Age Prediction

by Liu Jie

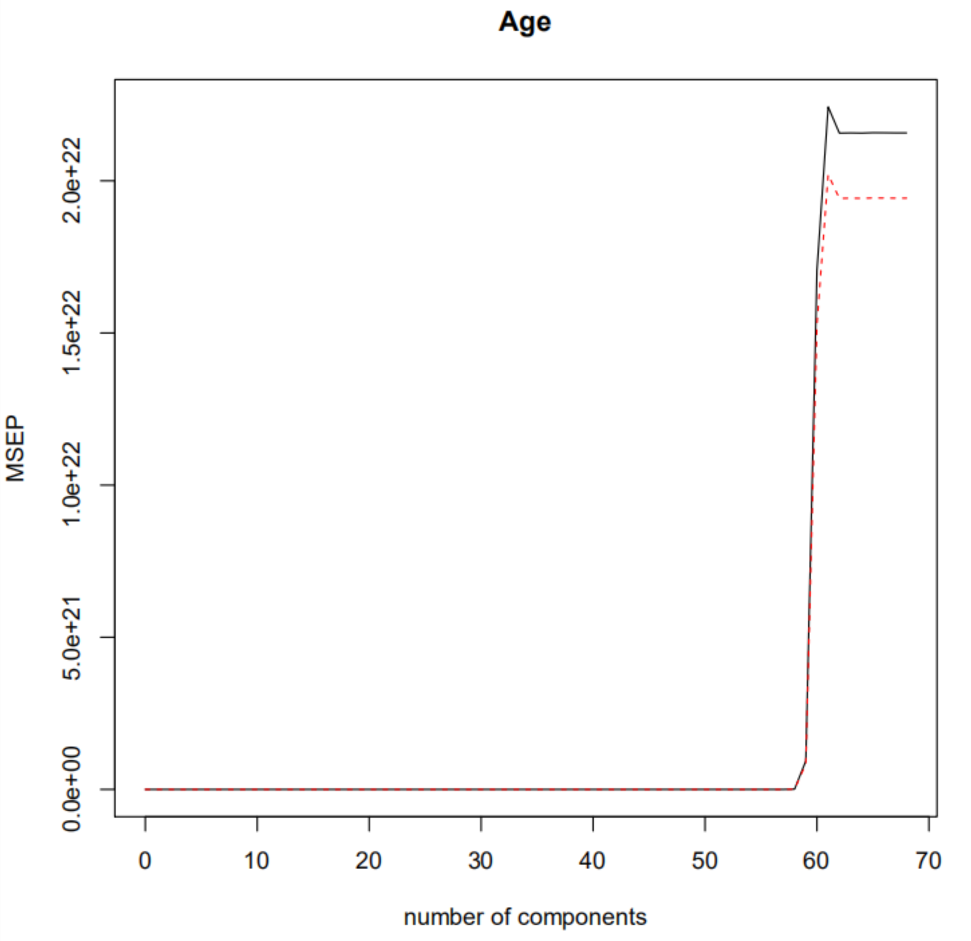
summary

- I use pls(*Partial Least Squares* **偏最小二乘法**) for prediction
- main package and function: `pls::plsr()`
- Batch effect: use 2 dummy variables for 3 levels of batch

R Script

```r
# data pre-process
sample_inform <- read.table("/public/home/liuj626/R_homework/sample_inform.txt",
                            header = T)
ref_counts <-
read.table("/public/home/liuj626/R_homework/RNAseq_hg19_refGene_counts.txt")

rownames(ref_counts) <- ref_counts[,1]
ref_counts <- ref_counts[,-1]
colnames(ref_counts) <- 1:91

# NA rmove
ref_counts <- na.omit(ref_counts)
ref_counts <- as.data.frame(t(ref_counts))

Id_NA <- rownames(sample_inform[!complete.cases(sample_inform),])
Id_NA <- as.integer(Id_NA)

train <- is.na(sample_inform$Age)
train <- !train
age_train <- sample_inform$Age[train]

ref_counts_1 <- cbind(ref_counts,sample_inform$Age)
colnames(ref_counts_2)[17321] <- c("Age")

# batch effect:
batch1 <- rep(0,91)
batch1[which(sample_inform$Batch==1)] <- 1
batch2 <- rep(0,91)
batch2[which(sample_inform$Batch==2)] <- 1

ref_count_batch=cbind(ref_counts_1,batch1=batch1,batch2=batch2)

############## pls

library(pls)
set.seed(1)
pls.fit <- plsr(Age~.,data=ref_count_batch,subset=train,
                scale=T,validation="CV")
summary(pls.fit)

pdf("plsfit_plot01.pdf")
validationplot(pls.fit,val.type = "MSEP")
dev.off()
```

- As we are unsure about the best parameter $ncomp$, which means the number of components we use in model for prediction, we use **cross-validation** on the training dataset for the best one.
- From the output we can say when $ncomp$=9~63, the model has almost the same effect on training set.
- So we choose $ncomp = 9$ for prediction model as it is the simplest.



```
# To predict
pls.pred <- predict(pls.fit,ref_count_batch[!train,-17321],ncomp=9)
pls.pred

pls.fit.9 <- plsr(Age~.,data=ref_count_batch,subset=train,
                  scale=T,ncomp=9)
summary(pls.fit.9)
```