1. 制作KEGG通路图

从https://rest.kegg.jp/list/pathway/hsa获取人体的所有通路图的信息，将通路的hsa编码保存到pathway.txt文档中。如图1所示

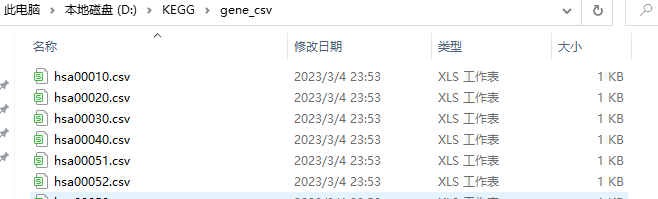
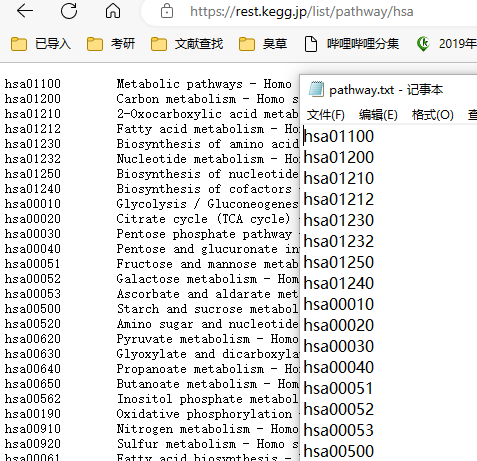


图1 获取通路的has编码 图2 使用代码保存的文件

在R中使用代码一，逐次获取每个通路图的所有基因名字，并保存到相应的通路名称的csv文件中，一共352个通路，如图2 所示。

在Python中使用代码二，整合所有的基因以及通路信息。所有的基因名字保存为final\_gene.csv文件，一共8200个基因。所有基因以及其所在的通路信息(8200个基因、340张包含基因的通路图)保存在gene\_path.csv中,具体的文件截图见图3和图4，并计算各个基因在通路上相似性。用以构建相似性网络，将相似性数据保存在path\_similarity.csv中，如图5所示

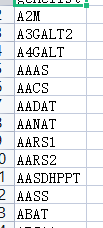
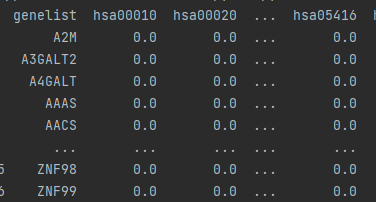
 

图3 final.csv部分文件截图 图4.gene\_path.csv数据截图

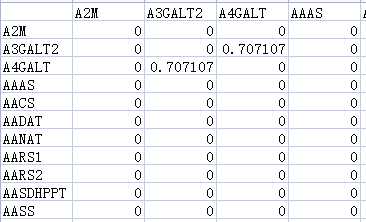


图5 path\_similarity.csv文件截图

代码一：

library("KEGGREST")

# 读取文件的内容

my\_lines <- readLines("D:\\KEGG\\pathway.txt")

# 计算文件的行数

num\_lines <- length(my\_lines)

# 输出结果

cat("The file contains", num\_lines, "lines.")

for (i in 1:num\_lines) {

path\_name <- my\_lines[i]

keggGet(path\_name)

gs<-keggGet(paste(path\_name))

#获取通路中gene信息

gs[[1]]$GENE

#查找所有基因

#genes<-unlist(lapply(gs[[1]]$GENE,function(x) strsplit(x,';')))

genes <- unlist(lapply(gs[[1]]$GENE, function(x) {

s <- strsplit(trimws(x), "; ")[[1]]

if (length(s) == 2) {

return(s)

} else {

print(paste0("Skipping element: ", x))

return(NULL)

}

}))

genelist <- genes[1:length(genes)%%2 ==1]

genelist <- data.frame(genelist)

#把结果写入表格中

path<-paste("D:\\KEGG\\gene\_csv\\" ,path\_name,".csv", sep ="")

write.table(genelist, path,

row.names=FALSE,col.names=TRUE,sep=",")

}

代码二：

import pandas as pd

import os

os.chdir(r"D:\KEGG\gene\_csv")

path\_list=os.listdir()

import numpy as np

def get\_genelist():

path\_list = os.listdir()

final\_gene = pd.DataFrame(columns=["genelist"])

for path in path\_list :

gene\_list =pd.read\_csv(os.path.join(path))

if gene\_list.empty:

continue

else:

final\_gene=pd.concat([final\_gene,gene\_list],axis=0)

final\_gene.drop\_duplicates("genelist",inplace=True)

final\_gene=final\_gene.sort\_values(by="genelist")

final\_gene.to\_csv(r"D:\桌面文件\代码\KEGG\final\_gene.csv",index=False)

return final\_gene

def get\_matrix():

final\_gene=get\_genelist()

path\_list = os.listdir()

for path in path\_list :

gene\_list =pd.read\_csv(os.path.join(path))

if gene\_list.empty:

continue

else:

tag = np.ones(shape=gene\_list.shape[0])

gene\_list[path[:8]]=pd.Series(tag)

final\_gene=pd.merge(final\_gene,gene\_list,how="left",on="genelist")

final\_gene.drop\_duplicates("genelist", inplace=True)

final\_gene=final\_gene.reset\_index(drop=True)

final\_gene.fillna(0,inplace=True)

final\_gene.to\_csv(r"D:\桌面文件\代码\KEGG\gene\_path.csv",index=False)

print(final\_gene)

print(final\_gene.shape)

A=final\_gene.iloc[:,1:].to\_numpy()

norm = np.linalg.norm(A, axis=1)

norm = norm.reshape(len(norm),1)

A = A / norm

A = np.matmul(A, A.T)

np.fill\_diagonal(A, 0)

path\_similarity = pd.DataFrame(A,index=final\_gene["genelist"].to\_numpy(),columns=final\_gene["genelist"].to\_numpy())

path\_similarity.to\_csv(r"D:\桌面文件\代码\KEGG\path\_similarity.csv",index=True)

print(path\_similarity)

get\_matrix()