Supplementary material

Supplementary figures

PCA of ChIP-seq data before normalization

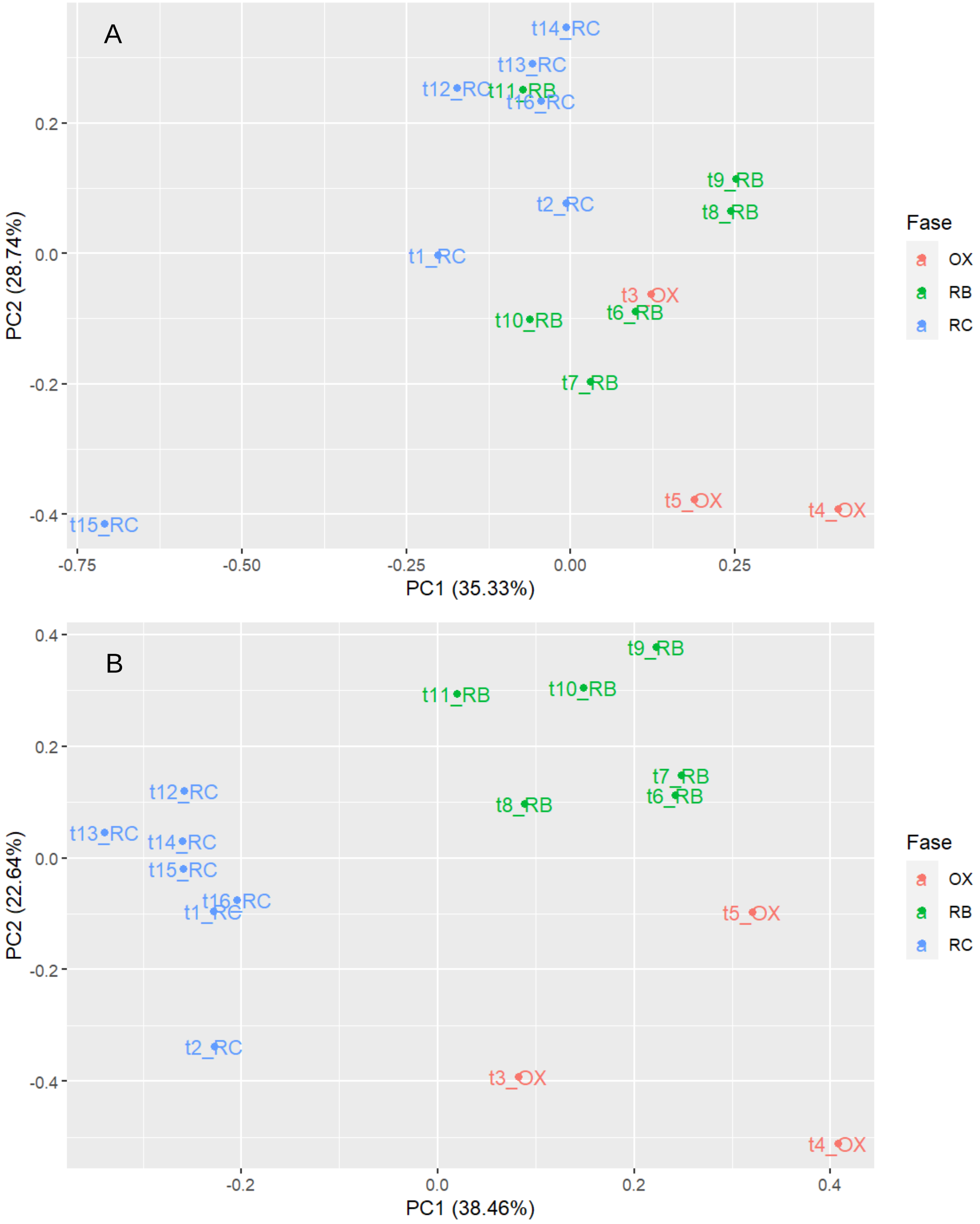


Figure 33: PCA of samples before H3 control normalization. Samples are coloured according to the corresponding phase of the cycle. Based on H3K18ac (A) and H3K9ac (B) ChIP-seq data.

Pathway Enrichment Score for MAMBA

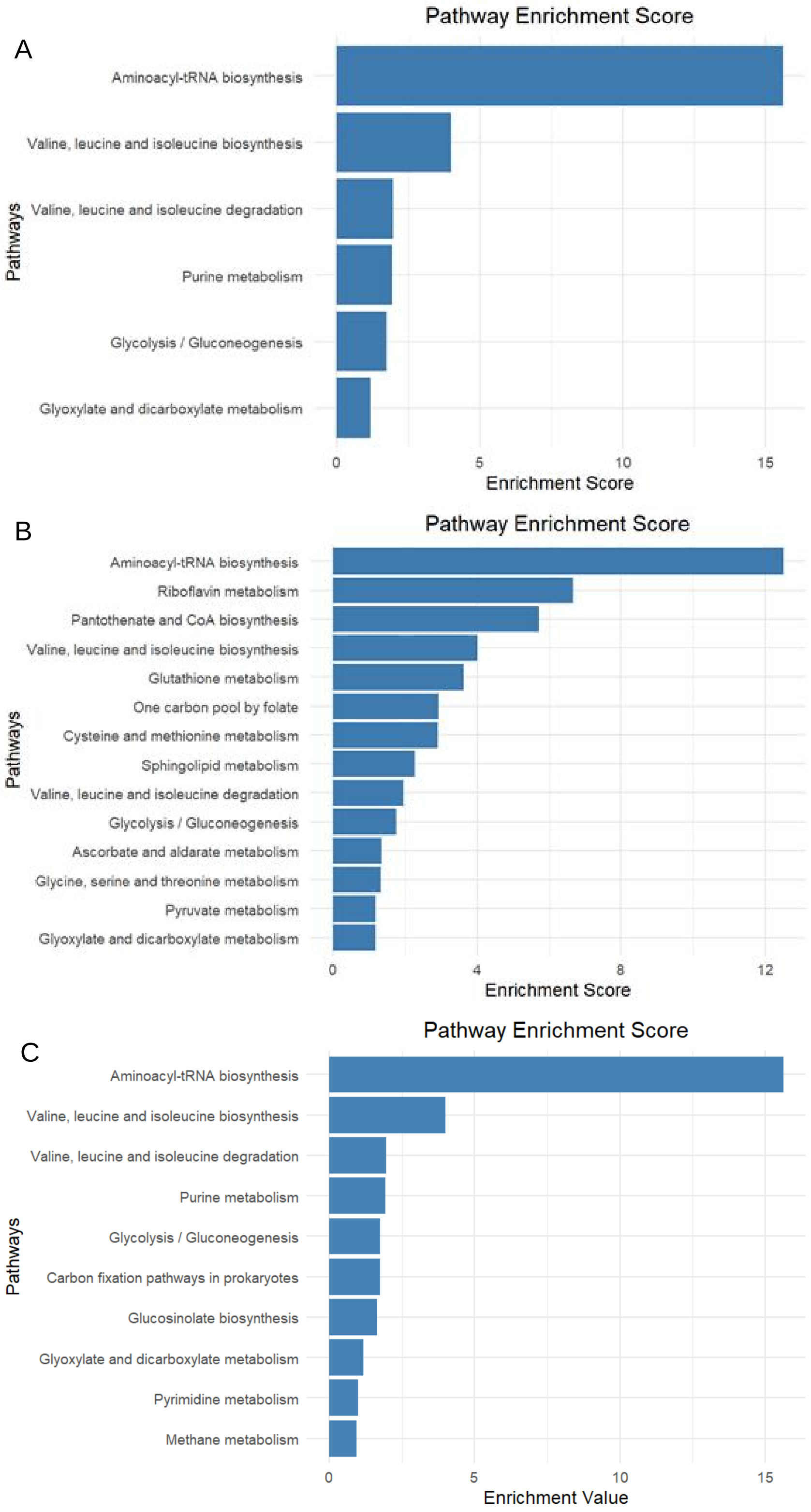


Figure 34: Pathway Enrichment Score (PES) in the OX phase. Relevant Pathways identified by MAMBA with the corresponding PES based on RNA (A), H3K18ac (B) and H3K18ac (C) information.

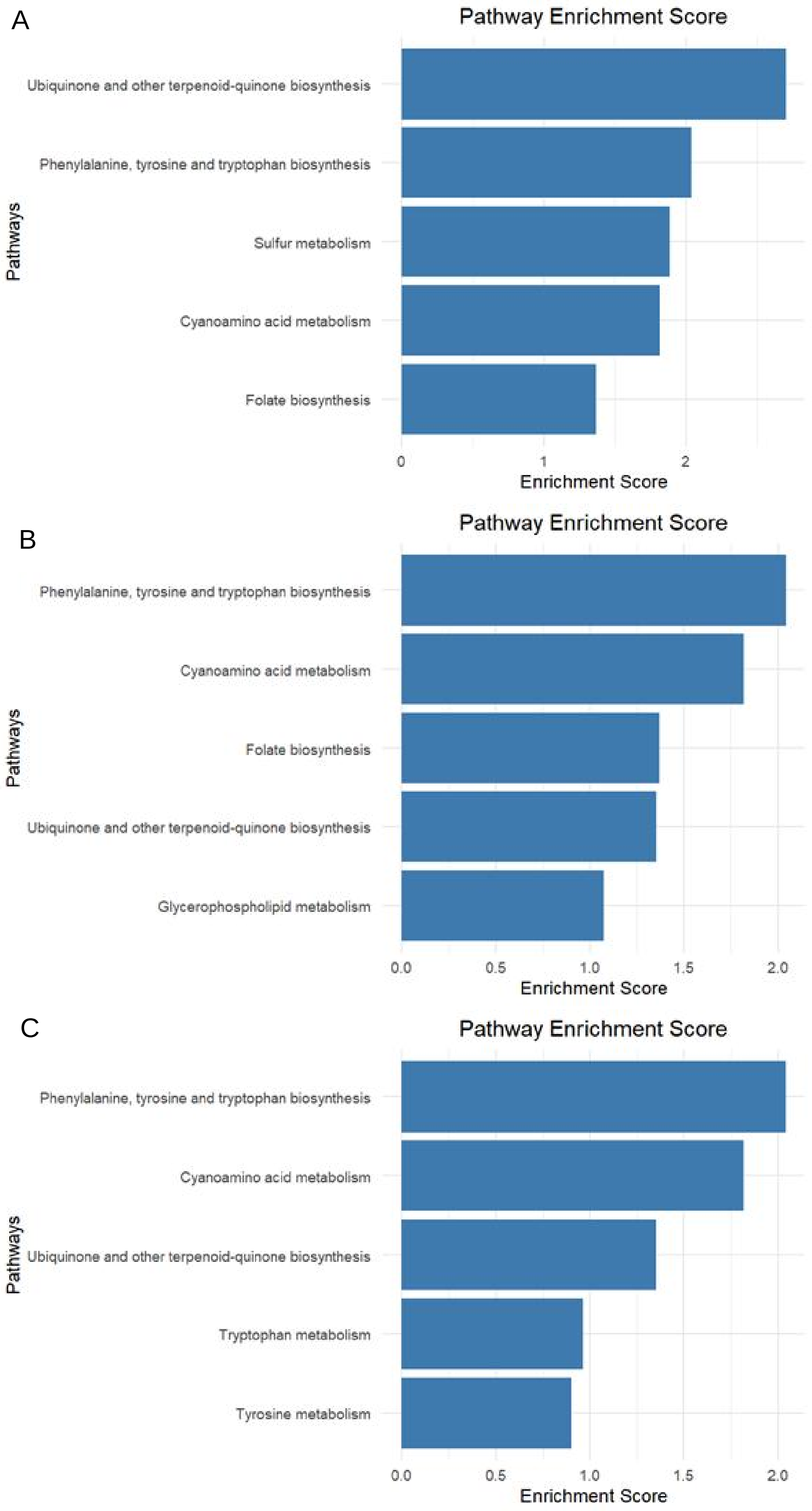


Figure 35: Pathway Enrichment Score (PES) in the RB phase. Relevant Pathways identified by MAMBA with the corresponding PES based on RNA (A), H3K18ac (B) and H3K18ac (C) information.

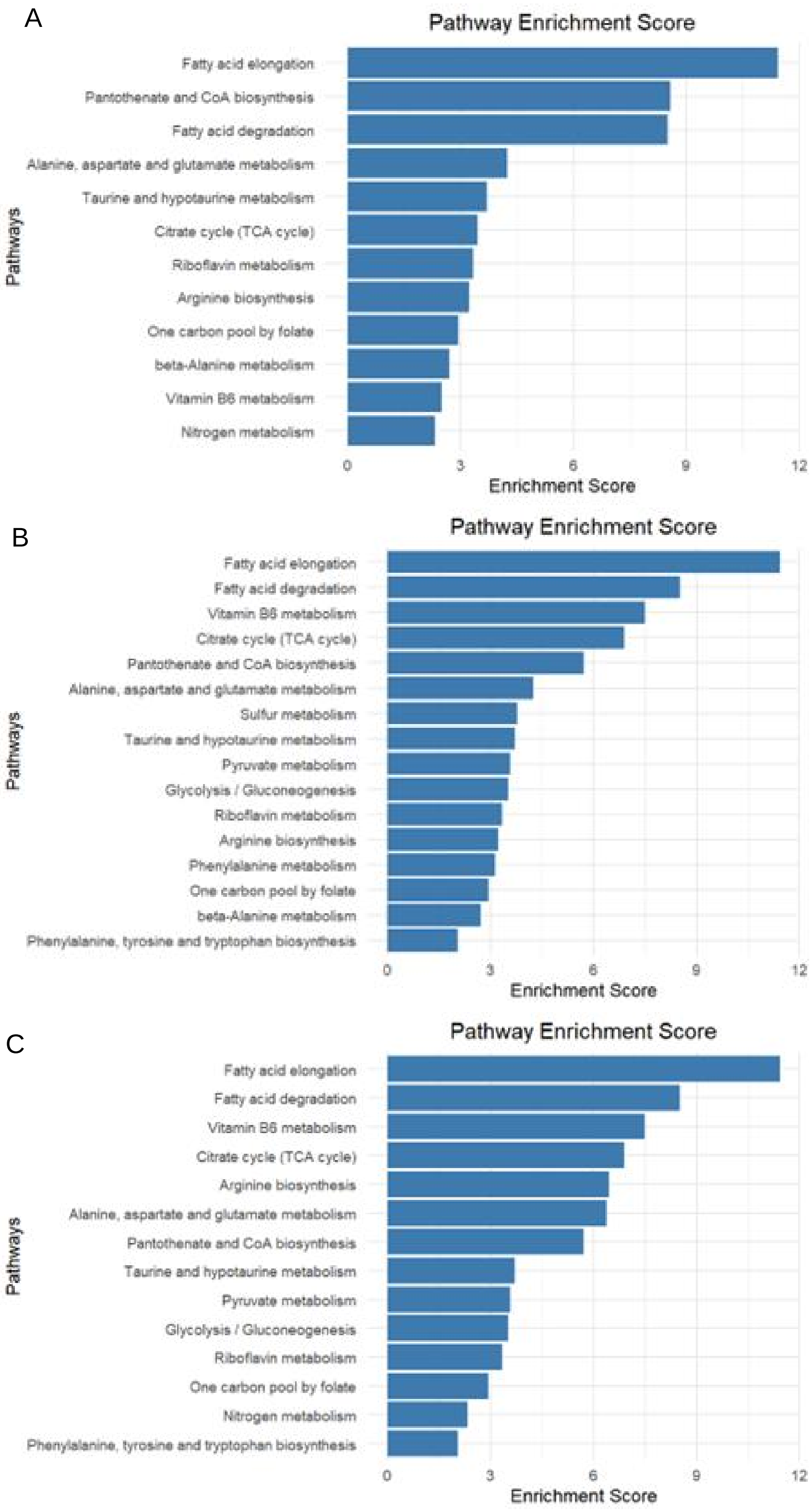
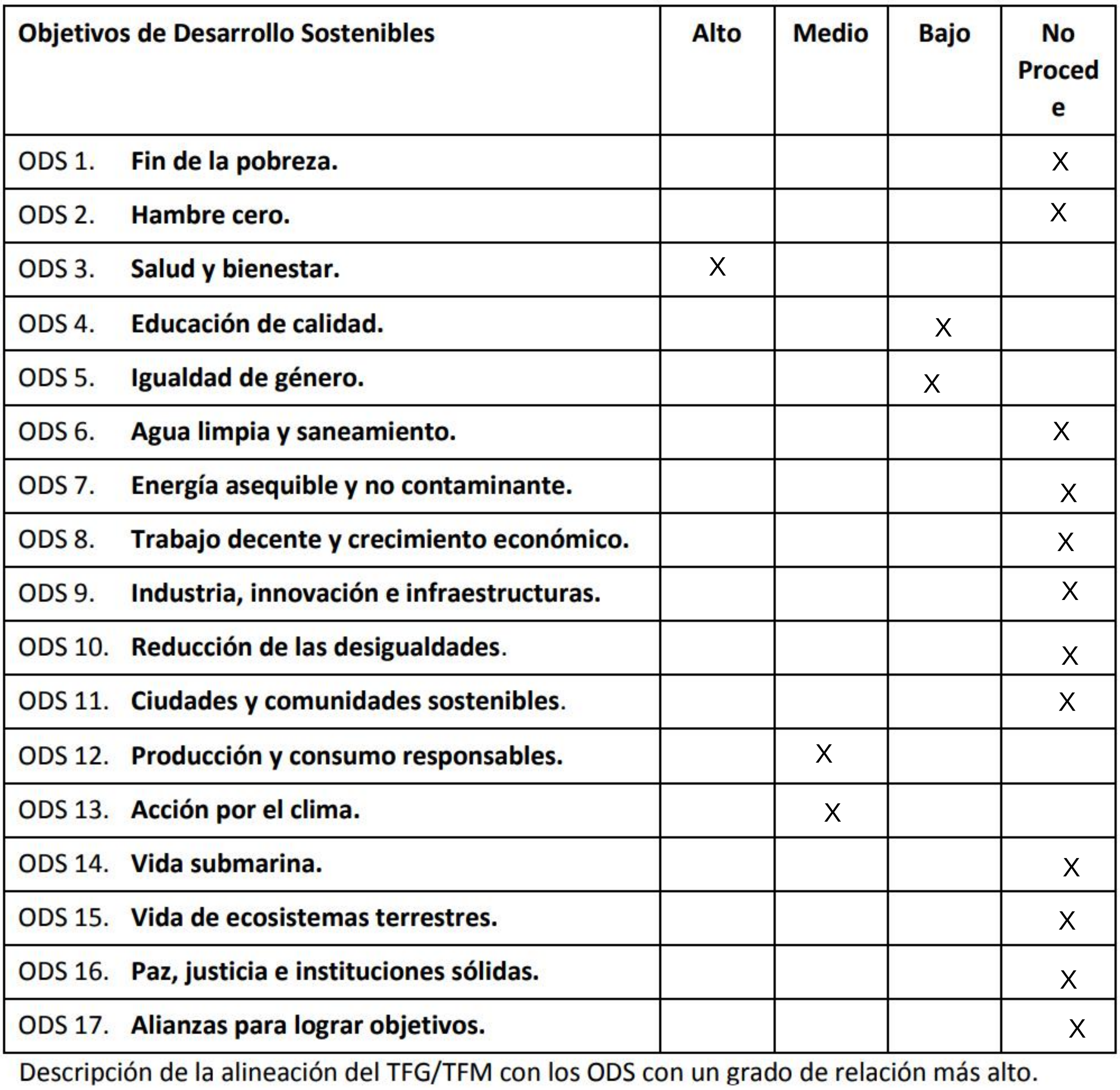


Figure 36: Pathway Enrichment Score (PES) in the RC phase. Relevant Pathways identified by MAMBA with the corresponding PES based on RNA (A), H3K18ac (B) and H3K18ac (C) information.

Sustainable Development Goals

This project is related to several Sustainable Development Goals (SDG) (Fig. 37):

- **Health and Well-being (SDG 3):** Developing strategies that enable to model cell metabolism based on experimental data is key for understanding the molecular mechanisms of complex diseases and creating effective treatments.
- **Climate Action (SDG 11)** and **Responsible production and consumption (SDG 12)**: In this work, experimental data were retrieved from public repositories. Analysing pre-existing data from a different scientific perspective allowed to draw significant biological conclusions while minimizing the production of plastic waste or other contaminants. In this sense, Bioinformatic approaches favour a more sustainable scientific when compared to wetlab.
- **Quality Education (SDG 4)** and **Gender Equality (SDG 5):** This work contributes the development and growth of girls and woman in science.



Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.			X	
ODS 5. Igualdad de género.			X	
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto.

Figure 37: Sustainable Development Goals (SDG). Each SDG has been assigned to a certain level of relationship with the project (High, Medium, Low, Not applicable).

R script

The R script used for functional analysis of MAMBA's flux prediction to obtain the biologically relevant pathways in the different phases is presented as an example of the bioinformatic workflow that has been followed in this work.

R script for functional analysis of MAMBA

2023-06-03

```
library(ggplot2)
library(KEGGREST)
library(GSVA)
library(RColorBrewer)
```

Para el análisis de flujos de MAMBA (H3K9ac)

PASOS PREVIOS GENERALES - Lo primero es sacar los ID de los pathways:

```
name_KEGG<-read.table("C:/Users/crist/Downloads/id_kegg.tsv",sep = "\t")
#Con esto tengo las reacciones, pero necesito los pathways

reaction<-name_KEGG$V2
pathway_id<-c()
no_annotated<-c()
for (i in reaction){
  result <- tryCatch(keggGet(i), error = function(x) {return(NA)})
  if (!is.na(result)){
    result_name<-names(result[[1]]$PATHWAY)
    if(is.null(result_name)){
      no_annotated<-append(no_annotated,i)
      }
    pathway_id<-append(pathway_id,result_name)
  }
}

KEGG_pathway_ID<-unique(pathway_id)
```

KEGG_pathway_ID –> todos los pathways con su KEGG ID no_annotated –> reacciones CON KEGG ID que no tienen información de a qué pathway pertenecen

- Una vez tenemos una lista con todos los ID de los PATHWAYS, queremos una lista con todos los ID de las reacciones que hay en ese pathway dentro.

```
pathway_reaction<-c()
no_reaction<-c()
for (m in 1:length(KEGG_pathway_ID)){
  z<-KEGG_pathway_ID[m]
  to<-keggGet(z)
  me<-names(to[[1]]$REACTION)
  if(is.null(me)){
      no_reaction<-append(no_reaction,m)
      }
  pathway_reaction[[KEGG_pathway_ID[m]]]<- me
}
```

AHORA YA ESPECÍFICO DE LA K9

1. Cargar los flujos y quedarte solo con los exclusivos en alguna condición (DAR)

```
fluxes_alejandro_k9<-read.csv("C:\\Users\\crist\\Downloads\\reaction_activity_k9.csv")

fluxes_alejandro_k9$total<-apply(fluxes_alejandro_k9[2:4],1,sum)
fluxes_dif<-fluxes_alejandro_k9[which(fluxes_alejandro_k9$total != 3 & fluxes_alejandro_k9$total != 0),]

fluxes_dif<-fluxes_alejandro_k9[which(fluxes_alejandro_k9$total ==1),]
```

2. Definir las DAR para las 3 condiciones. El orden es RC, OX, RB.

```
reactions_ox<-fluxes_dif[which(fluxes_dif$Condition_2 == 1),]
dif_reactions_ox<-as.vector(reactions_ox$Reaction_id)

reactions_rc<-fluxes_dif[which(fluxes_dif$Condition_1==1),]
dif_reactions_rc<-as.vector(reactions_rc$Reaction_id)

reactions_rb<-fluxes_dif[which(fluxes_dif$Condition_3 == 1),]
dif_reactions_rb<-as.vector(reactions_rb$Reaction_id)
```

3. Anotar las reacciones del modelo con el KEGG_ID correspondiente

```
name_KEGG<-read.table("C:/Users/crist/Downloads/id_kegg.tsv",sep = "\t")
all_dif_reactions<-list(RB=dif_reactions_rb,RC=dif_reactions_rc,
                        OX=dif_reactions_ox)

KEGG_ID<-c()
KEGG_ID_all<-list()
order<-c("RB","RC","OX")
for (i in order){
  for (s in all_dif_reactions[[i]]){
    index<-which(name_KEGG$V1 == s)
    id<-name_KEGG[index,2]
    KEGG_ID<-append(KEGG_ID,id)
  }
  KEGG_ID_all[[i]]<-KEGG_ID
  KEGG_ID<-c()
}
```

4. Calcular el Pathway Enrichment Score: previamente se ha generado una lista con todos los pathways y todas las reacciones que hay dentro (pathway_reaction)

```
PSE_list_all<-list()

PES_new<-data.frame()
PES_all<-list()
for (i in order){
  for (r in 1:length(pathway_reaction)){
      e<-intersect(pathway_reaction[[r]],KEGG_ID_all[[i]])
      v<-length(pathway_reaction[[r]])
      y<-length(e)
      PES_score<-y/v*100
      PES_df<-data.frame("pathway"= names(pathway_reaction)[r], "PES" = PES_score, "phase" = i)
      PES_new<-rbind(PES_new, PES_df)
  }
  PES_all[[i]]<-PES_new
  PES_new<-data.frame()
}
```

3. Quedarnos solo con aquellos pathways que tengan un PES significativo

```
for (i in order){
  PES_all[[i]]<-PES_all[[i]][which(PES_all[[i]]$PES !=0),]
}
```

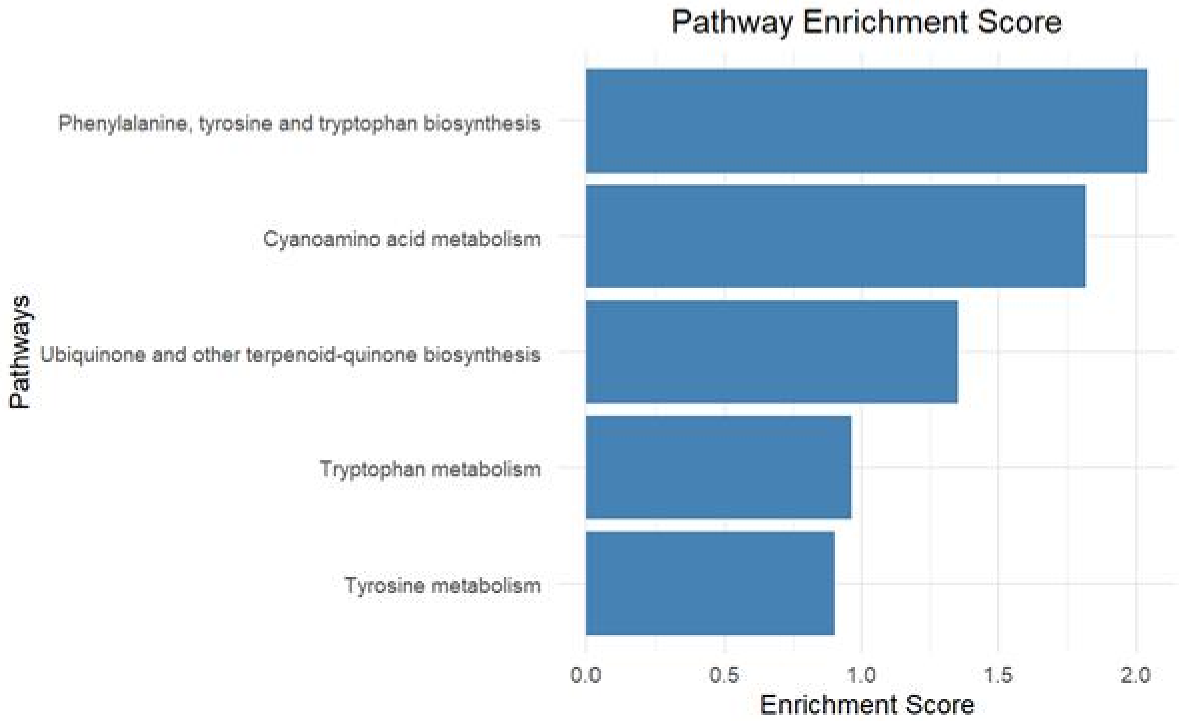
4. Extraer el nombre biológico descriptivo descriptivo de cada identificador. -cambiar el identificador de rn (general) a sce (de Saccharomyces Cerevisiae)
    - consulta de keggGet para saber el "name"
    - crear una data frame con todos los rn, sce y biological pathway
    - quedarse solo con los que tengan las tres cosas

```
df_ID_all<-data.frame()
for (i in order){
  for (p in names(pathway_reaction)){
    sce<-gsub("rn","sce",p)
    kegg_result<-tryCatch(keggGet(sce),error=function(x){return(NA)})
    if(!is.na(kegg_result))
      name<-as.character(kegg_result[[1]][["NAME"]])
      name_ok<-strsplit(name, "- Sac")[[1]][1]
      df_ID<-data.frame("sce"=sce,"biological_name"=name_ok,"pathway"=p)
      df_ID_all<-rbind(df_ID_all, df_ID)
      name<-"Hola"
    }
  df_ID_all<-unique(df_ID_all)
}
```

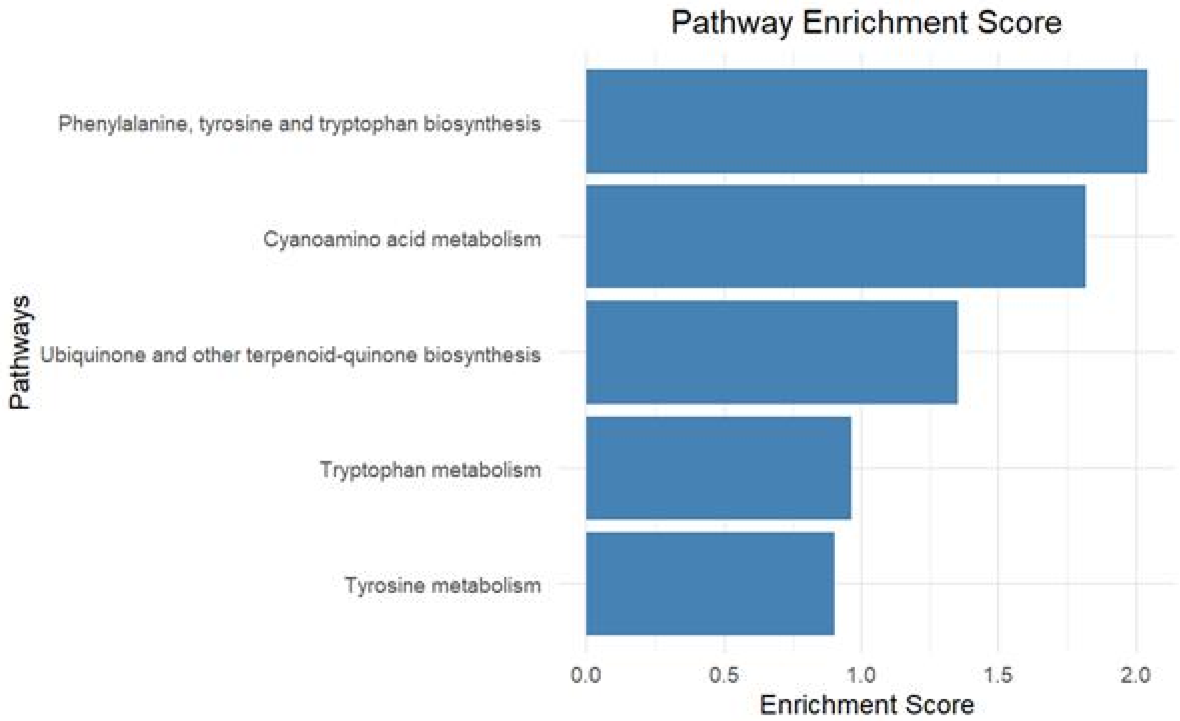
```
df_merged_all<-list()
final_df_all<-list()
for (i in order){
  df_merged_all[[i]]<-merge(PES_all[[i]],df_ID_all,by="pathway", all=TRUE)
  final_df_all[[i]]<-unique(na.omit(df_merged_all[[i]][which(df_merged_all[[i]]$phase ==

 i ),]))
}

for (i in order){
  final_df_all[[i]]<-final_df_all[[i]][which(final_df_all[[i]]$biological_name != "Hola"),]
}
```

5. Hacer un plot para cada fase por separado:

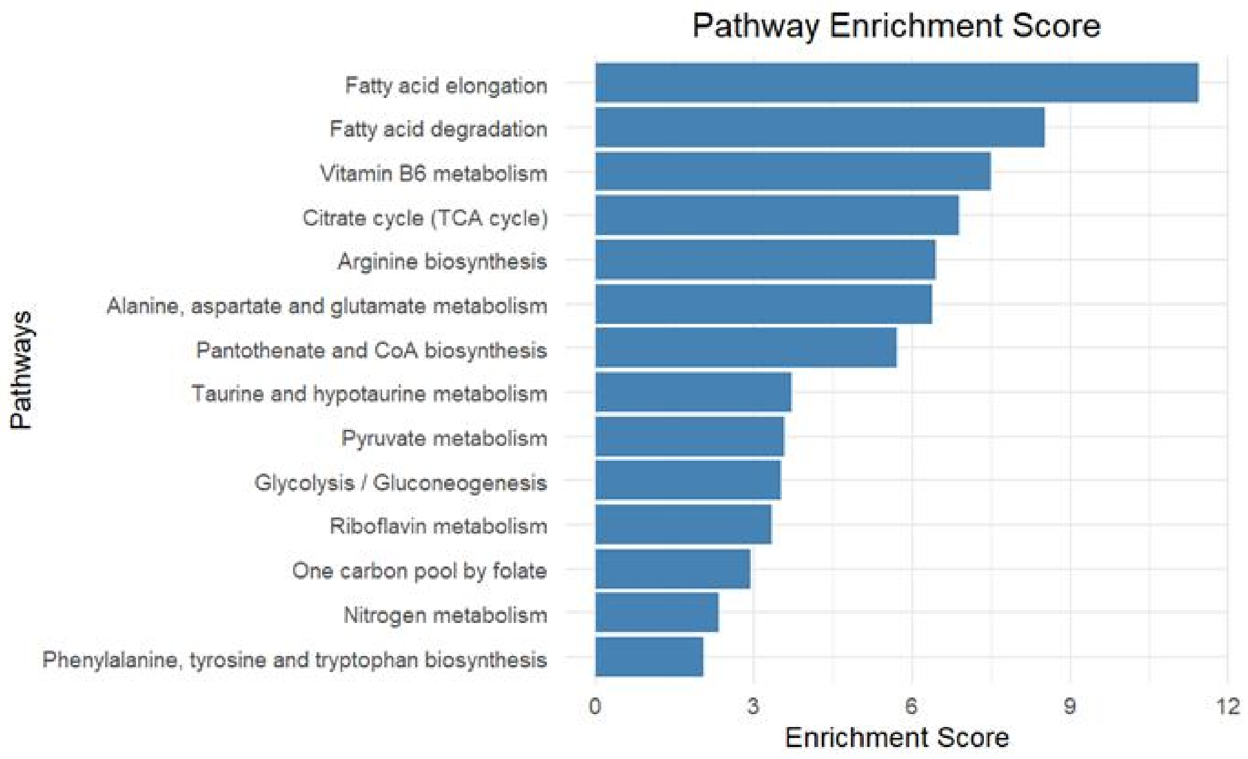
```
  ggplot(data = final_df_all[["OX"]], aes(x = PES, y = reorder(biological_name, PES))) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Enrichment Score", y = "Pathways") +
  ggtitle("Pathway Enrichment Score") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot(data = final_df_all[["RB"]], aes(x = PES, y = reorder(biological_name, PES))) +
geom_bar(stat = "identity", fill = "steelblue") +
labs(x = "Enrichment Score", y = "Pathways") +
ggtitle("Pathway Enrichment Score") +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5))
```



```
  ggplot(data = final_df_all[["RC"]][which(final_df_all[["RC"]]$PES>2),],

  aes(x = PES, y = reorder(biological_name, PES))) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Enrichment Score", y = "Pathways") +
  ggtitle("Pathway Enrichment Score") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



6. Quedarse solo con los pathways que tengan un PES mayor que el threshold

```
sig_pathways_sce_ox_k9<-final_df_all[["OX"]][which(final_df_all[["OX"]]$PES>2),]$sce
sig_pathways_sce_rb_k9<-final_df_all[["RB"]][which(final_df_all[["RB"]]$PES>2),]$sce
sig_pathways_sce_rc_k9<-final_df_all[["RC"]][which(final_df_all[["RC"]]$PES>2),]$sce

sig_pathways_sce_all_k9<-c(sig_pathways_sce_ox_k9,sig_pathways_sce_rb_k9,sig_pathways_sce_rc_k9)
```

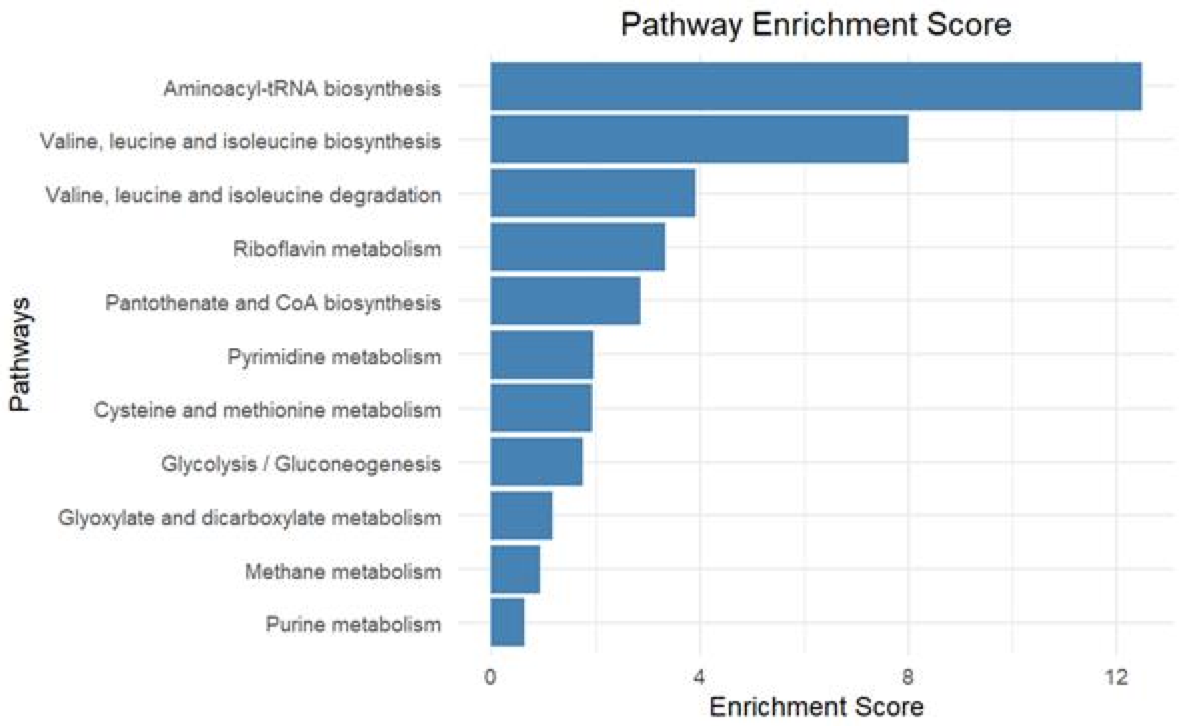
7. Para el GSVA:

- sacar todos los genes que que esten dentro de los pathways significativos

```
sig_genes_pathways_k9<-list()
for (m in sig_pathways_sce_all_k9){
  bu<-tryCatch(keggGet(m),error=function(x){return(NA)})
    if(!is.na(bu)){
      ho<-bu[[1]][["GENE"]]
      s<-length(ho)
      l<-seq(from=1,to=s,by=2)
      d<-bu[[1]][["NAME"]]
      e<-strsplit(d, "- Sac")
      e<-e[[1]][1]
      sig_genes_pathways_k9[[e]]<-ho[l]
      e<-"Hola"
  }
}
```

- hacer un GSVA solo considerando los pathways significativos

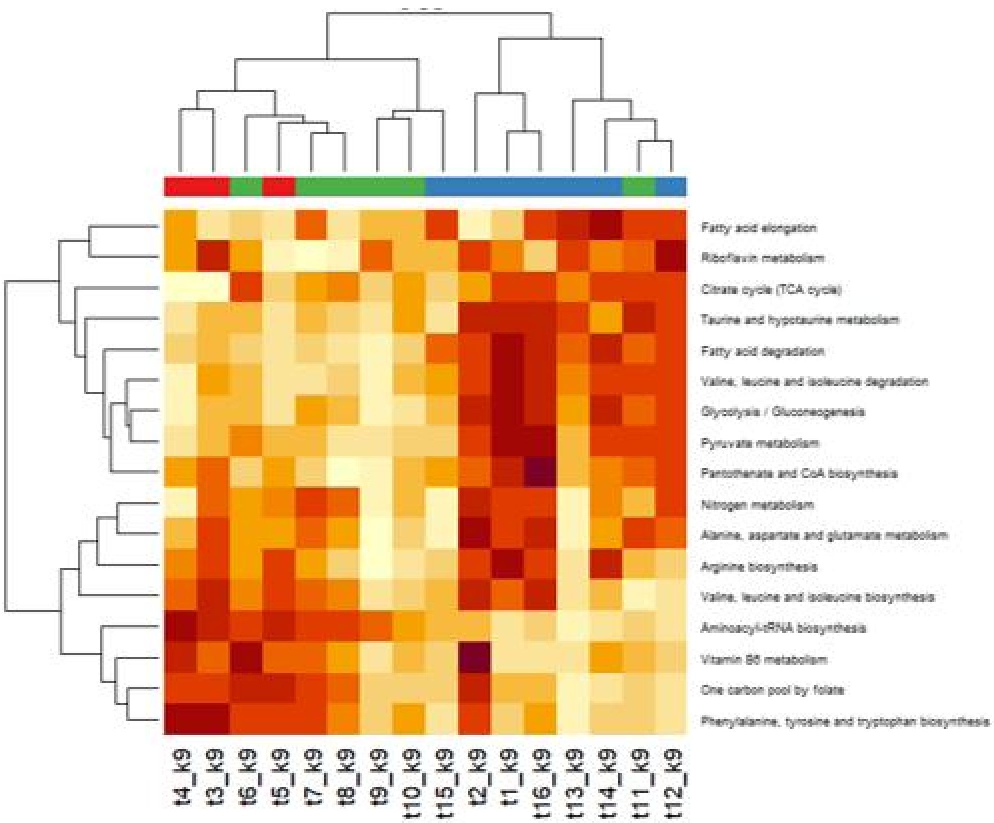
```
load("C:/Users/crist/Documents/crist_TFG/chip/countsK9_H3.RData")
boxplot(log(countsK9_H3))
```



```
gs<-sig_genes_pathways_k9
X<-countsK9_H3
gsva.es.k9 <- gsva(X, gs, verbose=TRUE)
```

- Heatmap (unsupervised clustering)

```
my_group<-c(1,1,1,3,3,3,3,3,3,2,2,2,2,2,2,2)
#my_group<-c(1,1,3,3,3,3,3,3,2,2,1,2,2,2,2,2)
colSide <- brewer.pal(9, "Set1")[my_group]
group_order<-c(10,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16)

#group_order<-c(2,1,3,4,5,6,7,8,15,11,10,16,13,14,9,12)

heatmap(gsva.es.k9, ColSideColors=colSide,main= "K9", Colv = group_order,cexRow = 0.5)
```


Fatty acid elongation
Riboflavin metabolism
Citrate cycle (TCA cycle)
Taurine and hypotaurine metabolism
Fatty acid degradation
Valine, leucine and isoleucine degradation
Glycolysis / Gluconeogenesis
Pyruvate metabolism
Pantothenate and CoA biosynthesis
Nitrogen metabolism
Alanine, aspartate and glutamate metabolism
Arginine biosynthesis
Valine, leucine and isoleucine biosynthesis
Aminoacyl-tRNA biosynthesis
Vitamin B6 metabolism
One carbon pool by folate
Phenylalanine, tyrosine and tryptophan biosynthesis
t4_k9
t3_k9
t6_k9
t5_k9
t7_k9
t8_k9
t9_k9
t10_k9
t15_k9
t2_k9
t1_k9
t16_k9
t13_k9
t14_k9
t11_k9
t12_k9