> rm(list = ls())

> options(warn=-1)

>

> #### PRELIMINARIES #############################################################################################

>

> #\*Uploads the needed libraries --------------------------------------------------------------------------------

>

> require(ggplot2)

>

> require(data.table)

>

> require(plotly)

>

> require(DT)

>

> require(R2HTML)

>

> require(stringr)

>

> #\*\* Sets the number of significant digits for the output --------------------------

> sig\_dig = 4

>

> #\*\* Sets the working directory ---------------------------------------------------------------------------------

>

> #Gets the default wd

> default\_wd <- getwd()

>

> setwd("C:/Users/Furse/Documents/OneDrive/Work/Cambridge/LTA -- Ozanne mouse model of GDM/PWD/data/Data sheets/PWD/data/")# <--- insert here the path to the working (output) directory

>

> new\_wd <- getwd()

>

> #Sets the input directory

> inputdir <-"C:/Users/Furse/Documents/OneDrive/Work/Cambridge/LTA -- Ozanne mouse model of GDM/PWD/data/Data sheets/PWD/"# <--- insert here the path to the input directory

>

>

> #### DEFINES FUNCTIONS #############################################################################################

>

> count\_zeroes <- function(x){length(which(x==0))}

> #considered\_mode <- "\\+ve"

> considered\_mode <- "-ve"

> considered\_generation <- "PWD"

> considered\_model\_1 <-"lean"

> considered\_model\_2 <- "obese"

>

>

> #### DATA UPLOAD ###############################################################################################################

>

>

>

>

> # - Uploads the .csv files containing the information related to the F1A, PW, -ve datasets.

> # - Separates the metadata from the main data

> # - Picks only the considered\_model\_1 and considered\_model\_2 sets

>

> files\_names\_originals <- list.files(inputdir)

> files\_names\_originals <- files\_names\_originals[which(str\_length(files\_names\_originals)==17)]

>

> modes <- unique(substr(files\_names\_originals, start=1, stop=3))

> tissues <- unique(substr(files\_names\_originals, start=6, stop=8))

> generations <- unique(substr(files\_names\_originals, start=11, stop=13))

>

> # PRODUCES TISSUE-SPECIFIC MATRICES

> # For each tissue, produces two matrices (one for each considered\_model)

> # in which the columns are the samples and the rows are the lipids

>

> for(j in 1: length(tissues)){

+

+ tissue <- tissues[j]

+ aa <- files\_names\_originals[grep(files\_names\_originals, pattern=considered\_mode)]

+ aa <- aa[grep(aa, pattern=considered\_generation)]

+ aa<- aa[grep(aa, pattern=tissue)]

+

+ if(length(aa)!=0){

+

+ bb<- read.csv(paste0(inputdir, aa), stringsAsFactors = F)

+ assign(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_metadata"), bb)

+

+ cc<- read.csv(paste0(inputdir, aa), stringsAsFactors = F , skip=10 )

+ cc <- cc[!is.na(cc$m.z),]

+ assign(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4)), cc)

+

+ cc\_1 <- cc[,grep(as.vector(bb[4, ]),pattern=considered\_model\_1)]

+ cc\_2 <- cc[,grep(as.vector(bb[4, ]),pattern=considered\_model\_2)]

+ rownames(cc\_1) <- cc$Lipid.variable

+ rownames(cc\_2) <- cc$Lipid.variable

+

+ cc\_1\_zeroes <- apply(cc\_1[, c(3:ncol(cc\_1))], 1, count\_zeroes)

+ cc\_2\_zeroes <- apply(cc\_2[, c(3:ncol(cc\_2))], 1, count\_zeroes)

+

+ cc\_1\_nozeroes <- cc\_1[-which(cc\_1\_zeroes > ncol(cc\_1)\*20/100) ,]

+ cc\_2\_nozeroes <- cc\_2[-which(cc\_2\_zeroes > ncol(cc\_2)\*20/100) ,]

+

+ assign(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1), cc\_1)

+ assign(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_2), cc\_2)

+

+ assign(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_nozeroes"), cc\_1\_nozeroes)

+ assign(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_2, "\_nozeroes"), cc\_2\_nozeroes)

+ }

+ }

>

> #### A LIPIDS ###############################################################################################################

>

> # Finds the A-type lipids for both the considered models --------------------------------------------------------------------------------------------------

>

> # \*\* Considered\_model\_1 --------------------------------------------------------------------------------------

>

> # Finds the A-lipids

>

> # Creates a list (called all\_tissues) in which each element is

> # a list of lipids coming from the row names of

> # each Tissue-specific Matrix

> all\_tissues <-list()

> for(k in 1:length(tissues)){

+ tissue <- tissues[k]

+ aa <- files\_names\_originals[grep(files\_names\_originals, pattern=considered\_mode)]

+ aa <- aa[grep(aa, pattern=considered\_generation)]

+ aa<- aa[grep(aa, pattern=tissue)]

+

+ if(length(aa)!=0){

+ print(tissue)

+ yy <- rownames(get(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_nozeroes") ) )

+ assign(paste0("xx\_",k), yy)

+ all\_tissues[[k]] <- yy

+

+ }

+

+ }

[1] "ADI"

[1] "BRA"

[1] "HEA"

[1] "LIV"

[1] "SER"

[1] "VAS"

> names(all\_tissues) <- tissues

>

> if(length(which(all\_tissues=="NULL")) !=0){

+ all\_tissues<-all\_tissues[-which(all\_tissues=="NULL")]

+ }

>

> # intersects all the elements of the all\_tissues list to find the A-lipids

> gg\_1<-Reduce(intersect, all\_tissues)

> assign(paste0 ("A\_Lipids\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1), gg\_1)

> write.csv(gg\_1, file=paste0 ("A\_Lipids\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, ".csv"))

>

> # Counts the A-lipids in each classes

> classes <- unique(substr(gg\_1, start=1, stop =5))

> classes\_counts\_model\_1 <- matrix(ncol=1, nrow=length(classes))

> rownames(classes\_counts\_model\_1) <- classes

> colnames(classes\_counts\_model\_1) <- paste0("A\_lipids\_",considered\_model\_1)

> for(i in 1:length(classes)){

+

+ classes\_counts\_model\_1[i] <- length(grep(gg\_1, pattern=paste0("^",classes[i])))

+ }

>

> # \*\* Considered\_model\_2 --------------------------------------------------------------------------------------

>

> # Finds the A-lipids

>

> # Creates a list (called all\_tissues) in which each element is

> # a list of lipids coming from the row names of

> # each Tissue-specific Matrix

> all\_tissues <-list()

> for(k in 1:length(tissues)){

+ tissue <- tissues[k]

+ aa <- files\_names\_originals[grep(files\_names\_originals, pattern=considered\_mode)]

+ aa <- aa[grep(aa, pattern=considered\_generation)]

+ aa<- aa[grep(aa, pattern=tissue)]

+

+ if(length(aa)!=0){

+ print(tissue)

+ yy <- rownames(get(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_2, "\_nozeroes") ) )

+ assign(paste0("xx\_",k), yy)

+ all\_tissues[[k]] <- yy

+

+ }

+

+ }

[1] "ADI"

[1] "BRA"

[1] "HEA"

[1] "LIV"

[1] "SER"

[1] "VAS"

> names(all\_tissues) <- tissues

> if(length(which(all\_tissues=="NULL")) !=0){

+ all\_tissues<-all\_tissues[-which(all\_tissues=="NULL")]

+ }

>

> # intersects all the elements of the all\_tissues list to find the A-lipids

> gg\_2<-Reduce(intersect, all\_tissues)

> assign(paste0 ("A\_Lipids\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_2), gg\_2)

> write.csv(gg\_2, file=paste0 ("A\_Lipids\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_2, ".csv"))

>

>

> # Counts the A-lipids in each class

> classes <- unique(substr(gg\_2, start=1, stop =5))

> classes\_counts\_model\_2 <- matrix(ncol=1, nrow=length(classes))

> rownames(classes\_counts\_model\_2) <- classes

> colnames(classes\_counts\_model\_2) <- paste0("A\_lipids\_",considered\_model\_2)

> for(i in 1:length(classes)){

+

+ classes\_counts\_model\_2[i] <- length(grep(gg\_2, pattern= paste0("^",classes[i])))

+ }

>

> # Creates the A\_lipids\_classes\_counts\_tot dataframe, where the vectors classes\_counts\_model\_1 and classes\_counts\_model\_2 are merged

> classes\_counts\_model\_1 <- as.data.frame(classes\_counts\_model\_1)

> classes\_counts\_model\_2 <- as.data.frame(classes\_counts\_model\_2)

> A\_lipids\_classes\_counts\_tot <- merge(classes\_counts\_model\_1, classes\_counts\_model\_2, by.x="row.names", by.y="row.names", all=T)

> A\_lipids\_classes\_counts\_tot[is.na(A\_lipids\_classes\_counts\_tot)]<-0

> rownames(A\_lipids\_classes\_counts\_tot) <- A\_lipids\_classes\_counts\_tot[,1]

> A\_lipids\_classes\_counts\_tot <- A\_lipids\_classes\_counts\_tot[,-1]

> #

> All\_glicerids <- A\_lipids\_classes\_counts\_tot[grep(rownames(A\_lipids\_classes\_counts\_tot), pattern="DGX|MGX|TGX"),]

> A\_lipids\_classes\_counts\_tot[nrow(A\_lipids\_classes\_counts\_tot)+1,] <- colSums(All\_glicerids)

> rownames(A\_lipids\_classes\_counts\_tot)[nrow(A\_lipids\_classes\_counts\_tot)] <- "Glyc"

>

> assign(paste0("A\_lipids\_classes\_counts\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), A\_lipids\_classes\_counts\_tot)

> write.csv(A\_lipids\_classes\_counts\_tot, file=paste0("A\_lipids\_classes\_counts\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"))

>

> # Computes the Jaccard distances between models --------------------------------------------------------------------------------------------------

>

> # This part is not needed for the computation Jaccard distances but only for printing the A\_lipids\_all\_matrix - START

> # Merges the vectors containing the A lipids "substituted" for each model into the A\_lipids\_matrix

> ss <- union(gg\_1, gg\_2)

> ss<- sort(ss)

> #

> A\_lipids\_matrix<- cbind(ss, rep(0, length(ss)), rep(0, length(ss)) )

> A\_lipids\_matrix[,2][which(A\_lipids\_matrix[,1] %in% gg\_1)] <- A\_lipids\_matrix[,1][which(A\_lipids\_matrix[,1] %in% gg\_1)]

> A\_lipids\_matrix[,3][which(A\_lipids\_matrix[,1] %in% gg\_2)] <- A\_lipids\_matrix[,1][which(A\_lipids\_matrix[,1] %in% gg\_2)]

> #

> colnames(A\_lipids\_matrix) <- c("rownames",considered\_model\_1, considered\_model\_2)

> rownames(A\_lipids\_matrix) <- A\_lipids\_matrix[,1]

> A\_lipids\_matrix <- A\_lipids\_matrix[,-1]

> assign(paste0 ("A\_Lipids\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), A\_lipids\_matrix)

> write.csv(file= paste0 ("A\_Lipids\_all\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), A\_lipids\_matrix)

> # This part is not needed for the computation Jaccard distances but only for printing the A\_lipids\_all\_matrix - END

>

>

> # Unifies the classes MG, DG and TG under the class Glyc, by changing the row names of the A-lipids

> gg\_1\_substituted <- gsub("MGXX|DGXX|TGXX", "Glyc", gg\_1)

> gg\_2\_substituted <- gsub("MGXX|DGXX|TGXX", "Glyc", gg\_2)

>

> # Merges the vectors containing the A lipids "substituted" for each model into the A\_lipids\_matrix

> ss <- union(gg\_1\_substituted, gg\_2\_substituted)

> ss<- sort(ss)

> #

> A\_lipids\_matrix<- cbind(ss, rep(0, length(ss)), rep(0, length(ss)) )

> A\_lipids\_matrix[,2][which(A\_lipids\_matrix[,1] %in% gg\_1\_substituted)] <- A\_lipids\_matrix[,1][which(A\_lipids\_matrix[,1] %in% gg\_1\_substituted)]

> A\_lipids\_matrix[,3][which(A\_lipids\_matrix[,1] %in% gg\_2\_substituted)] <- A\_lipids\_matrix[,1][which(A\_lipids\_matrix[,1] %in% gg\_2\_substituted)]

> #

> colnames(A\_lipids\_matrix) <- c("rownames",considered\_model\_1, considered\_model\_2)

> rownames(A\_lipids\_matrix) <- A\_lipids\_matrix[,1]

> A\_lipids\_matrix <- A\_lipids\_matrix[,-1]

> assign(paste0 ("A\_Lipids\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), A\_lipids\_matrix)

> write.csv(file= paste0 ("A\_Lipids\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), A\_lipids\_matrix)

>

> # Computes the Jaccard distances

> require(jaccard)

>

> if(considered\_mode =="\\+ve"){

+ classes <- rownames(A\_lipids\_classes\_counts\_tot)[-grep(rownames(A\_lipids\_classes\_counts\_tot), pattern="DG|TGX|MGX")]

+ classes <- c(classes, "Glyc")

+ Jaccard\_distances <- matrix(ncol=2, nrow = length(classes))

+ colnames(Jaccard\_distances) <- c("Distance", "Pvalue")

+ rownames(Jaccard\_distances) <- classes

+ } else {

+ classes <- rownames(A\_lipids\_classes\_counts\_tot)

+ Jaccard\_distances <- matrix(ncol=2, nrow = length(classes))

+ colnames(Jaccard\_distances) <- c("Distance", "Pvalue")

+ rownames(Jaccard\_distances) <- classes

+ }

>

> Global\_jaccard\_matrix <- A\_lipids\_matrix

> Global\_jaccard\_matrix[which(Global\_jaccard\_matrix!=0)] = 1

>

> for(i in 1:length(classes)){

+ zz<- as.matrix(Global\_jaccard\_matrix[grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]),])

+ if(length(grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]))==1 ){

+ uu<- jaccard(as.numeric(zz[1]), as.numeric(zz[2]))

+ vv <-jaccard.test(as.numeric(zz[1]), as.numeric(zz[2]), method = "exact")

+ Jaccard\_distances[i,c(1,2)] <- c(uu,vv$pvalue)

+ }

+ else if (length(grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]))==0) {

+ Jaccard\_distances[i,c(1,2)] <- c("NA","NA")

+ }

+ else{

+ uu<- jaccard(as.numeric(zz[,1]), as.numeric(zz[,2]))

+ vv <-jaccard.test(as.numeric(zz[,1]), as.numeric(zz[,2]), method = "exact")

+ Jaccard\_distances[i,c(1,2)] <- c(uu,vv$pvalue)

+ }

+ #print(zz)

+ }

>

> assign(paste0 ("Jaccard\_distances\_A\_Lipids\_", "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), Jaccard\_distances)

> write.csv(file=paste0 ("Jaccard\_distances\_A\_Lipids\_", "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), Jaccard\_distances)

>

> #### U LIPIDS ###############################################################################################################

>

> # Finds the U-lipids for each tissue

>

> # \*\* Considered\_model\_1 --------------------------------------------------------------------------------------

> # Creates a list (called all\_tissues\_1) in which each element is

> # a list of lipids coming from the row names of

> # each Tissue-specific and model-specific Matrix

>

> all\_tissues\_1 <-list()

> for(k in 1:length(tissues)){

+ tissue <- tissues[k]

+ aa <- files\_names\_originals[grep(files\_names\_originals, pattern=considered\_mode)]

+ aa <- aa[grep(aa, pattern=considered\_generation)]

+ aa <- aa[grep(aa, pattern=tissue)]

+

+ if(length(aa)!=0){

+ yy <- rownames(get(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_nozeroes") ) )

+ assign(paste0("xx\_",k), yy)

+ all\_tissues\_1[[k]] <- yy

+

+ }

+

+ }

> names(all\_tissues\_1) <- tissues

> if(length(which(all\_tissues\_1=="NULL")) !=0){

+ all\_tissues<-all\_tissues\_1[-which(all\_tissues\_1=="NULL")]

+ }

>

> # \*\* Considered\_model\_2 --------------------------------------------------------------------------------------

> # Creates a list (called all\_tissues\_2) in which each element is

> # a list of lipids coming from the row names of

> # each Tissue-specific and model-specific Matrix

>

> all\_tissues\_2 <-list()

> for(k in 1:length(tissues)){

+ tissue <- tissues[k]

+ aa <- files\_names\_originals[grep(files\_names\_originals, pattern=considered\_mode)]

+ aa <- aa[grep(aa, pattern=considered\_generation)]

+ aa <- aa[grep(aa, pattern=tissue)]

+

+ if(length(aa)!=0){

+ yy <- rownames(get(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_2, "\_nozeroes") ) )

+ assign(paste0("xx\_",k), yy)

+ all\_tissues\_2[[k]] <- yy

+

+ }

+

+ }

> names(all\_tissues\_2) <- tissues

> if(length(which(all\_tissues\_2=="NULL")) !=0){

+ all\_tissues<-all\_tissues\_2[-which(all\_tissues\_2=="NULL")]

+ }

>

> # \*\* Tissue-specific U-lipids --------------------------------------------------------------------------------------

>

> # \*\*\*\* Produces the Tissue-specific U-lipids matrices --------------------------------------------------------------------------------------

>

> for(o in 1:length(names(all\_tissues\_1))){

+ Tissue <- names(all\_tissues)[o]

+

+ considered\_tissue\_1 <- all\_tissues\_1[which(names(all\_tissues\_1)==Tissue)]

+ other\_tissues\_1 <- all\_tissues\_1[-which(names(all\_tissues\_1)==Tissue)]

+

+ ss\_1 <- Reduce(union, other\_tissues\_1)

+ tt\_1 <- considered\_tissue\_1[[1]]

+ uu\_1<- setdiff(tt\_1,ss\_1)

+

+ considered\_tissue\_2 <- all\_tissues\_2[which(names(all\_tissues\_2)==Tissue)]

+ other\_tissues\_2 <- all\_tissues\_2[-which(names(all\_tissues\_2)==Tissue)]

+

+ ss\_2 <- Reduce(union, other\_tissues\_2)

+ tt\_2 <- considered\_tissue\_2[[1]]

+ uu\_2<- setdiff(tt\_2,ss\_2)

+

+ vv <- union(uu\_1, uu\_2)

+ vv<- sort(vv)

+ #

+ U\_lipids\_matrix<- cbind(vv, rep(0, length(vv)), rep(0, length(vv)) )

+ U\_lipids\_matrix[,2][which(U\_lipids\_matrix[,1] %in% uu\_1)] <- U\_lipids\_matrix[,1][which(U\_lipids\_matrix[,1] %in% uu\_1)]

+ U\_lipids\_matrix[,3][which(U\_lipids\_matrix[,1] %in% uu\_2)] <- U\_lipids\_matrix[,1][which(U\_lipids\_matrix[,1] %in% uu\_2)]

+ #

+ colnames(U\_lipids\_matrix) <- c("rownames",considered\_model\_1, considered\_model\_2)

+ rownames(U\_lipids\_matrix) <- U\_lipids\_matrix[,1]

+ U\_lipids\_matrix <- U\_lipids\_matrix[,-1]

+ assign(paste0 ("U\_Lipids\_", "all\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), U\_lipids\_matrix)

+ write.csv(file= paste0 ("U\_Lipids\_", "all\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), U\_lipids\_matrix)

+

+ # \*\*\*\* Counts the U-lipids in each class --------------------------------------------------------------------------------------

+

+ classes\_1 <- unique(substr(uu\_1, start=1, stop =5))

+ classes\_counts\_model\_1 <- matrix(ncol=1, nrow=length(classes\_1))

+ rownames(classes\_counts\_model\_1) <- classes\_1

+ colnames(classes\_counts\_model\_1) <- paste0("U\_lipids\_",considered\_model\_1)

+

+ for(i in 1:length(classes\_1)){

+

+ classes\_counts\_model\_1[i] <- length(grep(uu\_1, pattern=classes\_1[i]))

+ }

+

+ classes\_2 <- unique(substr(uu\_2, start=1, stop =5))

+ classes\_counts\_model\_2 <- matrix(ncol=1, nrow=length(classes\_2))

+ rownames(classes\_counts\_model\_2) <- classes\_2

+ colnames(classes\_counts\_model\_2) <- paste0("U\_lipids\_",considered\_model\_2)

+

+ for(i in 1:length(classes\_2)){

+

+ classes\_counts\_model\_2[i] <- length(grep(uu\_2, pattern=classes\_2[i]))

+ }

+

+ # Creates the U\_lipids\_classes\_counts\_tot dataframe, where the vectors classes\_counts\_model\_1 and classes\_counts\_model\_2 are merged

+ classes\_counts\_model\_1 <- as.data.frame(classes\_counts\_model\_1)

+ classes\_counts\_model\_2 <- as.data.frame(classes\_counts\_model\_2)

+ U\_lipids\_classes\_counts\_tot <- merge(classes\_counts\_model\_1, classes\_counts\_model\_2, by.x="row.names", by.y="row.names", all=T)

+ U\_lipids\_classes\_counts\_tot[is.na(U\_lipids\_classes\_counts\_tot)]<-0

+ rownames(U\_lipids\_classes\_counts\_tot) <- U\_lipids\_classes\_counts\_tot[,1]

+ U\_lipids\_classes\_counts\_tot <- U\_lipids\_classes\_counts\_tot[,-1]

+ #

+ All\_glicerids <- U\_lipids\_classes\_counts\_tot[grep(rownames(U\_lipids\_classes\_counts\_tot), pattern="DGX|MGX|TGX"),]

+ U\_lipids\_classes\_counts\_tot[nrow(U\_lipids\_classes\_counts\_tot)+1,] <- colSums(All\_glicerids)

+ rownames(U\_lipids\_classes\_counts\_tot)[nrow(U\_lipids\_classes\_counts\_tot)] <- "Glyc"

+

+ assign(paste0("U\_lipids\_classes\_counts\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), U\_lipids\_classes\_counts\_tot)

+ write.csv(U\_lipids\_classes\_counts\_tot, file=paste0("U\_lipids\_classes\_counts\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"))

+

+

+ # Computes the Jaccard distances between models --------------------------------------------------------------------------------------------------

+

+ # Unifies the classes MG, DG and TG under the class Glyc, by changing the row names of the A-lipids

+ uu\_1\_substituted <- gsub("MGXX|DGXX|TGXX", "Glyc", uu\_1)

+ uu\_2\_substituted <- gsub("MGXX|DGXX|TGXX", "Glyc", uu\_2)

+

+ # Merges the vectors containing the A lipids "substituted" for each model into the A\_lipids\_matrix

+ ss <- union(uu\_1\_substituted, uu\_2\_substituted)

+ ss<- sort(ss)

+ #

+ U\_lipids\_matrix<- cbind(ss, rep(0, length(ss)), rep(0, length(ss)) )

+ U\_lipids\_matrix[,2][which(U\_lipids\_matrix[,1] %in% uu\_1\_substituted)] <- U\_lipids\_matrix[,1][which(U\_lipids\_matrix[,1] %in% uu\_1\_substituted)]

+ U\_lipids\_matrix[,3][which(U\_lipids\_matrix[,1] %in% uu\_2\_substituted)] <- U\_lipids\_matrix[,1][which(U\_lipids\_matrix[,1] %in% uu\_2\_substituted)]

+ #

+ colnames(U\_lipids\_matrix) <- c("rownames",considered\_model\_1, considered\_model\_2)

+ rownames(U\_lipids\_matrix) <- U\_lipids\_matrix[,1]

+ U\_lipids\_matrix <- U\_lipids\_matrix[,-1]

+ assign(paste0 ("U\_Lipids\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), U\_lipids\_matrix)

+ write.csv(file= paste0 ("U\_Lipids\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), U\_lipids\_matrix)

+

+ # Computes the Jaccard distances

+ require(jaccard)

+

+ classes <- rownames(U\_lipids\_classes\_counts\_tot)

+ Jaccard\_distances <- matrix(ncol=2, nrow = length(classes))

+ colnames(Jaccard\_distances) <- c("Distance", "Pvalue")

+ rownames(Jaccard\_distances) <- classes

+

+ Global\_jaccard\_matrix <- U\_lipids\_matrix

+ Global\_jaccard\_matrix[which(Global\_jaccard\_matrix!=0)] = 1

+

+ for(i in 1:length(classes)){

+ zz<- as.matrix(Global\_jaccard\_matrix[grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]),])

+ if(length(grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]))==1 ){

+ uu<- jaccard(as.numeric(zz[1]), as.numeric(zz[2]))

+ vv <-jaccard.test(as.numeric(zz[1]), as.numeric(zz[2]), method = "exact")

+ Jaccard\_distances[i,c(1,2)] <- c(uu,vv$pvalue)

+ }

+ else if (length(grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]))==0) {

+ Jaccard\_distances[i,c(1,2)] <- c("NA","NA")

+ }

+ else{

+ uu<- jaccard(as.numeric(zz[,1]), as.numeric(zz[,2]))

+ vv <-jaccard.test(as.numeric(zz[,1]), as.numeric(zz[,2]), method = "exact")

+ Jaccard\_distances[i,c(1,2)] <- c(uu,vv$pvalue)

+ }

+ #print(zz)

+ }

+

+ assign(paste0 ("Jaccard\_distances\_U\_Lipids\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), Jaccard\_distances)

+ write.csv(file=paste0 ("Jaccard\_distances\_U\_Lipids\_", Tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), Jaccard\_distances)

+

+ }

Error in grep(uu\_1, pattern = classes\_1[i]) : invalid 'pattern' argument

>

> #### B LIPIDS ###############################################################################################################

>

> # \*\* Considered\_model\_1 --------------------------------------------------------------------------------------

> # Creates a list (called all\_tissues\_1) in which each element is

> # a list of lipids coming from the row names of

> # each Tissue-specific and model-specific Matrix

>

> all\_tissues\_1 <-list()

> for(k in 1:length(tissues)){

+ tissue <- tissues[k]

+ aa <- files\_names\_originals[grep(files\_names\_originals, pattern=considered\_mode)]

+ aa <- aa[grep(aa, pattern=considered\_generation)]

+ aa <- aa[grep(aa, pattern=tissue)]

+

+ if(length(aa)!=0){

+ yy <- rownames(get(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_nozeroes") ) )

+ assign(paste0("xx\_",k), yy)

+ all\_tissues\_1[[k]] <- yy

+

+ }

+

+ }

> names(all\_tissues\_1) <- tissues

> if(length(which(all\_tissues\_1=="NULL")) !=0){

+ all\_tissues<-all\_tissues\_1[-which(all\_tissues\_1=="NULL")]

+ }

>

> # \*\* Considered\_model\_2 --------------------------------------------------------------------------------------

> # Creates a list (called all\_tissues\_2) in which each element is

> # a list of lipids coming from the row names of

> # each Tissue-specific and model-specific Matrix

>

> all\_tissues\_2 <-list()

> for(k in 1:length(tissues)){

+ tissue <- tissues[k]

+ aa <- files\_names\_originals[grep(files\_names\_originals, pattern=considered\_mode)]

+ aa <- aa[grep(aa, pattern=considered\_generation)]

+ aa <- aa[grep(aa, pattern=tissue)]

+

+ if(length(aa)!=0){

+ yy <- rownames(get(paste0(tissue, "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_2, "\_nozeroes") ) )

+ assign(paste0("xx\_",k), yy)

+ all\_tissues\_2[[k]] <- yy

+

+ }

+

+ }

> names(all\_tissues\_2) <- tissues

> if(length(which(all\_tissues\_2=="NULL")) !=0){

+ all\_tissues<-all\_tissues\_2[-which(all\_tissues\_2=="NULL")]

+ }

>

> #Creates the IDs of the pairwise comparisons

> possible\_pairs <- combn(names(all\_tissues\_1), 2)

>

> #Compares the two models for each pair of tissues

> for(d in 1:ncol(possible\_pairs)){

+

+ tissues\_to\_compare <- possible\_pairs[,d]

+

+ B\_lipids\_1 <- (intersect( all\_tissues\_1[[tissues\_to\_compare[1]]] , all\_tissues\_1[[tissues\_to\_compare[2]]] ))

+ B\_lipids\_2 <- (intersect( all\_tissues\_2[[tissues\_to\_compare[1]]] , all\_tissues\_2[[tissues\_to\_compare[2]]] ))

+

+ B\_lipids\_tot <- union(B\_lipids\_1, B\_lipids\_2)

+

+ B\_lipids\_matrix<- cbind(B\_lipids\_tot, rep(0, length(B\_lipids\_tot)), rep(0, length(B\_lipids\_tot)) )

+ B\_lipids\_matrix[,2][which(B\_lipids\_matrix[,1] %in% B\_lipids\_1)] <- B\_lipids\_matrix[,1][which(B\_lipids\_matrix[,1] %in% B\_lipids\_1)]

+ B\_lipids\_matrix[,3][which(B\_lipids\_matrix[,1] %in% B\_lipids\_2)] <- B\_lipids\_matrix[,1][which(B\_lipids\_matrix[,1] %in% B\_lipids\_2)]

+ #

+ colnames(B\_lipids\_matrix) <- c("rownames",considered\_model\_1, considered\_model\_2)

+ rownames(B\_lipids\_matrix) <- B\_lipids\_matrix[,1]

+ B\_lipids\_matrix <- B\_lipids\_matrix[,-1]

+ assign(paste0 ("B\_Lipids\_", "all\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), B\_lipids\_matrix)

+ write.csv(file= paste0 ("B\_Lipids\_", "all\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), B\_lipids\_matrix)

+

+

+ # Counts the B-lipids in each classes --------------------------------------------------------------------------------------

+

+ # Considered\_model\_1

+

+ classes <- unique(substr(B\_lipids\_1, start=1, stop =5))

+ classes\_counts\_model\_1 <- matrix(ncol=1, nrow=length(classes))

+ rownames(classes\_counts\_model\_1) <- classes

+ colnames(classes\_counts\_model\_1) <- paste0("B\_lipids\_",considered\_model\_1)

+ for(i in 1:length(classes)){

+

+ classes\_counts\_model\_1[i] <- length(grep(B\_lipids\_1, pattern=paste0("^",classes[i])))

+ }

+

+ # Considered\_model\_2

+

+ classes <- unique(substr(B\_lipids\_2, start=1, stop =5))

+ classes\_counts\_model\_2 <- matrix(ncol=1, nrow=length(classes))

+ rownames(classes\_counts\_model\_2) <- classes

+ colnames(classes\_counts\_model\_2) <- paste0("B\_lipids\_",considered\_model\_2)

+ for(i in 1:length(classes)){

+

+ classes\_counts\_model\_2[i] <- length(grep(B\_lipids\_2, pattern=paste0("^",classes[i])))

+ }

+

+ #Creates the B\_lipids\_classes\_counts\_tot dataframe, where the vectors classes\_counts\_model\_1 and classes\_counts\_model\_2 are merged

+ classes\_counts\_model\_1 <- as.data.frame(classes\_counts\_model\_1)

+ classes\_counts\_model\_2 <- as.data.frame(classes\_counts\_model\_2)

+ B\_lipids\_classes\_counts\_tot <- merge(classes\_counts\_model\_1, classes\_counts\_model\_2, by.x="row.names", by.y="row.names", all=T)

+ B\_lipids\_classes\_counts\_tot[is.na(B\_lipids\_classes\_counts\_tot)]<-0

+ rownames(B\_lipids\_classes\_counts\_tot) <- B\_lipids\_classes\_counts\_tot[,1]

+ B\_lipids\_classes\_counts\_tot <- B\_lipids\_classes\_counts\_tot[,-1]

+ #

+ All\_glicerids <- B\_lipids\_classes\_counts\_tot[grep(rownames(B\_lipids\_classes\_counts\_tot), pattern="DGX|MGX|TGX"),]

+ B\_lipids\_classes\_counts\_tot[nrow(B\_lipids\_classes\_counts\_tot)+1,] <- colSums(All\_glicerids)

+ rownames(B\_lipids\_classes\_counts\_tot)[nrow(B\_lipids\_classes\_counts\_tot)] <- "Glyc"

+

+ assign(paste0("B\_lipids\_classes\_counts\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), B\_lipids\_classes\_counts\_tot)

+ write.csv(B\_lipids\_classes\_counts\_tot, file=paste0("B\_lipids\_classes\_counts\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"))

+

+ # Computes the Jaccard Distances --------------------------------------------------------------------------------------

+

+ # Unifies the classes MG, DG and TG under the class Glyc, by changing the row names of the B-lipids

+ B\_lipids\_1\_substituted <- gsub("MGXX|DGXX|TGXX", "Glyc", B\_lipids\_1)

+ B\_lipids\_2\_substituted <- gsub("MGXX|DGXX|TGXX", "Glyc", B\_lipids\_2)

+

+ # Merges the vectors containing the A lipids "substituted" for each model into the B\_lipids\_matrix

+ vv <- union(B\_lipids\_1\_substituted, B\_lipids\_2\_substituted)

+ vv<- sort(vv)

+ #

+ B\_lipids\_matrix<- cbind(vv, rep(0, length(vv)), rep(0, length(vv)) )

+ B\_lipids\_matrix[,2][which(B\_lipids\_matrix[,1] %in% B\_lipids\_1\_substituted)] <- B\_lipids\_matrix[,1][which(B\_lipids\_matrix[,1] %in% B\_lipids\_1\_substituted)]

+ B\_lipids\_matrix[,3][which(B\_lipids\_matrix[,1] %in% B\_lipids\_2\_substituted)] <- B\_lipids\_matrix[,1][which(B\_lipids\_matrix[,1] %in% B\_lipids\_2\_substituted)]

+ #

+ colnames(B\_lipids\_matrix) <- c("rownames",considered\_model\_1, considered\_model\_2)

+ rownames(B\_lipids\_matrix) <- B\_lipids\_matrix[,1]

+ B\_lipids\_matrix <- B\_lipids\_matrix[,-1]

+ assign(paste0 ("B\_Lipids\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), B\_lipids\_matrix)

+ write.csv(file= paste0 ("B\_Lipids\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_",considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), B\_lipids\_matrix)

+

+ # Computes the Jaccard distances

+ require(jaccard)

+ classes <- rownames(B\_lipids\_classes\_counts\_tot)

+ Jaccard\_distances <- matrix(ncol=2, nrow = length(classes))

+ colnames(Jaccard\_distances) <- c("Distance", "Pvalue")

+ rownames(Jaccard\_distances) <- classes

+

+ Global\_jaccard\_matrix <- B\_lipids\_matrix

+ Global\_jaccard\_matrix[which(Global\_jaccard\_matrix!=0)] = 1

+

+ for(i in 1:length(classes)){

+ zz<- as.matrix(Global\_jaccard\_matrix[grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]),])

+ if(length(grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]))==1 ){

+ uu<- jaccard(as.numeric(zz[1]), as.numeric(zz[2]))

+ vv <-jaccard.test(as.numeric(zz[1]), as.numeric(zz[2]), method = "exact")

+ Jaccard\_distances[i,c(1,2)] <- c(uu,vv$pvalue)

+ }

+ else if (length(grep(rownames(Global\_jaccard\_matrix), pattern=classes[i]))==0) {

+ Jaccard\_distances[i,c(1,2)] <- c("NA","NA")

+ }

+ else{

+ uu<- jaccard(as.numeric(zz[,1]), as.numeric(zz[,2]))

+ vv <-jaccard.test(as.numeric(zz[,1]), as.numeric(zz[,2]), method = "exact")

+ Jaccard\_distances[i,c(1,2)] <- c(uu,vv$pvalue)

+ }

+ #print(zz)

+ }

+

+ assign(paste0 ("Jaccard\_distances\_B\_Lipids\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2), Jaccard\_distances)

+ write.csv(file=paste0 ("Jaccard\_distances\_B\_Lipids\_", tissues\_to\_compare[1], "\_", tissues\_to\_compare[2], "\_", considered\_generation, "\_", substr(considered\_mode, 2,4), "\_", considered\_model\_1, "\_", considered\_model\_2,".csv"), Jaccard\_distances)

+ }

>

> options(warn=0)