Demonstration of multivariate multiple linear regression in cdr3-QTL analysis

@Kaz

Contents

STEP1: Read in dataframe 'M'

- M includes the following data:
- 1, genotype of HLA DRB1 site13
- 2, amino acid frequencies at position 109 of CDR3 (L=13)
- 3, covariates of genotype PC1-3

```
load("example_data_HLA_DRB1_site13_L13P109.RData")
head(M,n=3)
```

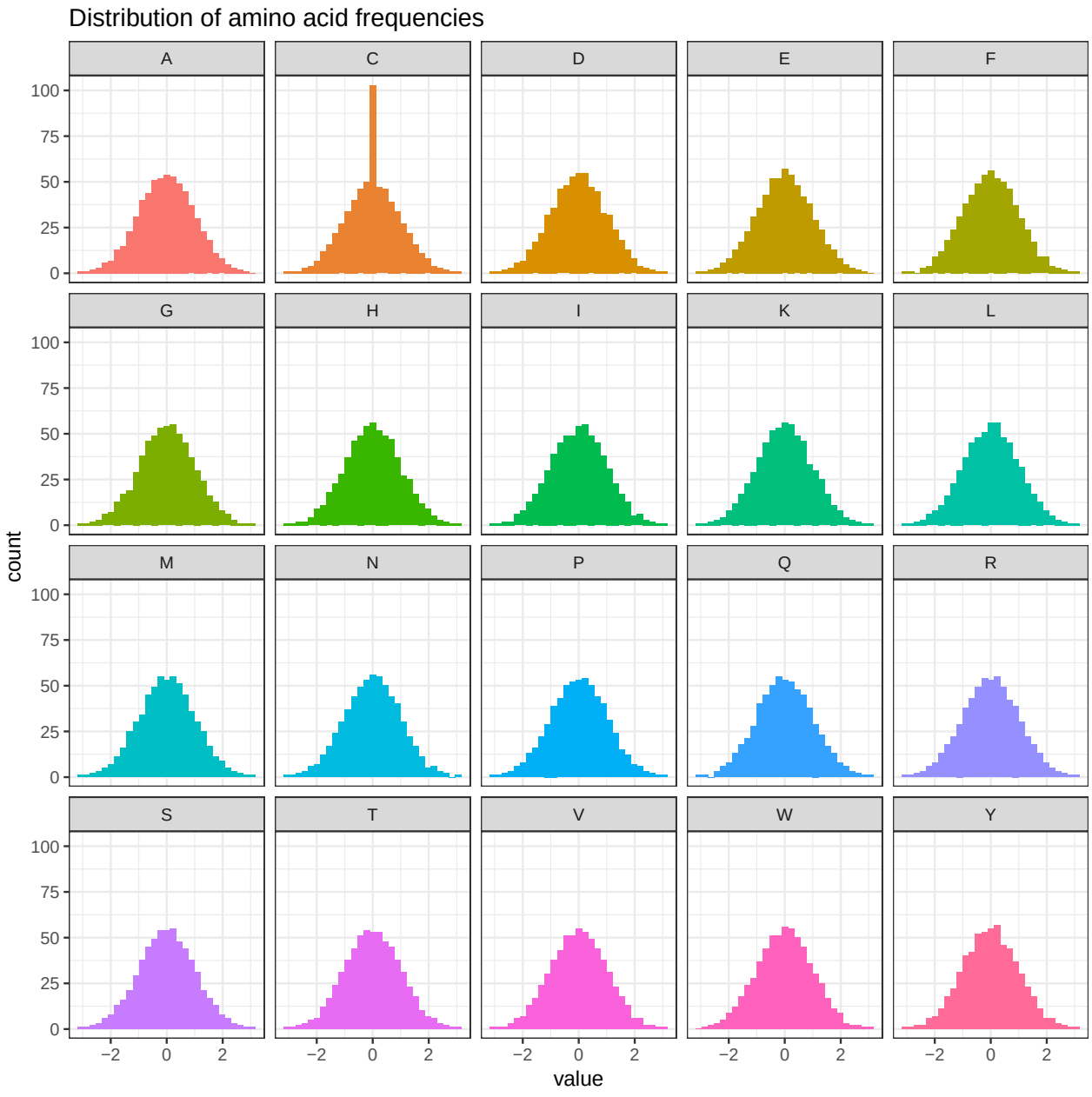
```
##     Sample dose1 dose2 dose3 dose4 dose5           A         C         D
## 1 HIP00110     0     0     2     0     0  0.64640359  1.952296 -1.201055
## 2 HIP00169     0     0     2     0     0 -0.09612232 -0.709793  0.284175
## 3 HIP00594     0     0     1     1     0  0.35148406  1.396508 -1.119885
##            E          F          G          H          I        K
## 1 -1.9385796 -0.1704431  1.2910201 -1.2079469 -1.0989915 1.118958
## 2 -2.1502995  1.5227569 -1.2010554  1.7351922  3.1743932 1.290168
## 3 -0.5782978  0.6591890  0.3554904  0.2669941  0.9644211 1.084382
##           L         M          N          P           Q         R
## 1 -1.3732660 0.1934373 -0.9465976 -0.2258458 -1.17064440 1.4660207
## 2  0.9004744 1.4542945  1.1706444 -1.3928549 -0.84001335 0.9061341
## 3 -0.5084881 1.6709232 -1.2088337 -0.8561914  0.01693749 0.5343591
##             S           T           V          W          Y        PC1
## 1  0.41221727 -0.06591255  1.24887250  1.6128969  0.2841750 -0.3715639
## 2 -0.08100819  0.09612232  3.17439317 -1.2901676 -1.4771343 -0.8574693
## 3  0.38775385  0.20273862 -0.07723269  0.4647573  0.1797391  0.3055545
##         PC2        PC3
## 1 0.1614687  0.9765430
## 2 1.4323182 -0.9377073
## 3 0.6570370 -1.2958039
```

```
#distribution of amino acid frequencies: already normalized into standard normal distribution
library(reshape)
library(ggplot2)
library(magrittr)

df <- M[,c("A","C","D","E","F","G","H","I","K", "L","M","N","P","Q","R","S","T","V","W","Y")]
df <- melt(df)
```

```
## Using  as id variables
```

```
df %>% ggplot(aes(x=value,fill=variable)) +
  geom_histogram(bins = 30) +
  facet_wrap(~variable) +
  theme_bw() +
  theme(legend.position = "none") +
  labs(title="Distribution of amino acid frequencies")
```



STEP2: Estimate P value using R function anova.mlm

- This is the way I calculate P value in the manuscript (Pillai statistics in MANOVA)

```
#full model (with dose1-5; although HLA-DRB1 site 13 has six possible amino acid, I excluded one as ref
mod1 <- lm( cbind(A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y) ~
              dose1+dose2+dose3+dose4+dose5+
```

```
              PC1+PC2+PC3, data = M)
#null model (no dose1-5 terms)
mod0 <- lm( cbind(A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y) ~
              PC1+PC2+PC3, data = M)

test <- anova(mod1, mod0)
test
```

```
## Analysis of Variance Table
##
## Model 1: cbind(A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W,
##     Y) ~ dose1 + dose2 + dose3 + dose4 + dose5 + PC1 + PC2 +
##     PC3
## Model 2: cbind(A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W,
##     Y) ~ PC1 + PC2 + PC3
##   Res.Df Df Gen.var. Pillai approx F num Df den Df    Pr(>F)
## 1    619     0.59673
## 2    624  5  0.64388 1.3286   10.929    100   3020 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test$`Pr(>F)`[2] #pvalue: 4.17218e-138; this is the p value we reported.
```

```
## [1] 4.17218e-138
```

STEP3: Estimate P value using custom script

- Successfully reproduced the same statistics as in STEP2.
- Useful online materials:
- https://online.stat.psu.edu/stat505/lesson/8
- https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_introreg_sect012.htm

```
mod1 <- lm( cbind(A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y) ~
              dose1+dose2+dose3+dose4+dose5+
              PC1+PC2+PC3, data = M)

mod0 <- lm( cbind(A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y) ~
              PC1+PC2+PC3, data = M)

#response variable matrix
Y <- M[,c("A","C","D","E","F","G","H","I","K", "L","M","N","P","Q","R","S","T","V","W","Y")]
Y <- as.matrix(Y)
dim(Y)
```

```
## [1] 628  20
```

```
Y[1:5,1:5]
```

```
##                A         C          D          E          F
## [1,]  0.64640359  1.952296 -1.2010554 -1.9385796 -0.1704431
## [2,] -0.09612232 -0.709793  0.2841750 -2.1502995  1.5227569
## [3,]  0.35148406  1.396508 -1.1198850 -0.5782978  0.6591890
## [4,]  0.57385479 -1.515204  0.2374441 -0.1759155 -0.1399218
## [5,]  2.84048462  1.042438 -2.6734518 -1.1269591 -0.9938400
```

```r
#degree of freedom of full and null model
DF_full <- mod1$df.residual
DF_null <- mod0$df.residual

#matrix of residuals
Res_full <- Y - mod1$fitted.values
Res_null <- Y - mod0$fitted.values
dim(Res_full)
```

```
## [1] 628  20
```

```r
dim(Res_null)
```

```
## [1] 628  20
```

```r
Emat <- crossprod(Res_full)
Hmat <- crossprod(Res_null) - crossprod(Res_full)
dim(Emat)
```

```
## [1] 20 20
```

```r
dim(Hmat)
```

```
## [1] 20 20
```

```r
Pillai <- sum(diag( Hmat %*% solve( Hmat + Emat ) ))
Pillai #the identical value as R function (see above)
```

```
## [1] 1.328616
```

```r
p=20 #DF of Y matrix (N of amino acids)
q=5  #DF of X (dose1-dose5)
s=min(p,q)
v=DF_full
m=(abs(p-q)-1)/2
n=(v-p-1)/2

appF <- (2*n + s + 1)/(2*m + s + 1) * ( Pillai / (s - Pillai) )
appF #the identical value as R function (see above)
```

```
## [1] 10.9289
```

```r
numDF <- s*(2*n + s + 1)
numDF #the identical value as R function (see above)
```

```
## [1] 3020
```

```r
dnomDF <- s*(2*m + s + 1)
dnomDF #the identical value as R function (see above)
```

```
## [1] 100
```

```r
pf(appF, dnomDF, numDF, lower.tail = FALSE) #the identical value as R function (see above)
```

```
## [1] 4.17218e-138
```

STEP4: Variance explained using R function MVLM

```r
library(MVLM)
```

```r
#variance explained by full model
full.res <- mvlm( cbind(A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y) ~
              dose1+dose2+dose3+dose4+dose5 +
              PC1+PC2+PC3,
              data = M)
full <- full.res$pseudo.rsq["Omnibus Effect",1]

#variance explained by null model
null.res <- mvlm( cbind(A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y) ~
              PC1+PC2+PC3,
              data = M)
null <- null.res$pseudo.rsq["Omnibus Effect",1]

#explained variance by dose1-5: this value was reported in the manuscript
full - null
```

```
## Omnibus Effect
##     0.09415378
```

STEP5-1: Variance explained using the custom script (matrix multiplication)

- Successfully reproduced the same statistics as in STEP4.

```r
Y <- M[,c("A","C","D","E","F","G","H","I","K", "L","M","N","P","Q","R","S","T","V","W","Y")]
Y <- as.matrix(Y)
dim(Y)
```

```
## [1] 628  20
```

```r
Y[1:4,1:4]
```

```
##                A         C          D          E
## [1,]  0.64640359  1.952296 -1.2010554 -1.9385796
## [2,] -0.09612232 -0.709793  0.2841750 -2.1502995
## [3,]  0.35148406  1.396508 -1.1198850 -0.5782978
## [4,]  0.57385479 -1.515204  0.2374441 -0.1759155
```

```r
#full model
X <- M[,c("dose1","dose2","dose3","dose4","dose5","PC1","PC2","PC3")]
X$Intercept <- 1
X <- as.matrix(X)
dim(X)
```

```
## [1] 628   9
```

```r
X[1:4,]
```

```
##      dose1 dose2 dose3 dose4 dose5        PC1        PC2        PC3
## [1,]     0     0     2     0     0 -0.3715639  0.1614687  0.9765430
## [2,]     0     0     2     0     0 -0.8574693  1.4323182 -0.9377073
## [3,]     0     0     1     1     0  0.3055545  0.6570370 -1.2958039
## [4,]     0     1     0     1     0  0.8394371 -1.1396599 -3.4511489
##      Intercept
## [1,]         1
## [2,]         1
## [3,]         1
## [4,]         1
```

```r
n <- nrow(X)
p <- ncol(X)
q <- ncol(Y)

H <- tcrossprod(tcrossprod(X, solve(crossprod(X))), X)
  # hat matrix
  # X %*% ( t(X) %*% X )^-1 %*% t(X)

mean.Y <- matrix(apply(Y, 2, mean), nrow = n, ncol = q, byrow = T)
sscp.mean.Y <- crossprod(mean.Y)
sscp.Y <- crossprod(Y)
sscp <- sscp.Y - sscp.mean.Y

sscp.r <- (crossprod(Y, H) %*% Y) - sscp.mean.Y

full <- sum(diag(sscp.r))/sum(diag(sscp)) # the identical value as above


#null model
X <- M[,c("PC1","PC2","PC3")]
X$Intercept <- 1
X <- as.matrix(X)
dim(X)
```

```
## [1] 628   4
```

```r
X[1:4,]
```

```
##             PC1        PC2        PC3 Intercept
## [1,] -0.3715639  0.1614687  0.9765430         1
## [2,] -0.8574693  1.4323182 -0.9377073         1
## [3,]  0.3055545  0.6570370 -1.2958039         1
## [4,]  0.8394371 -1.1396599 -3.4511489         1
```

```r
n <- nrow(X)
p <- ncol(X)
q <- ncol(Y)

H <- tcrossprod(tcrossprod(X, solve(crossprod(X))), X)
  # hat matrix
  # X %*% ( t(X) %*% X )^-1 %*% t(X)

mean.Y <- matrix(apply(Y, 2, mean), nrow = n, ncol = q, byrow = T)
sscp.mean.Y <- crossprod(mean.Y)
sscp.Y <- crossprod(Y)
sscp <- sscp.Y - sscp.mean.Y

sscp.r <- (crossprod(Y, H) %*% Y) - sscp.mean.Y

null <- sum(diag(sscp.r))/sum(diag(sscp)) # the identical value as above

#explained variance by dose1-5: the same value as above results with MVLM package
full - null
```

```
## [1] 0.09415378
```