#CG cross correaltion

read.csv(file.choose(), header=T, sep=",")->wetchem

wetchem [1:18,]-> wetchem.A4

wetchem [19:36,]-> wetchem.AP13

wetchem [37:54,]-> wetchem.VS16

genes [1:18,]-> genes.A4

genes[19:36,]-> genes.AP13

genes [37:54,]-> genes.VS16

wg.matrix <-cbind(wetchem, genes)

t(wg.matrix)-> wg.matrix.t

wg.matrix.t[,1:18]-> wg.matrix.t.A4

wg.matrix.t[,19:36]-> wg.matrix.t.AP13

wg.matrix.t[, 37:54]-> wg.matrix.t.VS16

library(rsgcc)

wg.gcc.total.A4 <-cor.matrix(wg.matrix.t.A4, cormethod = "GCC", cpus=3, style = "all.pairs", output= "matrix",sigmethod = "two.sided", pernum = 10000)

wg.gcc.total.AP13 <-cor.matrix(wg.matrix.t.AP13, cormethod = "GCC", cpus=3, style = "all.pairs", output= "matrix",sigmethod = "two.sided", pernum = 10000)

wg.gcc.total.VS16 <-cor.matrix(wg.matrix.t.VS16, cormethod = "GCC", cpus=3, style = "all.pairs", output= "matrix",sigmethod = "two.sided", pernum = 10000)

library(qvalue)

take.topr(wg.gcc.total.A4 [[2]],ncol(wetchem.A4),ncol(genes.A4))-> wg.gcc.total.A4.p

take.topr(wg.gcc.total.AP13 [[2]],ncol(wetchem.AP13),ncol(genes.AP13))-> wg.gcc.total.AP13.p

take.topr(wg.gcc.total.VS16 [[2]],ncol(wetchem.VS16),ncol(genes.VS16))-> wg.gcc.total.VS16.p

take.topr(wg.gcc.total.A4[[1]],ncol(wetchem.A4),ncol(genes.A4))-> wg.gcc.total.A4.gcc

take.topr(wg.gcc.total.AP13[[1]],ncol(wetchem.AP13),ncol(genes.AP13))-> wg.gcc.total.AP13.gcc

take.topr(wg.gcc.total.VS16[[1]],ncol(wetchem.VS16),ncol(genes.VS16))-> wg.gcc.total.VS16.gcc

library(gplots)

break.pairs<-c(seq(-1,-0.4,by=0.6/(127)),0, seq(0.4,1,by=0.6/(127)))

# adjust order of chem and genes anf rename them

wg.gcc.total.A4.gcc [,c(3,1,11,2,5,6,10,7,8,9,4)] -> wg.gcc.total.A4.gcc.ordered1

wg.gcc.total.A4.gcc.ordered1 [c(4,5,6,3,2,1,8,11,14,10,13,9,7), ]-> wg.gcc.total.A4.gcc.ordered2

rownames(wg.gcc.total.A4.gcc.ordered2) [10]<- "Glc"

colnames(wg.gcc.total.A4.gcc.ordered2) [8]<- "COMT1"

colnames(wg.gcc.total.A4.gcc.ordered2) [10]<- "COMT3"

wg.gcc.total.AP13.gcc [,c(3,1,11,2,5,6,10,7,8,9,4)] -> wg.gcc.total.AP13.gcc.ordered1

wg.gcc.total.AP13.gcc.ordered1 [c(4,5,6,3,2,1,8,11,14,10,13,9,7), ]-> wg.gcc.total.AP13.gcc.ordered2

rownames(wg.gcc.total.AP13.gcc.ordered2) [10]<- "Glc"

colnames(wg.gcc.total.AP13.gcc.ordered2) [8]<- "COMT1"

colnames(wg.gcc.total.AP13.gcc.ordered2) [10]<- "COMT3"

wg.gcc.total.VS16.gcc [,c(3,1,11,2,5,6,10,7,8,9,4)] -> wg.gcc.total.VS16.gcc.ordered1

wg.gcc.total.VS16.gcc.ordered1 [c(4,5,6,3,2,1,8,11,14,10,13,9,7), ]-> wg.gcc.total.VS16.gcc.ordered2

rownames(wg.gcc.total.VS16.gcc.ordered2) [10]<- "Glc"

colnames(wg.gcc.total.VS16.gcc.ordered2) [8]<- "COMT1"

colnames(wg.gcc.total.VS16.gcc.ordered2) [10]<- "COMT3"

heatmap.2(t(wg.gcc.total.A4.gcc.ordered2), dendrogram="none", Rowv= FALSE, Colv= FALSE, scale="none",col = bluewhitered256, margins=c(5,10),colsep=1: ncol(t(wg.gcc.total.A4.gcc.ordered2)),rowsep=1: nrow(t(wg.gcc.total.A4.gcc.ordered2)), sepcolor="lightgrey",trace="none",cexRow=1.2,cexCol=1.2,density.info="none", keysize=1,breaks= break.pairs, na.col="yellow")

heatmap.2(t(wg.gcc.total.AP13.gcc.ordered2), dendrogram="none", Rowv= FALSE, Colv= FALSE, scale="none",col = bluewhitered256, margins=c(5,10),colsep=1: ncol(t(wg.gcc.total.AP13.gcc.ordered2)),rowsep=1: nrow(t(wg.gcc.total.AP13.gcc.ordered2)), sepcolor="lightgrey",trace="none",cexRow=1.2,cexCol=1.2,density.info="none", keysize=1,breaks= break.pairs, na.col="yellow")

heatmap.2(t(wg.gcc.total.VS16.gcc.ordered2), dendrogram="none", Rowv= FALSE, Colv= FALSE, scale="none",col = bluewhitered256, margins=c(5,10),colsep=1: ncol(t(wg.gcc.total.VS16.gcc.ordered2)),rowsep=1: nrow(t(wg.gcc.total.VS16.gcc.ordered2)), sepcolor="lightgrey",trace="none",cexRow=1.2,cexCol=1.2,density.info="none", keysize=1,breaks= break.pairs, na.col="yellow")

(wg.gcc.total.A4.gcc+wg.gcc.total.AP13.gcc+wg.gcc.total.VS16.gcc)/3 -> wg.gcc.total.genotype.average.gcc

wg.gcc.total.A4.gcc-> wg.gcc.total.genotype.sd.gcc

for (i in 1:nrow(wg.gcc.total.genotype.sd.gcc)) { for (j in 1:ncol(wg.gcc.total.genotype.sd.gcc)){ sd(c(wg.gcc.total.A4.gcc[i,j], wg.gcc.total.AP13.gcc[i,j], wg.gcc.total.VS16.gcc [i,j]) ) -> wg.gcc.total.genotype.sd.gcc[i,j]

}}

log( abs(wg.gcc.total.genotype.average.gcc /wg.gcc.total.genotype.sd.gcc),2)-> wg.gcc.total.genotype.ratio.gcc

whitered256 <- colorRampPalette(c("white", "red") ) (256)

break.pairs<- c(seq(0,max(wg.gcc.total.genotype.ratio.gcc),by= max(wg.gcc.total.genotype.ratio.gcc)/(256)))

heatmap.2(t(wg.gcc.total.genotype.ratio.gcc), dendrogram="none", Rowv= FALSE, Colv= FALSE, scale="none", col = whitered256 , margins=c(5,10),colsep=1: ncol(t(wg.gcc.total.A4.gcc)),rowsep=1: nrow(t(wg.gcc.total.A4.gcc)), sepcolor="lightgrey",trace="none",cexRow=1,cexCol=1,density.info="none", keysize=1, na.col="yellow", breaks= break.pairs)

~~wg.gcc.total.p[-c(4,5,6),]-> wg.gcc.total.p.PS~~

~~wg.gcc.total.gcc[-c(4,5,6),]-> wg.gcc.total.gcc.PS~~

qvalue(wg.gcc.total.A4.p)-> wg.gcc.total.A4.q

qvalue(wg.gcc.total.AP13.p, lambda=seq(0,0.89999,0.05) )-> wg.gcc.total.AP13.q

qvalue(wg.gcc.total.VS16.p)-> wg.gcc.total.VS16.q

~~qvalue(wg.gcc.total.p.PS) -> wg.gcc.total.q.PS~~

qplot(wg.gcc.total.A4.q)

qplot(wg.gcc.total.AP13.q)

qplot(wg.gcc.total.VS16.q)

~~qplot(wg.gcc.total.q.PS)~~

gccheatmap (wg.gcc.total.A4.gcc, wg.gcc.total.A4.q, 0.01, 0) ->gcc.total.heatmap.A4

gccheatmap (wg.gcc.total.AP13.gcc, wg.gcc.total.AP13.q, 0.01, 0) ->gcc.total.heatmap.AP13

gccheatmap (wg.gcc.total.VS16.gcc, wg.gcc.total.VS16.q, 0.01, 0) ->gcc.total.heatmap.VS16

~~gccheatmap (wg.gcc.total.gcc.PS, wg.gcc.total.q.PS, 0.001, 0) ->gcc.total.heatmap.PS~~

range(gcc.total.heatmap)

range(abs(gcc.total.heatmap [gcc.total.heatmap!=0]))

~~range(gcc.total.heatmap.PS)~~

~~range(abs(gcc.total.heatmap.PS [gcc.total.heatmap.PS!=0]))~~

break.pairs<-c(seq(-0.9,-0.2,by=0.7/(127)),0, seq(0.2,0.9,by=0.7/(127)))

hist(gcc.total.heatmap[gcc.total.heatmap!=0],breaks=100)

#True NULL hypothesis

wg.gcc.total.q$qvalue->q.rank.matrix

wg.gcc.total.q.PS$qvalue->q.rank.matrix.PS

matrix(rank(wg.gcc.total.q $qvalue),nrow=nrow(wg.gcc.total.q$qvalue),ncol= ncol(wg.gcc.total.q$qvalue))->q.rank.matrix

matrix(rank(wg.gcc.total.q.PS $qvalue),nrow=nrow(wg.gcc.total.q.PS$qvalue),ncol= ncol(wg.gcc.total.q.PS$qvalue))->q.rank.matrix.PS

rownames(wg.gcc.total.q$qvalue)-> rownames(q.rank.matrix)

colnames(wg.gcc.total.q$qvalue)-> colnames(q.rank.matrix)

rownames(wg.gcc.total.q.PS$qvalue)-> rownames(q.rank.matrix.PS)

colnames(wg.gcc.total.q.PS$qvalue)-> colnames(q.rank.matrix.PS)

max(q.rank.matrix)\*(1- wg.gcc.total.q$pi0)->low.boundary

max(q.rank.matrix.PS)\*(1- wg.gcc.total.q.PS $pi0)->low.boundary.PS

gcc.total.heatmap -> gcc.total.heatmap.tn

gcc.total.heatmap.PS -> gcc.total.heatmap.tn.PS

gcc.total.heatmap.tn [low.boundary.PS <q.rank.matrix]<-NA

gcc.total.heatmap.tn.PS [low.boundary <q.rank.matrix.PS]<-NA

#dendrogram for heatmap

library(rsgcc)

t(as.matrix(wetchem))-> den.wet.total.t

t(as.matrix(wetchem[,-c(4,5,6)]))-> den.wet.total.t.PS

cor.matrix(den.wet.total.t, cormethod = "GCC", cpus=3, style = "all.pairs", output= "matrix",sigmethod = "two.sided", pernum = 0) -> den.wet.total.t.matrix

cor.matrix(den.wet.total.t.PS, cormethod = "GCC", cpus=3, style = "all.pairs", output= "matrix",sigmethod = "two.sided", pernum = 0) -> den.wet.total.t.matrix.PS

as.dist(1- den.wet.total.t.matrix[[1]])->corre.distance

as.dist(1- den.wet.total.t.matrix.PS[[1]])->corre.distance.PS

intermediate.dend<-as.dendrogram(hclust(corre.distance,method="average"))

intermediate.dend.PS<-as.dendrogram(hclust(corre.distance.PS,method="average"))

library(gplots)

tiff("wet chemistry and genes gcc correlation.matrix.total.tif" , width=2200,height=2200,res=150)

heatmap.2(t(gcc.total.heatmap.tn), dendrogram="column", Rowv= FALSE, Colv= intermediate.dend, scale="none",col = bluewhitered256, margins=c(8,10),colsep=1: ncol(t(gcc.total.heatmap.tn)),rowsep=1: nrow(t(gcc.total.heatmap.tn)), sepcolor="lightgrey",trace="none",cexRow=1,cexCol=1,density.info="none", keysize=0.6,breaks= break.pairs, na.col="yellow")

dev.off()

tiff("wet chemistry and genes gcc correlation.matrix.total without diestibility.tif" , width=2200,height=2200,res=300)

heatmap.2(t(gcc.total.heatmap.tn.PS), dendrogram="column", Rowv= FALSE, Colv= intermediate.dend.PS, scale="none",col = bluewhitered256, margins=c(8,10),colsep=1: ncol(t(gcc.total.heatmap.tn.PS)),rowsep=1: nrow(t(gcc.total.heatmap.tn.PS)), sepcolor="lightgrey",trace="none",cexRow=1,cexCol=1,density.info="none", keysize=0.6,breaks= break.pairs, na.col="yellow")

dev.off()

write.csv(t(gcc.total.heatmap.tn), file = " wet chemistry and genes gcc correlation.matrix.total.csv" )

write.csv(wg.gcc.total [[1]], file = " wet chemistry and genes gcc correlation.matrix.total.complete.csv" )

write.csv(wg.gcc.total.q$qvalues, file = " wet chemistry and genes gcc correlation.matrix.total.qvalues.csv" )

write.csv(wg.gcc.total.q.PS$qvalues, file = " wet chemistry and genes gcc correlation.matrix.total.qvalues.without digestibility.csv" )

#Multiplot

sapply(as.data.frame(wg.matrix),z.score.na)-> wetchemandgenes.z

pdf("all correlation plots.pdf", onefile=T, width=8,height=8)

for(i in 1:ncol(wetchemandgenes.z)){ for(j in 1:ncol(wetchemandgenes.z)){if(i!=j){

plot(wetchemandgenes.z[,i], wetchemandgenes.z[,j], xlab=colnames(wetchemandgenes.z)[i], ylab=colnames(wetchemandgenes.z)[j] , main= paste(colnames(wetchemandgenes.z)[i], colnames(wetchemandgenes.z)[j],sep=" cor "),xlim=c(-5,5),ylim=c(-5,5),pch=16, abline(lm(wetchemandgenes.z[,j]~ wetchemandgenes.z[,i]),col="red" ) )

}}

}

dev.off()

#wetchem correaltion

t(wetchem)-> wetchem.matrix.t

library(rsgcc)

wetchem.gcc.total<-cor.matrix(wetchem.matrix.t, cormethod = "GCC", cpus=3, style = "all.pairs", output= "matrix",sigmethod = "two.sided", pernum = 100000)

library(qvalue)

wetchem.gcc.total [[1]] -> wetchem.gcc.total.gcc

wetchem.gcc.total [[1]] -> wetchem.gcc.total.q

qvalue(wetchem.gcc.total [[2]][upper.tri(wetchem.gcc.total [[2]])] )->inter.q

inter.q $qvalue -> wetchem.gcc.total.q [upper.tri(wetchem.gcc.total.q)]

qplot(inter.q)

plot(wetchem.gcc.total.q [upper.tri(wetchem.gcc.total.q)] , wetchem.gcc.total [[2]][upper.tri(wetchem.gcc.total [[2]])] ) #q value and p-value plot

wetchem.gcc.total.gcc [lower.tri(wetchem.gcc.total.gcc, diag = T)]<-0

wetchem.gcc.total.q [lower.tri(wetchem.gcc.total.q, diag = T)]<-2

wetchem.gcc.total.q.l <-list(qvalue=wetchem.gcc.total.q)

gccheatmap (wetchem.gcc.total.gcc, wetchem.gcc.total.q.l, 0.001, 0) ->wetchem.gcc.total.heatmap

range(wetchem.gcc.total.heatmap)

range(abs(wetchem.gcc.total.heatmap [wetchem.gcc.total.heatmap!=0]))

break.pairs<-c(seq(-1,-0.4,by=0.6/(127)),0, seq(0.4,1,by=0.6/(127)))

hist(gcc.total.heatmap[gcc.total.heatmap!=0],breaks=100)

#True NULL hypothesis

wetchem.gcc.total.q->q.rank.matrix

matrix(rank(wetchem.gcc.total.q),nrow=nrow(wetchem.gcc.total.q),ncol= ncol(wetchem.gcc.total.q))->q.rank.matrix

rownames(wetchem.gcc.total.q)-> rownames(q.rank.matrix)

colnames(wetchem.gcc.total.q)-> colnames(q.rank.matrix)

max(q.rank.matrix [q.rank.matrix< max(q.rank.matrix)])\*(1- inter.q$pi0)->low.boundary

wetchem.gcc.total.heatmap -> wetchem.gcc.total.heatmap.tn

wetchem.gcc.total.heatmap.tn [q.rank.matrix <max(q.rank.matrix)& q.rank.matrix > low.boundary]<-NA

wetchem.gcc.total.heatmap.tn->wetchem.gcc.total.heatmap.tn.mirro

for(i in 1:nrow(wetchem.gcc.total.heatmap.tn)){ for(j in 1:ncol (wetchem.gcc.total.heatmap.tn)){ if(i<j){ wetchem.gcc.total.heatmap.tn[i,j]-> wetchem.gcc.total.heatmap.tn.mirro[j,i]

}}}

library(gplots)

tiff("wet chemistry gcc correlation.matrix.total.tif" , width=2200,height=2200,res=200)

heatmap.2(wetchem.gcc.total.heatmap.tn.mirro, dendrogram="none", Rowv= FALSE, Colv= NA, scale="none",col = bluewhitered256, margins=c(8,10),colsep=1: ncol(wetchem.gcc.total.heatmap.tn.mirro),rowsep=1: nrow(wetchem.gcc.total.heatmap.tn.mirro), sepcolor="lightgrey",trace="none",cexRow=1,cexCol=1,density.info="none", keysize=0.6,breaks= break.pairs, na.col="yellow")

dev.off()

write.csv(wetchem.gcc.total.heatmap.tn.mirro, file = " wet chemistry gcc correlation.matrix.total.csv" )

write.csv(wetchem.gcc.total [[1]], file = " wet chemistry gcc correlation.matrix.total.complete.csv" )

write.csv(wetchem.gcc.total.q, file = " wet chemistry gcc correlation.matrix.total.qvalues.csv" )

#Only DG and CW component

wetchem.gcc.total.heatmap.tn.mirro[-c(4,5,6),c(4,5,6)]->DGandCW

tiff("cell wall and digestibility gcc.total.tif" , width=800,height=2200,res=300)

heatmap.2(DGandCW, dendrogram="none", Rowv= FALSE, Colv= NA, scale="none",col = bluewhitered256, margins=c(8,10),colsep=1: ncol(wetchem.gcc.total.heatmap.tn.mirro),rowsep=1: nrow(wetchem.gcc.total.heatmap.tn.mirro), sepcolor="lightgrey",trace="none",cexRow=1,cexCol=1,density.info="none", keysize=0.6,breaks= break.pairs, na.col="yellow")

dev.off()

#all component and genes

wg.gcc.total[[2]]-> cnet.gcc.total.p

wg.gcc.total[[1]]-> cnet.gcc.total.gcc

cnet.gcc.total.p-> cnet.gcc.total.q

qvalue(cnet.gcc.total.p[upper.tri(cnet.gcc.total.p)] )$ qvalue-> cnet.gcc.total.q[upper.tri(cnet.gcc.total.q)]

cnet.gcc.total.q.l <-list(qvalue= cnet.gcc.total.q)

gccheatmap (cnet.gcc.total.gcc, cnet.gcc.total.q.l, 0.001, 0) ->cnet.gcc.total.heatmap

cnet.gcc.total.heatmap [lower.tri(cnet.gcc.total.heatmap, diag=T)]<-NA

Term1<-character(0)

Term2<- character(0)

GiniCoeff<-numeric(0)

for(i in 1:nrow(cnet.gcc.total.heatmap)){for(j in 1:ncol(cnet.gcc.total.heatmap)){c(GiniCoeff, cnet.gcc.total.heatmap [i,j])-> GiniCoeff ; c(Term1,rownames(cnet.gcc.total.heatmap)[i])-> Term1; c(Term2,colnames(cnet.gcc.total.heatmap)[j])-> Term2}}

cbind(Term1, Term2, GiniCoeff)-> networkoutput

write.csv(networkoutput, file = "complete.network.gcc.network.format.csv" )

#End

# For network

Term1<-character(0)

Term2<- character(0)

GiniCoeff<-numeric(0)

for(i in 1:nrow(wetchem.gcc.total.heatmap)){for(j in 1:ncol(wetchem.gcc.total.heatmap)){c(GiniCoeff, wetchem.gcc.total.heatmap[i,j])-> GiniCoeff ; c(Term1,rownames(wetchem.gcc.total.heatmap)[i])-> Term1; c(Term2,colnames(wetchem.gcc.total.heatmap)[j])-> Term2}}

cbind(Term1, Term2, GiniCoeff)-> networkoutput

write.csv(networkoutput, file = "wetchem.gcc.network.format.csv" )

##point check

rownames(wg.matrix.t)[3]

rownames(wg.matrix.t)[19]

cor.pair(c(3,19), wg.matrix.t, cormethod = "GCC", pernum = 0, sigmethod = "two.sided")