R Notebook

#Universal mammalian clock2 of age # #author for developing clock2: Ake T. Lu #author for developing clock1 and clock3: Zhe Fei #

rm(list=ls())  
options(stringsAsFactors = F)  
setwd('/Users/alu/Desktop')  
myinput.list=readRDS('mydata\_GitHub.Rds')

# The following labels the coefficient values of the three universal clocks

beta.name=c('beta\_clock1','beta\_clock2','beta\_clock3')  
y.name=c('Y.pred1','Y.pred2','Y.pred3')  
age.name=c('DNAmAgePanMammalianClock1','DNAmAgePanMammalianClock2','DNAmAgePanMammalianClock3')

#clock2

F2\_antitrans\_clock2<-function(y,y.maxAge,y.gestation,const=1){  
 x0=const\*exp(-exp(-1\*y))  
 x1=x0\*(y.maxAge+y.gestation)  
 x=x1-y.gestation  
 x  
}

#clock3 #

F1\_logli <- function(age1, m1, m2 = m1, c1=1){  
 ifelse(age1 >= m1, (age1-m1)/m2 , c1\*log((age1-m1)/m2/c1 +1) )  
}  
#RelativeAdultAge  
F2\_revtrsf\_clock3 <- function(y.pred, m1, m2 = m1, c1=1){  
 ifelse(y.pred<0, (exp(y.pred/c1)-1)\*m2\*c1 + m1, y.pred\*m2+m1 )  
}

# The loglifn function shows how to calculate m1 for the transformation

# It is the a\_Logli in the function

F3\_loglifn = function(dat1,b1=1,max\_tage = 4,  
 c1=5, c2 = 0.38, c0=0){  
 n=nrow(dat1)  
   
 age1 = (dat1$maxAge+dat1$GestationTimeInYears)/(dat1$averagedMaturity.yrs+dat1$GestationTimeInYears)  
   
 a1 = age1/(1+max\_tage)  
 dat1$a1\_Logli = a1 #x/m1 in manuscript  
   
 a2 = (dat1$GestationTimeInYears + c0)/(dat1$averagedMaturity.yrs)   
 dat1$a\_Logli = a\_Logli = c1\*a2^c2  
 #m=5\*(G/ASM)^0.38 from regression analysis/formula(7)  
   
   
 x = dat1$Age + dat1$GestationTimeInYears  
 t2 = dat1$averagedMaturity.yrs\*b1 + dat1$GestationTimeInYears  
 x2 = x/t2 #### log(x/t2)  
 y = F1\_logli(x2, a\_Logli, a\_Logli)  
   
 dat1$LogliAge <- y  
 return(dat1)  
}

#(1)generate variable HighmaxAge

names(myinput.list)

## [1] "SampleAnnotation" "meth\_betas"   
## [3] "anAge" "ThreeUniversalPanMammalianClocks"

info=myinput.list[[1]]#The only required variables are SpeciesLatinName and Basename   
anage=myinput.list[[3]]  
info=merge(by='SpeciesLatinName',info,subset(anage,select=c(SpeciesLatinName,GestationTimeInYears,  
 averagedMaturity.yrs,maxAge)))  
head(info)

## SpeciesLatinName Basename MammalNumberHorvath SpeciesCommonName  
## 1 Tursiops truncatus 202894750036\_R05C02 4.13.6 Bottlenose dolphin  
## 2 Tursiops truncatus 202897220027\_R01C02 4.13.6 Bottlenose dolphin  
## 3 Tursiops truncatus 202897220027\_R02C02 4.13.6 Bottlenose dolphin  
## 4 Tursiops truncatus 202897220027\_R06C01 4.13.6 Bottlenose dolphin  
## 5 Tursiops truncatus 202897220040\_R02C01 4.13.6 Bottlenose dolphin  
## 6 Tursiops truncatus 202897220040\_R06C01 4.13.6 Bottlenose dolphin  
## Age Female Tissue GestationTimeInYears averagedMaturity.yrs maxAge  
## 1 11.42 0 Blood 1.030137 8.931507 67  
## 2 15.44 1 Blood 1.030137 8.931507 67  
## 3 19.30 1 Blood 1.030137 8.931507 67  
## 4 18.23 1 Blood 1.030137 8.931507 67  
## 5 14.66 1 Blood 1.030137 8.931507 67  
## 6 28.27 0 Blood 1.030137 8.931507 67

#Description for mymax=1.3 #We were concerned that the uneven evidence surrounding the maximum age of different species #could bias our analysis. While billions of people have been evaluated for estimating #the maximum age of humans (122.5 years) or mice (4 years), #the same cannot be said for any other species. #To address this concern, we made the following assumption: #the true maximum age is 30% higher than that reported in AnAge #for all species except for humans and mice (Mus musculus). #Therefore, we multiplied the reported maximum lifespan of non-human or non-mouse species by 1.3. #Our predictive models turn out to be highly robust with respect to this assumption.

MYMAX=1.3  
info$HighmaxAge=MYMAX\*info$maxAge  
info$HighmaxAge[info$SpeciesLatinName=='Homo sapiens']=info$maxAge[info$SpeciesLatinName=='Homo sapiens']  
info$HighmaxAge[info$SpeciesLatinName=='Mus musculus']=info$maxAge[info$SpeciesLatinName=='Mus musculus']

#(2)merge info and metharray beta values

glmnet.list=myinput.list[[4]]#The three universal clock prediction models  
mycpgs=c(glmnet.list[[1]]$var,glmnet.list[[2]]$var,glmnet.list[[3]]$var)  
mycpgs=unique(mycpgs)  
mycpgs=mycpgs[mycpgs!='Intercept']  
  
#dat.meth0: number of Mammalian array CpGs (n=37554) x [number of samples+1]  
#  
dat.meth0=myinput.list[[2]]  
dat.meth0=subset(dat.meth0,CGid%in%mycpgs)#only keep the CpGs in the three clocks  
dat.meth=t(dat.meth0[,-c(1)])  
colnames(dat.meth)=dat.meth0$CGid  
dat.meth=data.frame(Basename=colnames(dat.meth0)[-c(1)],dat.meth)  
dat.meth$Intercept=1  
#  
info=merge(by='Basename',info,dat.meth)  
#  
#predict RelativeAge  
#  
  
for(k in 1:3){  
glmnet=glmnet.list[[k]]  
glmnet$beta=glmnet[,beta.name[k]]  
#glmnet$var[1]=ifelse(glmnet$var[1]=="(Intercept)",'Intercept',glmnet$var[1])  
temp=as.matrix(subset(info,select=as.character(glmnet$var)))  
info[,y.name[k]]=as.numeric(as.matrix(subset(info,select=as.character(glmnet$var)))%\*%glmnet$beta)  
}

#(1) Clock 1

info[,age.name[1]]=exp(info[,y.name[k]])-2

#(2) Clock 2

info$DNAmRelativeAge=exp(-exp(-1\*info[,y.name[2]]))  
info[,age.name[2]]= F2\_antitrans\_clock2(info[,y.name[2]],info$HighmaxAge,info$GestationTimeInYears,const=1)

#(3) Clock 3

info=F3\_loglifn(info)#to compute m estimate for tuning point in the log-linear transformation  
info$m1=info$a\_Logli  
info$DNAmRelativeAdultAge=F2\_revtrsf\_clock3(info[,y.name[3]], info$m1)  
info[,age.name[3]]<-  
 info$DNAmRelativeAdultAge \*(info$averagedMaturity.yrs + info$GestationTimeInYears) -info$GestationTimeInYears

#output

output=subset(info,select=c('Basename','SpeciesLatinName','MammalNumberHorvath','Age','Tissue','DNAmRelativeAge','DNAmRelativeAdultAge',age.name))

#plot

x=output$Age  
y=output$DNAmAgePanMammalianClock2  
cor0=round(cor(x,y),2)  
mae0=round(median(abs(x-y)),2)  
plot(x,y,xlab='Age',ylab='Clock2: DNAm Age',  
 main=paste0('Bottlenose dolphin (n=',length(x),')\n','cor=',cor0,  
 ', MAE=',mae0),  
 col='white',xlim=c(0,max(max(x),max(y))+1),  
 ylim=c(0,max(max(x),max(y))+1))  
text(x,y,label=output$MammalNumberHorvath,col='blue')  
abline(0,1,lwd=2,lty=2,col="grey")  
model1=lm(y~x)  
abline(model1,lwd=2,lty=2,col='black')

