```
# Correcting pervasive errors in genotypic datasets to develop genetic maps by Sadal Hwang and Tong Geon Lee. For use in custom follow-up studies, additional functions are arranged after the pound sign.

#
#
# Pre-process datasets.

date()

install.packages("qtl")
install.packages("MASS")
install.packages("calibrate")
install.packages("moments")
install.packages("psych")

library(qtl)
qtlversion()
library("MASS")
library("calibrate")
library(moments)
library(psych)

date()
rm(list = ls(all = TRUE))
version
setwd("C:/HxC Results")

memory.limit()
memory.limit(2048)
memory.limit(3583)
memory.limit(4027)

# Load data.

OBJNAMEBC <- read.cross (format=c("csvs"), dir="C:/HxC Results", genfile="Additional file 2.csv",
phefile="Additional file 3.csv", na.strings=c("-","N","NA"), genotypes=c("A","B"),
alleles=c("A","B"), error.prob=0.0001, map.function=c("kosambi"))
summary(OBJNAMEBC)
jittermap(OBJNAMEBC, amount=1e-6)
OBJNAME <- convert2riself(OBJNAMEBC)
jittermap(OBJNAME, amount=1e-6)
summary(OBJNAME)
jittermap(OBJNAME, amount=1e-6)
hcsnpssr <- OBJNAME

# Set graphics margin and font parameters.

par(mar=c(5.1,5.1,5.1,3.1),mfrow=c(1,1),las=0,cex=1,cex.axis=1,cex.lab=1,cex.main=1,cex.sub=1,font=12,font.main=2,font.sub=2,font.axis=2,font.lab=2)
oldpar <-
par(mar=c(5.1,5.1,5.1,3.1),mfrow=c(1,1),las=0,cex=1,cex.axis=1,cex.lab=1,cex.main=1,cex.sub=1,font=12,font.main=2,font.sub=2,font.axis=2,font.lab=2)

#
#
# Exclude duplicated samples (Fig. 2).

# Find samples with 97% or more of genotypes shared. Note that the R/qtl index number is not the same as the input RIL or F2 RIL number. The wh97 command displays "row col" and index numbers.

ALLpairs <- comparegeno(hcsnpssr, what="proportion")
par(mfrow=c(1,1),las=0)
hist(ALLpairs, breaks=seq(0, 1, len=101), xlab="Fractional Degree of SNP/SSR Genotypic Identity Between a Pair of RILs")
rug(ALLpairs)
wh97 <- which(ALLpairs > 0.97, arr=TRUE)
wh97 <- wh97[wh97[,1] < wh97[,2],]
wh97

# Check the number of markers genotyped in each sample.
```

```
nt <- ntyped(hcsnpssr)

# Check samples with 97% or more of genotypes shared and the number of markers genotyped. Can
keep samples that carry more markers genotyped.
nt[wh97]

# Match the R/qtl index number corresponds to the input sample number.
match(names(nt[wh97]), getid(hcsnpssr))

# Alternatively, run functions below to remove samples with 97% or more of genotypes shared.
# todrop <- rep(NA, nrow(wh97))
# for(i in seq(along=todrop)) {
# this.nt <- nt[ wh97[i,] ]
# if(this.nt[1] <= this.nt[2]) todrop[i] <- wh[i,1]
# else todrop[i] <- wh97[i,2]
# }

# Confirm samples removed.

# todrop
# nt[todrop]

# Check duplicate markers.
print(duptrue  <- findDupMarkers(hcsnpssr, exact.only=TRUE, adjacent.only=FALSE))

# To remove duplicate markers, run functions below.
# totmar(hcsnpssr)
# dupmar.exact <- findDupMarkers(hcsnpssr, exact.only=TRUE, adjacent.only=FALSE)
# hcsnpssr <- drop.markers(hcsnpssr, unlist(dupmar.exact))
# totmar(hcsnpssr)

#
#
# Identify problematic individuals based on crossovers (Fig. 3a).

plot(countXO(hcsnpssr, bychr=FALSE), ylab="Number of Crossovers", ylim=c(0,200))

# Check the parametric percentile of the value distribution.

countXO(hcsnpssr)
crossover <- countXO(hcsnpssr)
quantile(crossover, 0.05)
quantile(crossover, 0.05)

# Choose threshold values.

XO100up <- subset(hcsnpssr, ind=(countXO(hcsnpssr) > 99))
getid(XO100up)

XO100up <- subset(hcsnpssr, ind=(countXO(hcsnpssr) < 42))
getid(XO100up)

# Plot numbers of observed crossovers.

plot(countXO(hcsnpssr, bychr=FALSE), ylab="Number of Crossover", xlab="individual",
ylim=c(0,200))
abline(h=100, lty=2)
abline(h=42, lty=2)

# Output samples retained
# nind(hcsnpssr)
# hcsnpssr <- subset(hcsnpssr, ind=(countXO(hcsnpssr) > 41))
# nind(hcsnpssr)
# hcsnpssr <- subset(hcsnpssr, ind=(countXO(hcsnpssr) < 99))
# nind(hcsnpssr)
# getid(hcsnpssr)

# Identify problematic markers based on crossovers (Fig. 3b).

hcsnpssr <- calc.errorlod(hcsnpssr, error.prob=0.0001, map.function=c("kosambi"))
```

```
# Identify marker genotypes with a large LOD score.

print(toperr5 <- top.errorlod(hcsnpssr, cutoff=5, msg=TRUE))

# Plot genotypes that show potential false crossovers on chromosome 19.

plot.geno(hcsnpssr, chr=19, ind=toperr5$id[toperr5$chr==19], cutoff=5, include.xo=TRUE)

# Markers with potential crossovers identified in three or more samples can be dropped.
# hcsnpssr <- drop.markers(hcsnpssr,
c("S12624","S13819","S14393","S14569","S14852","S15648","S16344","S16356","S21193","S24479","S261
18","S28451","S28613","S29088","Sat_134","Sat_210","Sat_342","Satt272"))
# Redo calc.errorlod as below.
# hcsnpssr <- calc.errorlod(hcsnpssr, error.prob=0.0001, map.function=c("kosambi"))
# print(toperr5 <- top.errorlod(hcsnpssr, cutoff=5, msg=TRUE))

# Replace genotypic errors with NA. R/qtl treats NA codes as missing genotypes. This procedure
can be useful to identify QTL in the subsequence procedure.
# hcsnpssr.clean <- hcsnpssr
# for(i in 1:nrow(toperr)) {
# chr <- toperr$chr[i]
# id <- toperr$id[i]
# mar <- toperr$marker[i]
# hcsnpssr.clean$geno[[chr]]$data[hcsnpssr$pheno$ID==id, mar] <- NA
# }

# Calculate and print the top.errorlod output for the "hcsnpssr.clean".
# hcsnpssr.clean <- calc.errorlod(hcsnpssr.clean, error.prob=0.0001, map.function=c("kosambi"))
# print(toperr5 <- top.errorlod(hcsnpssr.clean, cutoff=5, msg=TRUE))

#
#
# Color polymorphic markers (Fig. 4a).

geno.image(hcssr, reorder=FALSE, main="Marker Genotypes: A=red  B=blue  Missing=white",
alternate.chrid=FALSE)

# Plot mono-allelic genotypes on chromosomes 1, 8, 12, and 20 (Fig. 4b).

plot.missing(hcsnpssr, chr=c("1", "8", "12", "20"), reorder=FALSE,
main="Chromosome",alternate.chrid=FALSE)
abline(h=c(37), lty=1, col="black")
abline(h=c(77), lty=1, col="black")
abline(h=c(113), lty=1, col="black")
abline(h=c(149), lty=1, col="black")
abline(h=c(188), lty=1, col="black")
abline(h=c(228), lty=1, col="black")
abline(h=c(266), lty=1, col="black")

#
#
# Plot marker positions based on the consensus map (Fig. 5a).

x <-  c(1,      2,     3,     4,    5,     6,     7,     8,     9,    10,    11,    12,    13,
14,    15,   16,    17,    18,    19,    20)
y <-  c(0.00,   0.00,  0.00,  0.00, 0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
0.00,  0.00, 0.00,  0.00,  0.00,  0.00,  0.00)
z <-  c(98.41,140.63, 99.51,112.32,86.75,136.51,135.15,146.67,
99.60,132.89,124.24,120.50,120.03,108.18, 99.88,92.27,119.19,107.09,101.14,112.77)
chrlenglab <- c(" 98",141,100,112," 87",  137,   135,   147,   100,   133,   124,   121,   120,
108,   100, " 92",  119,   107,   101,   113)
x1 <- c(0.5,     1.5,   2.5,   3.5,   4.5,   5.5,   6.5,   7.5,   8.5,   9.5,  10.5,  11.5,
12.5,  13.5,  14.5,  15.5,  16.5,  17.5,  18.5,  19.5)
z1 <- c(104.41,146.63,105.51,118.32,
92.75,142.51,141.15,152.67,105.60,138.89,130.24,126.50,126.03,114.18,105.88,
98.27,125.19,113.09,107.14,118.77)

plot.map(hcsnpssr, horizontal=FALSE, shift=FALSE, show.marker.names=FALSE, alternate.chrid=FALSE,
ylab="Kosambi Map Distance (cM)", ylim=c(160,0))

points(x,y, cex=0.75, pch=1)
points(x,z, cex=0.75, pch=1)

textxy(x1, z1, chrlenglab, cx = 0.7, dcol = "black", m = c(0, 0))

summary(hcsnpssr)

jittermap(hcsnpssr, amount=1e-6)

# Plot the logarithm of odds scores against the estimated recombination frequency for marker
pairs to indiciate erroneous markers (Fig. 5b).

rf <- pull.rf(hcsnpssr)
lod <- pull.rf(hcsnpssr, what="lod")
plot(as.numeric(rf), as.numeric(lod), xlab="Recombination fraction", ylab="LOD score")
abline(v=0.5,h=c(3), lty=2)
checkAlleles(hcsnpssr, threshold=3, verbose=TRUE)

#
#
# Exclude markers with unusual segregation patterns (Fig. 6).

gtAll <- geno.table(hcsnpssr, scanone.output=TRUE)
par(mfrow=c(1,1), las=0)
plot(gtAll, ylab=expression(paste(-log[10], " P-value")))
abline(h=c(4.18), lty=2, col="black")
plot(gtAll, lod=3:4, ylab="Genotype frequency", ylim=c(0.2,0.8))
abline(h=c(0.5), lty=1, col="black")

# Find the threshold value.

gt <- geno.table(hcsnpssr, scanone.output=FALSE)
gt

sortgt <- gt[order(gt$P.value) , ]
sortgt

gt[ gt$P.value < 0.01, ]

0.05/(totmar(hcsnpssr))

gt[ gt$P.value < 0.05/(totmar(hcsnpssr)), ]

sortedsuspect.markers <- rownames(sortgt[ sortgt$P.value < 0.05/(totmar(hcsnpssr)), ])
sortedsuspect.markers

# Remove fluctuating genotypes.

totmar(hcsnpssr)
hcsnpssr <- drop.markers(hcsnpssr, sortedsuspect.markers)
hcsnpssr <- drop.markers(hcsnpssr, c("Sat_356","S13675","Satt126"))
totmar(hcsnpssr)

#
#
# Create a draft genetic map (Fig. 7a)

hmohcsnpssrm <- est.map(hcsnpssr, error.prob=0.0001, map.function=c("kosambi"), maxit=10000,
tol=1e-6, verbose=FALSE)
plot.map(hmohcsnpssrm, horizontal=FALSE, shift=FALSE, show.marker.names=FALSE,
alternate.chrid=FALSE, ylab="Kosambi Map Distance (cM)")

# Correct undesirable gaps

fix(hmohcsnpssrm)

`13` = structure(c(5.145, 16.6542353032272, 20.3318918151553,
21.4710873809762, 25.2627582975522, 26.4999834433155, 28.6556993519845,
35.5476471584497, 37.1905135091004, 37.1905135591004, 40.1471763535349,
51.1300316196731, 54.4643244340662, 60.3049068620222, 64.9985381747556,
65.3771130122005, 65.3771130622005, 65.3771131122005, 65.3771131622005,
65.7403884060601, 65.7403884560601, 65.7403885060601, 65.7403885560601,
68.8009249229582, 68.8009249729582, 69.8600277601051, 70.508394120728,
71.4206148267275, 572.173579811769, 588.139271233326, 598.84188899469,
599.375221949952, 610.429664229542, 610.429664279542, 610.837298247536,
646.248748314974, 652.256976949426, 652.256976999426, 652.256977049426,
657.903827713032, 658.086959040769, 658.72940880796, 658.963808097771,
665.782486971493, 669.45640350342, 674.224273227412, 677.393889844228,
701.844756189299, 707.367663913158, 707.905226224239, 724.120703046941),

# Manually correct undesirable gaps (bold above) while you keep the distance between two flanking
markers. The new position of the first marker in bold is 145.03 [71.42 plus 73.61 (the Kosambi
map distance)].

71.4206148267275, 145.031589305889, 160.997280727446, 171.69989848881,
172.233231444072, 183.287673723662, 183.287673773662, 183.695307741656,
219.106757809094, 225.114986443546, 225.114986493546, 225.114986543546,
230.761837207152, 230.944968534889, 231.58741830208, 231.821817591891,
238.640496465613, 242.31441299754, 247.082282721532, 250.251899338348,
274.702765683419, 280.225673407278, 280.763235718358, 296.978712541061),

# Compare the final map to the consensus map (Fig. 7b)

plot.map(hcsnpssr,hmohcsnpssrm, horizontal=FALSE, shift=FALSE, show.marker.names=FALSE,
alternate.chrid=FALSE, ylab="Kosambi Map Distance (cM)")

# end
```