Prepare Genotype Data

Izel Fourie Sørensen, Pernille Merete Sarup, Palle Duun Rohde

April 3, 2018

In this script we prepare a data frame of genotype data to be used in downstream genomic analyses. Genotypes from the *Drosophila melanogaster* Genetic Reference Panel (DGRP) is used as an example. The genotype data is available at http://dgrp2.gnets.ncsu.edu/data.html under the heading "Genotype files" at the top of the page. The "tabular formatted genotype" data is used.

Download and read the genotype data

```
download.file("http://dgrp2.gnets.ncsu.edu/data/website/dgrp2.tgeno",
              destfile = "C:/Users/Izel/Dropbox/qgg-usersguide/data/dgrp2.tgeno")
```

```
genotypes <- read.table( "C:/Users/Izel/Dropbox/qgg-usersguide/data/dgrp2.tgeno",
                         header = TRUE, sep="")

dim(genotypes)
```

```
## [1] 4438427     214
```

```
genotypes[1:5,1:12]
```

```
##   chr  pos          id ref alt refc altc qual cov line_21 line_26 line_28
## 1  2L 4998 2L_4998_SNP   G   A  117    5  999  12       0       -       0
## 2  2L 5002 2L_5002_SNP   G   T  127    1  999  13       0       -       0
## 3  2L 5039 2L_5039_SNP   C   T    1  118  999  21       2       -       2
## 4  2L 5040 2L_5040_SNP   G   A    1  118  999  21       2       -       2
## 5  2L 5092 2L_5092_SNP   C   T    6  119  999  22       2       -       2
```

The data is space delimited and alleles are encoded as '0' for reference allele and '2' for alternative allele (does not imply minor allele). Genotypes that were heterozygous or could not be called reliably (low quality of base calls, low quality of mapping, or shallow read depth) are annotated by "-".

Edit data frame

Edit column names: remove the "line_" prescript for each of the lines. Use `id` columns (SNP ids) as rownames and change `class` to `character`.

```
colnames(genotypes) <- gsub("line_", "",colnames(genotypes))
rownames(genotypes) <- as.character(genotypes$id)
```

Subset genotype data - include SNP variants only

Create a vector containing the variant type of genetic marker, i.e., single nucleotide polymophism ('SNP'), insertion ('INS'), deletion ('DEL') and microsatellite ('MNP'). The third component of the `id` column (separated by "_") indicates the variant type.

```
vtype <- sapply(as.character(genotypes$id), function(x){
strsplit(x,split="_")[[1]][3]
    } )
unique(vtype)
```

```
## [1] "SNP" "INS" "DEL" "MNP"
```

Prepare a subset of the genotype data containing only the SNP variants. Subsets of insertion, deletion and microsatellite variants can be prepared in a similar way.

```
gsnp <- genotypes[vtype=="SNP",]
dim(gsnp)
```

```
## [1] 3963420     214
```

Create data frames for SNP information and for genotypes

The genotype data frame is split into 2 data frames:

1. A data frame (`snpI`) that contains information about each variant (e.g. variant ID, chromosome, position on the chromosome, reference (ref)/alternative (alt) alleles, ref/alt alleles count, phred quality scores, etc.).
2. A data frame (`snpG`) that contains the genotypes for each variant of the 205 DGRP lines.

```
  snpI <- gsnp[,1:9]
  snpG <- gsnp[,-c(1:9)]
```

Remove the "SNP" suffix from the row names in the `snpI` and `snpG` data frames.

```
rownames(snpI) <- gsub("_SNP","",rownames(snpI))
rownames(snpG) <- gsub("_SNP","",rownames(snpG))
head(rownames((snpG)))
```

```
## [1] "2L_4998" "2L_5002" "2L_5039" "2L_5040" "2L_5092" "2L_5095"
```

Look at the data.

```
dim(snpI)
```

```
## [1] 3963420       9
```

```
head(snpI)
```

```
##         chr  pos          id ref alt refc altc qual cov
## 2L_4998  2L 4998 2L_4998_SNP   G   A  117    5  999  12
## 2L_5002  2L 5002 2L_5002_SNP   G   T  127    1  999  13
## 2L_5039  2L 5039 2L_5039_SNP   C   T    1  118  999  21
## 2L_5040  2L 5040 2L_5040_SNP   G   A    1  118  999  21
## 2L_5092  2L 5092 2L_5092_SNP   C   T    6  119  999  22
## 2L_5095  2L 5095 2L_5095_SNP   T   A    4  115  999  22
```

```
dim(snpG)
```

```
## [1] 3963420     205
```

```
str(snpG[1:5])
```

```
## 'data.frame':    3963420 obs. of  5 variables:
##  $ 21: Factor w/ 3 levels "-","0","2": 2 2 3 3 3 3 2 2 2 2 ...
##  $ 26: Factor w/ 3 levels "-","0","2": 1 1 1 1 1 1 2 2 2 2 ...
##  $ 28: Factor w/ 3 levels "-","0","2": 2 2 3 3 3 3 2 2 2 2 ...
##  $ 31: Factor w/ 3 levels "-","0","2": 2 2 3 3 3 3 2 2 2 3 ...
##  $ 32: Factor w/ 3 levels "-","0","2": 1 1 1 1 1 1 2 2 2 2 ...
```

```
snpG[1:5,1:15]
```

```
##         21 26 28 31 32 38 40 41 42 45 48 49 57 59 69
## 2L_4998  0  -  0  0  -  0  0  0  -  0  0  0  -  -  0
## 2L_5002  0  -  0  0  -  0  0  0  -  0  0  0  -  -  0
## 2L_5039  2  -  2  2  -  2  2  2  -  2  -  2  -  -  2
## 2L_5040  2  -  2  2  -  2  2  2  -  2  -  2  -  -  2
## 2L_5092  2  -  2  2  -  2  2  2  -  2  2  2  -  -  2
```

It takes some time to run the script thus far and it also fills the R environment. It may be convenient to save the data frames `snpI` and `snpG` at this stage. This step is optional.

```
save(snpG,file="./genotypes/snpG.Rdata")
save(snpI,file="./genotypes/snpI.Rdata")
```

Here we clean the R environment since R may have trouble executing the script that follows with a filled environment.

```
rm(list = ls(all=TRUE))
gc()
```

```
##           used (Mb) gc trigger   (Mb)   max used    (Mb)
## Ncells  499062 26.7   11554252  617.1   14442815   771.4
## Vcells 9232846 70.5 1285449637 9807.3 2084476824 15903.3
```

Filter the SNP information (snpI) and genotype (snpG) data frames

Edit the SNP genotype data (`snpG` and `snpI`): filter data according to Phred scaled variant quality, genotype call rate and minor allele frequency criteria.

Load the `snpG` and `snpI` data frames. If the R environment hasn't been cleared as suggested above, ignore this step.

```
load(file = "./genotypes/snpG.Rdata")
load(file = "./genotypes/snpI.Rdata")
```

Phred scaled quality

List SNPs with a Phred scaled quality (`qual` variable) of more than or equal to 500. If the environment was cleaned as suggested above, reload the `snpG` and `snpI` data frames.

```
phred500 <- snpI$qual>=500
```

Genotype call rate

Select SNPs with a genotype call rate of 80%. The sum of the number of alternative alleles scored `snpI$altc` and number of reference alleles scored `snpI$refc` (saved as `nC`) gives the number of genotypes that could be scored for a SNP across all lines. 164 genotypes scored out of a possible 205 = 80% call rate.

`nC80` is a vector of logicals, where SNPs with a sum of `snpI$altc` and `snpI$refc` more than or equal to 164 is TRUE.

```
nC <- snpI$altc + snpI$refc
nC80 <- nC>=164
```

Minor allele frequency

SNPs with a minor allele frequency (MAF) < 0.05 were removed. `mafA` is the frequency of the alternative allele and `mafR` is the frequency of the reference allele. The DGRP was generated by 20 generations of full sib mating resulting in an approximated inbreeding coefficient of 0.986, resulting in most segregating sites being homozygous (encoded as 0 or 2). Genotypes that were heterozygous or could not be called reliably (low quality of base calls, low quality of mapping, or shallow read depth) are annotated by "-".

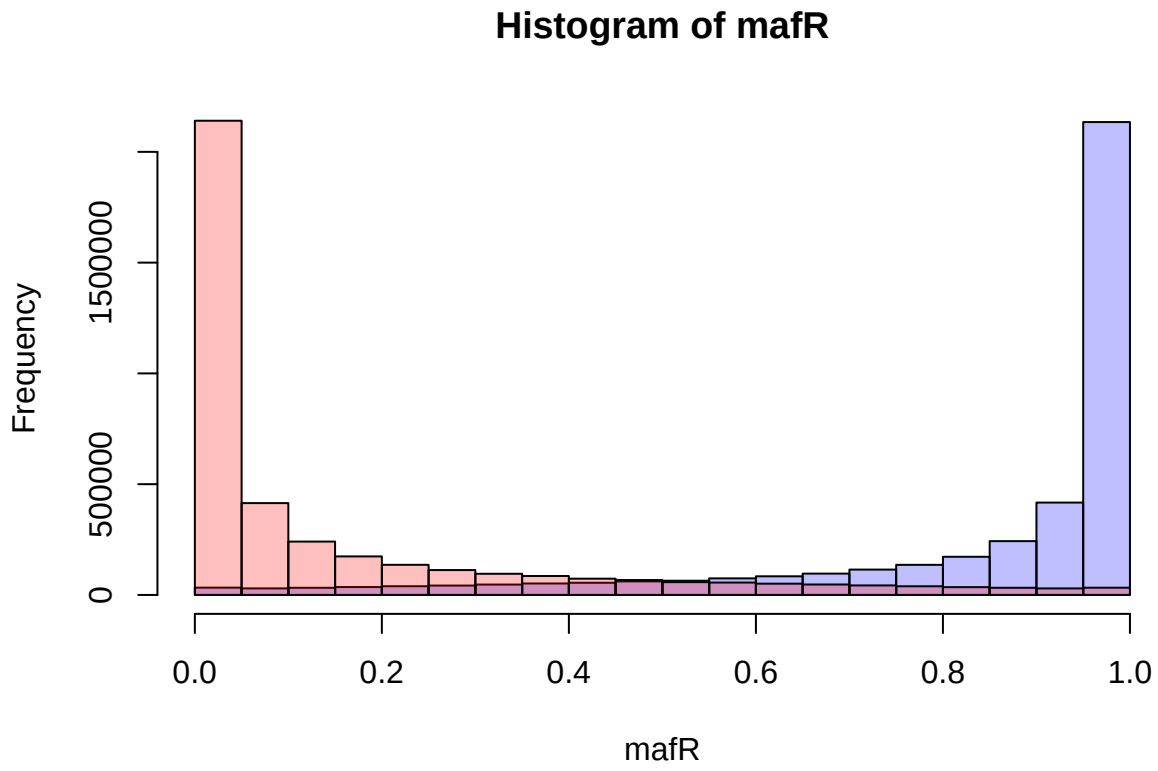
```
mafR <- snpI$refc/nC
mafA <- snpI$altc/nC
    head(mafR)
```

```
## [1] 0.959016393 0.992187500 0.008403361 0.008403361 0.048000000 0.033613445
```

```
    head(mafA)
```

```
## [1] 0.04098361 0.00781250 0.99159664 0.99159664 0.95200000 0.96638655
```

```
hist( mafR, col=rgb(0,0,1,1/4), xlim=c(0,1))
hist( mafA, col=rgb(1,0,0,1/4), xlim=c(0,1), add=T)
```



Create logicals that are TRUE if maf is >= 0.05 and FALSE otherwise: `mafA05` for the alternative allele and `mafR05` for the reference allele.

```
mafA05 <- mafA>=0.05
mafR05 <- mafR>=0.05
```

Alleles with a frequency of more than 0.05 could also be selected with `maf05 <- 0.05<=mafA & mafA<=0.95`.

Subset relevant SNPs

Create the `keep` vector of logicals that are TRUE for SNPs that meet all of the criteria regarding phred scaled quality, call rate and MAF. Subset the snpI and snpG data frames, containing the selected subset of SNPs, based on this vector.

```
keep <- phred500 & nC80 & mafA05 & mafR05
    sum(keep)
```

```
## [1] 1725755
```

```
head(rownames(snpI)[keep])
```

```
## [1] "2L_5317" "2L_5372" "2L_5390" "2L_5403" "2L_5465" "2L_5598"
```

```
snpG <- snpG[keep,]
snpI <- snpI[keep,]
```

Convert the data frame `snpG` to a matrix. Contents of the matrix is `character`.

```
snpG <- as.matrix(snpG)
head(snpG[,1:20])
```

```
##         21  26  28  31  32  38  40  41  42  45  48  49  57  59  69  73
## 2L_5317 "0" "0" "0" "2" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
## 2L_5372 "0" "0" "0" "0" "0" "2" "2" "0" "2" "-" "2" "0" "0" "-" "-" "2"
## 2L_5390 "2" "0" "2" "0" "0" "2" "2" "2" "2" "-" "2" "-" "-" "-" "0" "2"
## 2L_5403 "0" "0" "0" "0" "0" "0" "0" "0" "0" "2" "2" "0" "0" "0" "0" "2"
## 2L_5465 "0" "0" "0" "0" "0" "0" "0" "0" "0" "2" "2" "0" "0" "0" "0" "2"
## 2L_5598 "0" "0" "0" "0" "0" "0" "0" "0" "0" "2" "0" "0" "0" "0" "0" "2"
##         75  83  85  88
## 2L_5317 "0" "0" "-" "0"
## 2L_5372 "2" "2" "-" "2"
## 2L_5390 "2" "2" "-" "2"
## 2L_5403 "2" "2" "0" "0"
## 2L_5465 "2" "2" "0" "0"
## 2L_5598 "-" "2" "0" "0"
```

Convert matrix content of `snpG` to `numeric`. The second argument (the number 2) in `apply` specifies to work by columns instead of by rows.

```
snpG <- apply(snpG,2,as.numeric)
rownames(snpG) <- rownames(snpI)
head(snpG[,1:20])
```

```
##         21 26 28 31 32 38 40 41 42 45 48 49 57 59 69 73 75 83 85 88
## 2L_5317  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0 NA  0
## 2L_5372  0  0  0  0  0  2  2  0  2 NA  2  0  0 NA NA  2  2  2 NA  2
## 2L_5390  2  0  2  0  0  2  2  2  2 NA  2 NA NA NA  0  2  2  2 NA  2
## 2L_5403  0  0  0  0  0  0  0  0  0  2  2  0  0  0  0  2  2  2  0  0
## 2L_5465  0  0  0  0  0  0  0  0  0  2  2  0  0  0  0  2  2  2  0  0
## 2L_5598  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  2 NA  2  0  0
```

Save edited SNP data as `.Rdata` and `.rds` files. The `.rds` files are loaded in the compute_W script. For more information see `?saveRDS`.

```
save(snpG,file="./genotypes/snpGE.Rdata")
save(snpI,file="./genotypes/snpIE.Rdata")

saveRDS(snpG, file = "./genotypes/snpGE.rds")
saveRDS(snpI,file="./genotypes/snpIE.rds")
```