```
rm(list=ls()) #empty the workspace
library(tidyverse)
library(vegan)
library(phyloseq)

#Read file as dataframe
Peptides <- as.data.frame(
  read.delim2("Peptides_minCont_mint14.txt",head=TRUE,sep="\t")
  )

#QC normalisation output: Peptides_norm
{
#make a list of normalization ratio using an imported function

QC_normalise_peptides <- function(x) {
  x <- 1+(x/10)
  y <- ((-0.0048*x^2)+0.0216*x+1.011)
  return(y)
}

Peptides_norm <- Peptides
for (i in seq(1,60)){
  Peptides_norm [,i] <- Peptides_norm[,i]*QC_normalise_peptides(i)
}
}

#ADH normalisation
rownames(Peptides_norm) <- Peptides_norm$T..id


#getting ratio of deviation form average
ADHROW <- as.numeric(Peptides_norm[1987,1:60])
SpikeRowMean <- mean(ADHROW)
SpikeRatio <- SpikeRowMean/as.numeric(Peptides_norm[1987,1:60])

#Multiplying column by ratio
Peptides_norm_SPIKE <- as.data.frame(Peptides_norm[,1]*SpikeRatio[[1]])

for (i in seq(2,60)) {
  Peptides_norm_SPIKE <-
    cbind(Peptides_norm_SPIKE,Peptides_norm[,i]*SpikeRatio[[i]])
  print(SpikeRatio[[i]])
}

#Fixing names
colnames(Peptides_norm_SPIKE) <- sub("LFQ.intensity.20210917_TTP_P_","",
                                     colnames(Peptides_norm[,1:60]))
colnames(Peptides_norm_SPIKE) <- sub("_1_.....","",
                                     colnames(Peptides_norm_SPIKE[,1:60]))
rownames(Peptides_norm_SPIKE) <- Peptides_norm$T..id
```

```
#adding metadata after normalisation
Peptides_norm_SPIKE <- cbind(Peptides_norm_SPIKE, Peptides_norm[,61:78])

#Peptides_norm_SPIKE_num <- apply(Peptides_norm_SPIKE[,1:60], 2, as.numeric)
#Peptides_norm_SPIKE_num <- cbind(Peptides_norm_SPIKE_num, Peptides_norm[,61:78])

#Transposing Data.frame
T_Peptides_norm <- t(Peptides_norm_SPIKE)
#clean names
{
#rownames(T_Peptides_norm) <-
  #sub("LFQ.intensity.20210917_TTP_P_","", rownames(T_Peptides_norm))
#rownames(T_Peptides_norm) <-
  #sub("_..._1_.....","", rownames(T_Peptides_norm))

colnames(T_Peptides_norm) <- T_Peptides_norm["T..Proteins",]
}
#Adding CVD column
{

CVD <- c("before","before","after","after","control","control",
         "before","before","after","after","control","control",
         "before","before","after","after","control","control",
         "before","before","after","after","control","control",

         "before","before","after","after","control","control",
         "before","before","after","after","control","control",

         "before","before","after","after","control","control",
         "before","before","after","after","control","control",

         "before","before","after","after","control","control",

         "before","before","after","after","control","control",
         "NA","NA","NA","NA","NA","NA","NA",
         "NA","NA","NA","NA","NA","NA","NA",
         "NA","NA","NA","NA")


T_Peptides_norm_CVD <- cbind(T_Peptides_norm,
CVD=CVD)
}
T_Peptides_norm_CVD_incl_meta <- T_Peptides_norm_CVD
T_Peptides_norm_CVD <- T_Peptides_norm_CVD[1:60,]

T_Peptides_norm_CVD <- as.data.frame(T_Peptides_norm_CVD)

Peptide_file <- cbind.data.frame(sample = rownames(T_Peptides_norm_CVD),
                                 T_Peptides_norm_CVD)
```

```
write_tsv(Peptide_file, "T_Peptides_norm_CVD.tsv")
#write.table(Peptides_norm_SPIKE_num,"Peptides_norm_SPIKE_num.tsv", sep= '\t',
row.names=F)

#CLR transformation
#euclidian distance

df <- T_Peptides_norm_CVD[,c(1:4696)]
df <- apply(df, 2, as.numeric)
#Replacing all values +1 for log transformation
df <- df+1

clrdat <- as.data.frame(compositions::clr((t(df))))
clrdat1 <- otu_table((clrdat),taxa_are_rows= T)
dist <- philentropy::distance(t(clrdat), method = "euclidean")
#check of de distance matrix gebaseerd is op je subjecten; niet je variabelen

T_Peptides_norm_CVD <- as.data.frame(T_Peptides_norm_CVD)
class(T_Peptides_norm_CVD)
adonis2(dist ~ CVD, data = T_Peptides_norm_CVD,
        permutations = 9999,
        method = "bray",
        na.rm = T)
```