
}

```

GO term enrichment computations

```

Met005<-read.table("DMR005_cg.txt",header=F)
adjlt0005<-read.table("negcorrelator_cg_005.txt",header=F)
adjbt0005<-read.table("poscorrelator_cg_005.txt",header=F)
KEGGDMR005<-gometh(sig.cpg=Met005[,1],collection="KEGG",array.type="EPIC",
prior.prob=TRUE)
KEGGadj005lt0<-gometh(sig.cpg=adjlt0005[,1],collection="KEGG",array.type="EPIC",
prior.prob=TRUE)
goDMR005<-gometh(sig.cpg=Met005[,1],collection="GO",array.type="EPIC",
prior.prob=TRUE)
goadj005lt0<-gometh(sig.cpg=adjlt0005[,1],collection="GO",array.type="EPIC",
prior.prob=TRUE)
goadj005bt0<-gometh(sig.cpg=adjbt0005[,1],collection="GO",array.type="EPIC",
prior.prob=TRUE)
KEGGadj005bt0<-gometh(sig.cpg=adjbt0005[,1],collection="KEGG",array.type="EPIC",
prior.prob=TRUE)

write.table(KEGGDMR005[KEGGDMR005$FDR<0.05,],file="DMR_p005_KEGG.tsv",sep="\t",
quote=FALSE)
write.table(goadj005lt0[goadj005lt0$FDR<0.05,],file="ADJ_p005_lt0_GO.tsv",sep="\t",
quote=FALSE)
write.table(KEGGadj005lt0[KEGGDMR005lt0$FDR<0.05,],file="ADJ_p005_lt0_KEGG.tsv",sep="\t",
quote=FALSE)
write.table(goDMR005[goDMR005$FDR<0.05,],file="DMR_p005_GO.tsv",sep="\t",
quote=FALSE)

```