```
#Cài đặt chương trình Rvà Rstudio: Download the RStudio IDE - RStudio
#Cài đặt package R/qtl
install.packages("qtl")
library(qtl)



###################
# Nhập dữ liệu
# Tóm tắt thông tin về số liệu
###################

#Nhập các bộ thông tin vào phần mềm (dữ liệu kiểu gene, dữ liệu kiểu hình)
#Thông thường, việc nhập số liệu vào phần mềm R được thực hiện bằng lệnh: read.cross command
#e.g.
mapthis <- read.cross("csv", "https://rqtl.org/tutorials", "mapthis.csv",estimate.map=FALSE)
# hoặc
mapthis <- read.cross("csv", , file.choose(), estimate.map=FALSE)
#hoặc
data(mapthis)

summary(mapthis)

#############################
# LOẠI BỎCÁC CHỈ THỊ VÀ CÁ THỂCÓ NHIỀU SỐLIỆU BỊ THIẾU#
#############################
plotMissing(mapthis)

par(mfrow=c(1,2), las=1)
plot(ntyped(mapthis), ylab="No. typed markers", main="No. genotypes by individual")
plot(ntyped(mapthis, "mar"), ylab="No. typed individuals", main="No. genotypes by marker")

#Loại bỏ các cá thể có ít hơn 51 kết quả phân tích từcác chỉ thị phân tử
mapthis <- subset(mapthis, ind=(ntyped(mapthis)>50))

#Loại bỏ các chỉ thị có kết quả phân tích của ít hơn 200 cá thể.
nt.bymar <- ntyped(mapthis, "mar")
todrop <- names(nt.bymar[nt.bymar < 200])
mapthis <- drop.markers(mapthis, todrop)

#Kiểm tra số chỉ thị được loại bỏ#
```

```
summary(mapthis)

############################
# NHẬN BIẾT CÁC CÁ THỂ VÀ CHỈ THỊ BỊ LẶP
############################

#Nhận biết các cá thể có kết quả kiểu gene xác đị nh bằng các chỉ thị giống nhau bằng cách sử dụng câu lệnh comparegeno(). Kết quả của câu lệnh là một ma trận sẽ được dùng để tạo bản đồ histogram bằng lệnh hist(). Lệnh rug() là một cách khác để hiển thị mật độ của số liệu trên bản đồ histogram. Phần đuôi bên phải của bản đồ được dùng để xác đị nh các chỉ thị có kết quả kiểm tra giống nhau cao.#
cg <- comparegeno(mapthis)
hist(cg[lower.tri(cg)], breaks=seq(0, 1, len=101), xlab="No. matching genotypes")
rug(cg[lower.tri(cg)])
# Một số cặp có kết quả kiểm tra giống nhau đến hơn 90%. Để xác đị nh rõ những cặp này, chúng ta sử dụng lệnh which()#
wh <- which(cg > 0.9, arr=TRUE)
wh <- wh[wh[,1] < wh[,2],]
wh

g <- pull.geno(mapthis)
table(g[144,], g[292,])

table(g[214,], g[216,])
table(g[238,], g[288,])

#Giữ lại một cá thể và loại bỏ các các cá thể giống với cá thể giống nhau khác. #
for(i in 1:nrow(wh)) {
 tozero <- !is.na(g[wh[i,1],]) & !is.na(g[wh[i,2],]) & g[wh[i,1],] != g[wh[i,2],]
 mapthis$geno[[1]]$data[wh[i,1],tozero] <- NA
  }
 mapthis <- subset(mapthis, ind=-wh[,2])


# Tiếp theo, tìm kiếm các chỉ thị giống nhau, giữ lại một chỉ thị và loại bỏ các chỉ thị còn lại. #
print(dup <- findDupMarkers(mapthis, exact.only=FALSE))

#mapthis <- drop.markers(mapthis, unlist(dup))

############################
# Tìm kiếm các chỉ thị không tuân theo tỷ lệ phân ly
# Chỉ thị phân tử xấu

# Trong phần này, chúng ta tìm kiếm các chỉ thị không tuân theo tỷ lệ phân ly. Lưu ý, đây là thế hệ F2 nên tỷ lệ phân ly phải là 1:2:1. Kiểm tra giá trị p-value của các chỉ thị có tỷ lệ khác biệt với 1:2:1.
```

```
gt <- geno.table(mapthis)
gt[gt$P.value < 0.05/totmar(mapthis),]

#Các chỉ thị có pvalue<10^-10 sẽ bị loại bỏ khỏi bảng số liệu. #
todrop <- rownames(gt[gt$P.value < 1e-10,])
mapthis <- drop.markers(mapthis, todrop)

##############################
#Để vẽ đồ thị về sự phân bố của các tần suất kiểu gene, có thể sử dụng các câu lệnh sau:
g <- pull.geno(mapthis)
gfreq <- apply(g, 1, function(a) table(factor(a, levels=1:3)))
gfreq <- t(t(gfreq) / colSums(gfreq))
par(mfrow=c(1,3), las=1)
for(i in 1:3)
plot(gfreq[i,], ylab="Genotype frequency", main=c("AA", "AB", "BB")[i],
ylim=c(0,1))

#Câu lệnh phức tạp, nên chép và dán câu lệnh vào R.

#################################################################
# Tính tần số tái tổ hợp (recombinant fraction, rf)                    #Tính ở đâu
# Sử dụng câu lệnh checkAlleles để nhận biết các tỷ lệ phân ly không theo 1:2:1  #
#################################################################

# Ước tính giá trị tần suất tái tổ hợp của bộ số liệu, bao gồm việc kiểm tra xem các số liệu có bị chuyển đổi (switched) hay không.Điều này sẽ thể hiện thông qua việc một số chỉ thị có liên kết chặt với nhau nhưng lại có tần suất tái tổ hợp bằng 0.5 hoặc lớn hơn 0.5 rất nhiều.
# Câu lệnh checkAlleles() cung cấp nhiều thông tin hơn về các allele có thể không tốt. Tuy nhiên, câu lệnh này chỉ sử dụng được với quần thể F2.
mapthis <- est.rf(mapthis)

checkAlleles(mapthis, threshold=5)

# Tạo đồ thị thể hiện mối tương quan giữa LOD score và tần suất tái tổ hợp ước tính cho tất cả các chỉ thị . Sử dụng câu lệnh pull.rf() để trích xuất dữ liệu về tần suất tái tổ hợp và LOD score và lưu lại dưới dạng ma trận.
rf <- pull.rf(mapthis)
lod <- pull.rf(mapthis, what="lod")
plot(as.numeric(rf), as.numeric(lod), xlab="Recombination fraction", ylab="LOD score")

#################################################################
# Bước đầu xây dựng các nhóm liên kết                    #
#################################################################
```

#Sử dụng câu lệnh formLinkageGroups() để xây dựng các nhóm liên kết sử dụng kết quả từ câu lệnh est.rf(). Trong câu lệnh formLinkageGroups() có hai thành phần là max.rf và min.lod; có thể diễn giải câu lệnh như sau: hai chỉ thị sẽ được cho vào cùng một nhóm liên kết nếu tần suất tái tổ hợp ước tính của hai chỉ thị này <=35, và có giá trị LOD score >=min.lod.

```
lg <- formLinkageGroups(mapthis, max.rf=0.35, min.lod=6)
table(lg[,2])
```

Vì sao rf lại lấy là 0.35

#Sắp xếp các chỉ thị vào các nhóm

```
mapthis <- formLinkageGroups(mapthis, max.rf=0.35, min.lod=6, reorgMarkers=TRUE)
plotRF(mapthis, alternate.chrid=TRUE)
```

#Lấy một chỉ thị từ nhóm 4, phân tích rf và LOD của chỉ thị này với tất cả chỉ thị còn lại.

```
rf <- pull.rf(mapthis)
lod <- pull.rf(mapthis, what="lod")
mn4 <- markernames(mapthis, chr=4)
par(mfrow=c(2,1))
plot(rf, mn4[3], bandcol="gray70", ylim=c(0,1), alternate.chrid=TRUE)
abline(h=0.5, lty=2)
plot(lod, mn4[3], bandcol="gray70", alternate.chrid=TRUE)
```

```
########################################
#Các vấn đề liên quan đến bảng số liệu#
########################################
```

#Chúng ta có thể xem xét vấn đề rõ hơn bằng cách kiểm tra một số bảng kiểu gene gồm 2 locus. Việc kiểm tra được thực hiện bằng lệnh geno.crosstab()

```
geno.crosstab(mapthis, mn4[3], mn4[1])

mn5 <- markernames(mapthis, chr=5)
geno.crosstab(mapthis, mn4[3], mn5[1])
```

Nếu kiểu gene của C3M13 đúng, các allele của C3M16 bị tráo đổi. Dùng câu lệnh switchAlleles() để chuyển đổi alleles

```
toswitch <- markernames(mapthis, chr=c(5, 7:11))
mapthis <- switchAlleles(mapthis, toswitch)
```

#Sau khi sử dụng câu lệnh switchAlleles(), chúng ta phải chạy lại các lệnh từ est.rf() trở về sau.

```
mapthis <- est.rf(mapthis)
plotRF(mapthis, alternate.chrid=TRUE)

rf <- pull.rf(mapthis)
lod <- pull.rf(mapthis, what="lod")
plot(as.numeric(rf), as.numeric(lod), xlab="Recombination fraction", ylab="LOD score")
```

```
#################################
# Xây dựng các nhóm liên kết    #
# Thứ tự các chỉ thị trên chromosome 5 #
#################################

# Xây dựng bản đồ liên kết
lg <- formLinkageGroups(mapthis, max.rf=0.35, min.lod=6)
table(lg[,2])

# Sắp xếp lại các chỉ thị
mapthis <- formLinkageGroups(mapthis, max.rf=0.35, min.lod=6, reorgMarkers=TRUE)

plotRF(mapthis)
# Xác đị nh thứ tự của các chỉ thị trên chromosome 5.
mapthis <- orderMarkers(mapthis, chr=5)
pull.map(mapthis, chr=5)

# Sử dụng lệnh ripple() để kiểm tra các thứ tự có thể có khác.
rip5 <- ripple(mapthis, chr=5, window=7)
summary(rip5)

#  Xem xét các khả năng xảy ra của các trình tự khác nhau
rip5lik <- ripple(mapthis, chr=5, window=4, method="likelihood",
 error.prob=0.005)
summary(rip5lik)

# Kết quả tương đối nhạy với xác suất sai số. Xác suất sai số càng nhỏ, chiều dài DNA sẽ càng lớn.

compareorder(mapthis, chr=5, c(1:7,9,8), error.prob=0.01)
compareorder(mapthis, chr=5, c(1:7,9,8), error.prob=0.001)
compareorder(mapthis, chr=5, c(1:7,9,8), error.prob=0)


#Đổi trình tự chỉ thị số 8 và số 9
mapthis <- switch.order(mapthis, chr=5, c(1:7,9,8), error.prob=0.005)
pull.map(mapthis, chr=5)

#################################
# Trình tự chỉ thị trên chromosome số 4 #
#################################

mapthis <- orderMarkers(mapthis, chr=4)
pull.map(mapthis, chr=4)
rip4 <- ripple(mapthis, chr=4, window=7)
```

```
summary(rip4)
rip4lik <- ripple(mapthis, chr=4, window=4, method="likelihood",
error.prob=0.005)
summary(rip4lik)

mapthis <- switch.order(mapthis, chr=4, c(1:8,10,9), error.prob=0.005)
pull.map(mapthis, chr=4)

##################################
# Trình tự các chỉ  thị  trên chromosome số 3 #
##################################

mapthis <- orderMarkers(mapthis, chr=3)
pull.map(mapthis, chr=3)
rip3 <- ripple(mapthis, chr=3, window=7)
summary(rip3)
rip3lik <- ripple(mapthis, chr=3, window=4, method="likelihood",
error.prob=0.005)
summary(rip3lik)
##################################
# Trình tự chỉ  thị  trên chromosome số 2 #
##################################

summary(rip2)
pull.map(mapthis, chr=2)
rip2 <- ripple(mapthis, chr=2, window=7)
summary(rip2)
rip2lik <- ripple(mapthis, chr=2, window=4, method="likelihood",
error.prob=0.005)
summary(rip2lik)

pat2 <- apply(rip2[,1:24], 1, paste, collapse=":")
pat2lik <- apply(rip2lik[,1:24], 1, paste, collapse=":")
rip2 <- rip2[match(pat2lik, pat2),]
plot(rip2[,"obligXO"], rip2lik[,"LOD"], xlab="obligate crossover count",
ylab="LOD score")

##################################
# Trình tự chỉ  thị  trên chromosome 1 #
##################################

mapthis <- orderMarkers(mapthis, chr=1)
pull.map(mapthis, chr=1)
rip1 <- ripple(mapthis, chr=1, window=7)
summary(rip1)
```

```
rip1lik <- ripple(mapthis, chr=1, window=4, method="likelihood",
error.prob=0.005)
summary(rip1lik)

######################
# Bước cuối cùng trong lập bản đồ liên kết #
######################
summaryMap(mapthis)
plotMap(mapthis, show.marker.names=TRUE)

plotRF(mapthis)

#Loại bỏ chỉ thị có nghi vấn và đánh giá sựthay đổi của chiều dài chromosome.
dropone <- droponemarker(mapthis, error.prob=0.005)
par(mfrow=c(2,1))
plot(dropone, lod=1, ylim=c(-100,0))
plot(dropone, lod=2, ylab="Change in chromosome length")

# Nhận biết các chỉ thị có thể dẫn đến sựthay đổi chiều dài chromosome
summary(dropone, lod.column=2)

# Loại bỏ các chỉ thị xấu
badmar <- rownames(summary(dropone, lod.column=2))[1:3]
mapthis <- drop.markers(mapthis, badmar)

newmap <- est.map(mapthis, error.prob=0.005)
mapthis <- replace.map(mapthis, newmap)
summaryMap(mapthis)

# Việc loại bỏ các chỉ thị xấu giúp rút ngắn chiều dài của bản đồ từ650 cM còn 524cM.


# Nhận biết các cá thể có vấn đề
plot(countXO(mapthis), ylab="Number of crossovers")

# Phát hiện hai cá thể có 73 và 86 trao đổi chéo. Loại bỏ các cá thể có nhiều hơn 50 trao đổi chéo.
mapthis <- subset(mapthis, ind=(countXO(mapthis) < 50))

#Trên lý thuyết, cần lặp lại quá trình xác đị nh thứtựcủa các chỉ thị . Trong trường hợp này, chúng ta
chỉ kiểm tra thứtựcác chỉ thị ởchromosome 5.
summary(rip <- ripple(mapthis, chr=5, window=7))

summary(rip <- ripple(mapthis, chr=5, window=2, method="likelihood",
error.prob=0.005))
```

```
# Chúng ta thấy rằng thứ tự của chỉ thị thứ 8 và 9 cần được tráo đổi.
mapthis <- switch.order(mapthis, chr=5, c(1:7,9,8), error.prob=0.005)
pull.map(mapthis, chr=5)

#Xây dựng bản đồ di truyền lần nữa
newmap <- est.map(mapthis, error.prob=0.005)
mapthis <- replace.map(mapthis, newmap)
summaryMap(mapthis)
##########################
#Ước tính xác suất sai số (error rate) #
##########################

loglik <- err <- c(0.001, 0.0025, 0.005, 0.0075, 0.01, 0.0125, 0.015, 0.0175, 0.02)
for(i in seq(along=err)) {
cat(i, "of", length(err), "\n")
tempmap <- est.map(mapthis, error.prob=err[i])
loglik[i] <- sum(sapply(tempmap, attr, "loglik"))
}
lod <- (loglik - max(loglik))/log(10)

plot(err, lod, xlab="Genotyping error rate", xlim=c(0,0.02),
ylab=expression(paste(log[10], " likelihood")))

#Nhận biết các chỉ thị có trao đổi chéo đôi - có tỷ lệ sai số khác biệt so với các chỉ thị còn lại của bộ số liệu
mapthis <- calc.errorlod(mapthis, error.prob=0.005)
print(toperr <- top.errorlod(mapthis, cutoff=6))

plotGeno(mapthis, chr=1, ind=toperr$id[toperr$chr==1],
cutoff=6, include.xo=FALSE)

#Các số liệu này không được loại bỏ khỏi bộ số liệu. Tuy nhiên, nếu muốn loại bỏ thì sử dụng câu lệnh ở bên dưới (nhớ loại bỏ ký tự "#")
#mapthis.clean <- mapthis
#for(i in 1:nrow(toperr)) {
#   chr <- toperr$chr[i]
#   id <- toperr$id[i]
#   mar <- toperr$marker[i]
#   mapthis.clean$geno[[chr]]$data[mapthis$pheno$id==id, mar] <- NA
# }

##########################################
# Kiểm tra tỷ lệ phân ly không hợp lý/ khác biệt #
##########################################
```

```
gt <- geno.table(mapthis, scanone.output=TRUE)
par(mfrow=c(2,1))
plot(gt, ylab=expression(paste(-log[10], " P-value")))
plot(gt, lod=3:5, ylab="Genotype frequency")
abline(h=c(0.25, 0.5), lty=2, col="gray")
################
#Xong phần xây dựng bản đồ liên kết#
################
```

Bước 1: Xử lý dữ liệu trước khi xây dựng các nhóm liên kết.

- Tóm tắt bộ dữ liệu phân tích
- Loại bỏ cá thể và chỉ thị bị thiếu dữ liệu
- Loại bỏ cá thể và chỉ thị trùng lặp (loại bỏ 1 cá thể/chỉ thị và giữ lại cá thể/chỉ thị còn lại)
- Kiểm tra khả năng tỷ lệ phân kiểu gene ở F2 và mức độ liên kết giữa các cặp chỉ thị

Bước 2: Xây dựng các nhóm liên kết và thứ tự các marker.

- Xây dựng các nhóm liên kết
- Kiểm tra từng cặp chỉ thị liên kết và nhận biết các allene bị chuyển đổi
- Sắp xếp thứ tự các chỉ thị trong từng nhóm liên kết
- Xác định thứ tự các chỉ thị ý nghĩa nhất

Bước 3: Kiểm tra các nghi vấn bản đồ liên kết ban đầu.

- Kiểm tra khoảng cách lớn trên các chromosome
- Xem xét số lượng trao đổi chéo quan sát ở mỗi cá thể
- Kiểm tra marker có xảy ra trao đổi chéo đôi
- Kiểm tra ý nghĩa của tỷ lệ phân ly kiểu gene

Bước 4: Xây dựng bản đồ liên kết cuối cùng.