EXPECTATION_MAXIMISATION_HAPLOTYPE ESTIMATIONS

KIRABO GLORIA

2023-05-19

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.2.3
```

```
library(haplo.stats)
```

```
## Warning: package 'haplo.stats' was built under R version 4.2.3
```

```
## Loading required package: arsenal
```

```
## Warning: package 'arsenal' was built under R version 4.2.3
```

```
library(dplyr)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:arsenal':
##
##     set_attr
##
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(genetics)
```

```
## Warning: package 'genetics' was built under R version 4.2.3
```

```
## Loading required package: combinat
##
## Attaching package: 'combinat'
##
## The following object is masked from 'package:utils':
##
##     combn
##
## Loading required package: gdata
```

```
## Warning in system(cmd, intern = intern, wait = wait | intern,
## show.output.on.console = wait, : running command 'C:\WINDOWS\system32\cmd.exe /c
## ftype perl' had status 2
```

```
## Warning in system(cmd, intern = intern, wait = wait | intern,
## show.output.on.console = wait, : running command 'C:\WINDOWS\system32\cmd.exe /c
## ftype perl' had status 2
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLSX' (Excel 2007+) files.
##
## gdata: Run the function 'installXLSXsupport()'
## gdata: to automatically download and install the perl
## gdata: libaries needed to support Excel XLS and XLSX formats.
##
## Attaching package: 'gdata'
##
## The following objects are masked from 'package:dplyr':
##
##     combine, first, last
##
## The following object is masked from 'package:purrr':
##
##     keep
##
## The following object is masked from 'package:stats':
##
##     nobs
```

```
##
## The following object is masked from 'package:utils':
##
##     object.size
##
## The following object is masked from 'package:base':
##
##     startsWith
##
## Loading required package: gtools

## Warning: package 'gtools' was built under R version 4.2.3

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.2.3

##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
## Loading required package: mvtnorm
##
##
## NOTE: THIS PACKAGE IS NOW OBSOLETE.
##
##
##
##   The R-Genetics project has developed an set of enhanced genetics
##
##   packages to replace 'genetics'. Please visit the project homepage
##
##   at http://rgenetics.org for informtion.
##
##
##
##
## Attaching package: 'genetics'
##
## The following object is masked from 'package:haplo.stats':
##
##     locus
##
## The following objects are masked from 'package:base':
##
##     %in%, as.factor, order
```

```
##STEP ONE-READING IN THE DATASET FROM THE LINK
fams <- read.delim("http://www.biostat.umn.edu/~cavanr/FMS_data.txt",header = T,
                sep = "\t")
```

```
#Vieing the first five rows and 10columns of the dataset

fams[1:5,1:10]
```

```
##        id acdc_rs1501299 ace_id actn3_r577x actn3_rs540874 actn3_rs1815739
## 1 FA-1801             CA     DD          CC             GG              CC
## 2 FA-1802             CA     ID          CT             GA              TC
## 3 FA-1803             CA     ID          CT             GA              TC
## 4 FA-1804             CC     DD          CT             GA              TC
## 5 FA-1805             CA     ID          CC             GG              CC
##   actn3_1671064 ardb1_1801253 adrb2_1042713 adrb2_1042714
## 1            AA          <NA>            GA            CG
## 2            GA          <NA>            GA            CC
## 3            GA          <NA>            GA            CG
## 4            GA          <NA>            AA            CC
## 5            AA          <NA>            GA            CG
```

```
##SELECTING OUT COLUMNS WITH RESISTIN GENES

fams_actn3 <- fams[grepl("^actn3", names(fams))]
head(fams_actn3)
```

```
##   actn3_r577x actn3_rs540874 actn3_rs1815739 actn3_1671064
## 1          CC             GG              CC            AA
## 2          CT             GA              TC            GA
## 3          CT             GA              TC            GA
## 4          CT             GA              TC            GA
## 5          CC             GG              CC            AA
## 6          CT             GA              TC            GA
```

```
##Counting the number of generated columns

actn3_col <- ncol(fams_actn3)
cat("The columns with actn3 gene are:", actn3_col)
```

```
## The columns with actn3 gene are: 4
```

```
##IDENTIFYING UNIQUE SNPS IN THE COLUMNS

##STEP ONE CREATE A VECTOR
actn3_snp <- unlist(fams_actn3)
head(actn3_snp)
```

```
## actn3_r577x1 actn3_r577x2 actn3_r577x3 actn3_r577x4 actn3_r577x5 actn3_r577x6
##         "CC"         "CT"         "CT"         "CT"         "CC"         "CT"
```

```
##STEP TWO:IDENTIFYING UNIQUE VALUES IN THE VCECTOR
num_snps_actn3 <- length(unique(actn3_snp))
cat("The total number of snps in the restitin genes is :",num_snps_actn3)
```

```
## The total number of snps in the restitin genes is : 8
```

```
##CREATING A GENOTYPE OBJECT FOR OUR GENES OF THE RESISTIN GENE

geno_actn3 <- as.data.frame(lapply(fams_actn3, genotype, sep=""))
geno_actn3[1:5,]
```

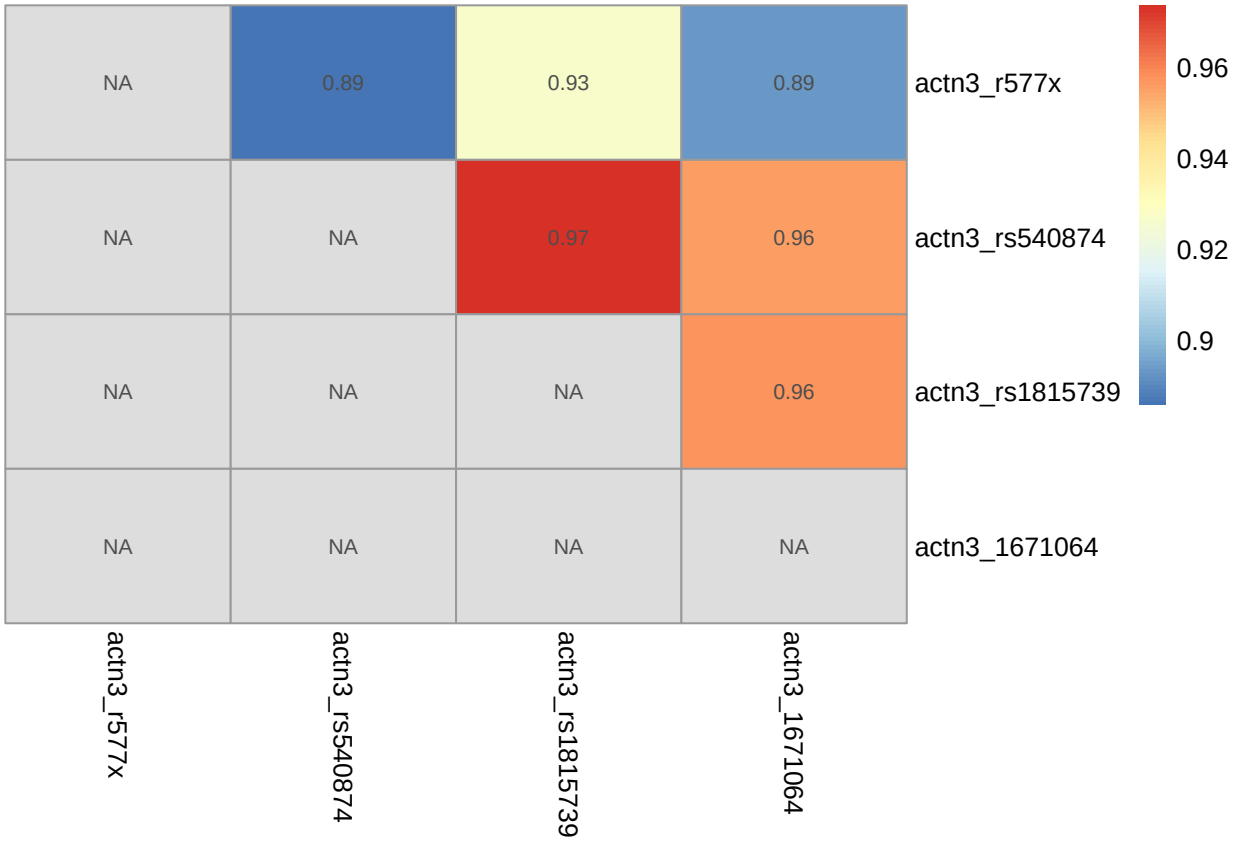
```
##   actn3_r577x actn3_rs540874 actn3_rs1815739 actn3_1671064
## 1         C/C            G/G             C/C           A/A
## 2         C/T            G/A             C/T           A/G
## 3         C/T            G/A             C/T           A/G
## 4         C/T            G/A             C/T           A/G
## 5         C/C            G/G             C/C           A/A
```

```
##Calculating D' OF THE SNPS IN THE RESISTIN GENE

actn3_D <- LD(geno_actn3)$`D'`
actn3_D
```

```
##                 actn3_r577x actn3_rs540874 actn3_rs1815739 actn3_1671064
## actn3_r577x              NA      0.8858385       0.9266828     0.8932708
## actn3_rs540874           NA             NA       0.9737162     0.9556019
## actn3_rs1815739          NA             NA              NA     0.9575870
## actn3_1671064            NA             NA              NA            NA
```

```
##GENERATING A HEATMAP TO SHOW THE D' OF SNPS
pheatmap(actn3_D, cluster_cols = FALSE, cluster_rows = FALSE, display_numbers = TRUE)
```



```
##From the heatmap, it shows that the SNP at actn3_rs540874 is in high LD with actn3_rs1815739




###CALCULATING OF HARDY WEINBERG EQUILLIBRIUM(USING CHISQUARE AND FISCHER'S EXACT TEST)

#chi_pval <- vector() #create empty vector

#for (col in geno_actn3){
  #chiq <- HWE.chisq(col)
  #chi_pval <- c(chi_pval,chisq$p.value)
#}
#sort(chi_pval)

##naming p-values
#names(chi_pval)=colnames(geno_actn3) ##assigning names with those in the genotype object
 #sort(chi_pval)

#sum(chi_pval<0.05)##finding total pvalues <0.5

#names(chi_pval[chi_pval<0.05])##getting names of columns with pvalues < 0.5


##CALCULATING THE FISCHER'S EXACT
ext_pval <- vector()

for (col in geno_actn3){
  exact <- HWE.exact(col)
  ext_pval <- c(ext_pval,exact$p.value)
}
sort(ext_pval)
```

```
## [1] 0.000293572 0.727001248 0.953356704 0.953375084
```

```
sum(ext_pval<0.05)
```

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

```
names(ext_pval)=colnames(geno_actn3)
sort(ext_pval)
```

```
##     actn3_r577x   actn3_1671064 actn3_rs1815739  actn3_rs540874
##     0.000293572     0.727001248     0.953356704     0.953375084
```

```
names(ext_pval[ext_pval<0.05])
```

```
## [1] "actn3_r577x"
```

```
##Adjusting using Bonferoni to cater for multiple testing

##Adjusting chi-square values
#set.seed(100)
#adj_pva <- p.adjust(chi_pval, method ="bonferroni")
#sort(adj_pva)
#sum(adj_pva<0.05)
#names(adj_pva[adj_pva<0.05])

##Adjusting exact p-values
set.seed(42)#for reproducibility
aj_val <- p.adjust(ext_pval,method = "bonferroni")
sort(aj_val)
```

```
##     actn3_r577x  actn3_rs540874 actn3_rs1815739   actn3_1671064
##     0.001174288     1.000000000     1.000000000     1.000000000
```

```
sum(aj_val<0.05)
```

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

```
names(aj_val[aj_val<0.05])
```

```
## [1] "actn3_r577x"
```

```
##CALCULATING MINOR ALLELE FREQUENCY(MAF) OF SNPS

##We first identify missing values

miss_gen <- data.frame(summary(is.na(geno_actn3))[3,])

names(miss_gen) <- c("Missing values")
miss_gen
```

```
##                  Missing values
## actn3_r577x     TRUE :662
## actn3_rs540874  TRUE :181
## actn3_rs1815739 TRUE :180
## actn3_1671064   TRUE :176
```

```
##Regardless, we take it that our allele frequencies will remain the same even if we had got the missing


##Calculating the MINOR ALLELE FREQUENCY(MAF) FOR AT EACH SNP

round(summary(geno_actn3$actn3_r577x)$"allele.freq",1)
```

```
##    Count Proportion
## C    750        0.5
## T    720        0.5
## NA  1324         NA
```

```
round(summary(geno_actn3$actn3_rs540874)$"allele.freq",1)
```

```
##    Count Proportion
## G   1385        0.6
## A   1047        0.4
## NA   362         NA
```

```
round(summary(geno_actn3$actn3_rs1815739)$"allele.freq",1)
```

```
##    Count Proportion
## C   1389        0.6
## T   1045        0.4
## NA   360         NA
```

```
##CALCULATION OF EXPECTATION MAXIMISATION FOR THE HAPLOTYPES

library(haplo.stats)## downloading the required package

locus.labels <- colnames(geno_actn3)##assigning locus names to match those in the genotype
locus.labels
```

```
## [1] "actn3_r577x"     "actn3_rs540874"  "actn3_rs1815739" "actn3_1671064"
```

```
Haplo.EM_actn3 <- haplo.em(geno_actn3,control = haplo.em.control(min.posterior = 1e-4))
```

```
## Warning in haplo.em(geno_actn3, control = haplo.em.control(min.posterior = 1e-04)): Subject(s) 240 r
##  Try decreasing min.posterior control parameter to reduce trimming.
```

```
Haplo.EM_actn3
```

```
## ================================================================================
##                                    Haplotypes
## ================================================================================
##    loc-1 loc-2 hap.freq
## 1    A/A   C/T  0.00189
## 2    A/A   G/G  0.00199
## 3    A/A   T/T  0.08881
## 4    C/C   A/A  0.13270
## 5    C/C   A/G  0.01686
## 6    C/C   G/G  0.00201
## 7    C/T   A/A  0.01063
## 8    C/T   A/G  0.20150
## 9    C/T   G/G  0.00055
## 10   G/A   C/C  0.00352
## 11   G/A   C/T  0.23953
## 12   G/A   T/T  0.00204
## 13   G/G   C/C  0.15890
## 14   G/G   C/T  0.00159
## 15   G/G   T/T  0.00121
## 16   T/T   A/A  0.01916
```

```
## 17   T/T   A/G  0.02628
## 18   T/T   G/G  0.09084
## ================================================================================
##                                     Details
## ================================================================================
## lnlike =  -3895.39
## lr stat for no LD =  3147.833 , df =  7 , p-val =  0
##
## Results may be incomplete because one or more subjects was removed
```

```
##determing the structure of the created object
str(Haplo.EM_actn3)
```

```
## List of 18
##  $ lnlike       : num -3895
##  $ lnlike.noLD  : num -5469
##  $ lr           : num 3148
##  $ df.lr        : num 7
##  $ hap.prob     : num [1:18] 0.00189 0.00199 0.08881 0.1327 0.01686 ...
##  $ hap.prob.noLD: 'table' num [1:18(1d)] 0.0575 0.0225 0.0217 0.0364 0.0548 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ : chr [1:18] "1" "1" "1" "2" ...
##  $ converge     : int 1
##  $ locus.label  : chr [1:2] "loc-1" "loc-2"
##  $ indx.subj    : num [1:15892] 1 2 3 4 5 6 7 8 9 10 ...
##  $ subj.id      : int [1:15892] 1 2 3 4 5 6 7 8 9 10 ...
##  $ post         : num [1:15892] 1 1 1 1 1 1 1 1 1 1 ...
##  $ hap1code     : num [1:15892] 4 11 11 11 4 8 18 8 11 13 ...
##  $ hap2code     : num [1:15892] 13 8 8 8 13 11 3 11 8 4 ...
##  $ haplotype    :'data.frame':   18 obs. of  2 variables:
##   ..$ loc-1: 'AsIs' chr [1:18] "A/A" "A/A" "A/A" "C/C" ...
##   ..$ loc-2: 'AsIs' chr [1:18] "C/T" "G/G" "T/T" "A/A" ...
##  $ nreps        : 'table' int [1:1396(1d)] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ indx.subj: chr [1:1396] "1" "2" "3" "4" ...
##  $ rows.rem     : num 240
##  $ max.pairs    : num [1:1397] 2 2 2 2 2 2 2 2 2 2 ...
##  $ control      :List of 10
##   ..$ loci.insert.order: int [1:2] 1 2
##   ..$ insert.batch.size: num 2
##   ..$ min.posterior    : num 1e-04
##   ..$ tol              : num 1e-05
##   ..$ max.iter         : num 5000
##   ..$ random.start     : num 0
##   ..$ n.try            : num 10
##   ..$ iseed            : int [1:626] 10403 1 -1577024373 1699409082 1745430460 -928819969 -175402385
##   ..$ max.haps.limit   : num 2e+06
##   ..$ verbose          : num 0
##  - attr(*, "class")= chr "haplo.em"
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.