Package tutorial TD

Yanjun Zan, Thibaut Payen, Örjan Carlborg

3 Sep 2018

1. Evaluating the input

```
input_folder <-"/home/thibaut/Gallus/Projects/genotypeTIGER/data/with.fam.f2.call2-v2/" # this is a lin
all_vcf <- setdiff(list.files(path = input_folder),list.dirs(path = input_folder,full.names = F)) #  ex
all_id <- gsub(pattern = "(\\d+)\\.vcf$",replacement = "\\1",x = all_vcf) # extract the ID
out.put <- data.frame(array(NA,dim=c(length(all_vcf),1039))) # generate a holder
rownames(out.put) <- all_id
total <- numeric(length(all_id))
require(data.table)

## extract the number of marker at each 1 Mb bin across individuls
for( i in 1:nrow(out.put)){
    input_now <- paste0(input_folder,all_vcf[i])
    total[i] <- nrow(fread(input_now))
    #num_now<- get_density_input()
    #out.put[i,] <- wrap_get_density(inputfile = input_now,binsize =1,cut = T,cutoff = 0, chr.match = #
    cat(i, "\n")
}
# get density
density <- apply(out.put,1,mean)
## get id with more than 5 Markers/Mb
id.keep <- names(density)[density >5]
```

2. QC

Sample mix-ups, DNA contaminations and pedigree errors in the data will lead to inaccurate genotype imputation. Individuals affected by these errors are likely to have higher number of genome-wide crossover events and therefore a lower call rate after filtering. Here, what we shown as "double-/single- cross over" is just an approximation. Users are free to use number of crossover estimated from Rqtl as an more accurated filtering criteria, which gave very similar result in our case.

2.1 Double cross over

Here we first have a look by filtering out genotype swithed twich in less than 3Mb

```
all <- list.files(all_vcf,pattern = "\\d+\\.vcf\\.(\\d+)\\.rough_COs\\.refined\\.breaks.txt")
chr <- sort(as.numeric(unique(gsub(pattern = "\\d+\\.vcf\\.(\\d+)\\.rough_COs\\.refined\\.breaks.txt",r
index.keep <- which(id_all %in% id.keep)

Double_xo_w_f <- Extract_Double_co_all_chr(id_all[index.keep],all_vcf[index.keep],chromosome = chr,gap=
```

2.2 Check the number of cross over

Alternatively, we could check the number of cross over with/without filtering(by setting gap=NULL and filter=F)

```
cross_w_f_3 <- Extract_co_all_chr(chromosome = chr,id_all = id_all[index.keep],all_vcf = all_vcf[index.
```

3. Extract genotypes

we will get a matrix, with each row represents one individual and each column a block in the genome in which no crossover event was identified in the population.

Here we decided to remove genotypes which swiched twice within 3 Mb.

```
out_3 <- Extract_all(chromosome = chr,id_all = id_all[index.keep],all_vcf = all_vcf[index.keep],gap=3e6
```

4. Format genotype to consecutive genomic bins

The genotypes we obtained might has too many bins due to gaps in the imputed data or imputation errors, which is not favoured in linkage map estimation. Here, we further averaged these bins into evenly spaced genomic bins.

```
chr.match <- read.table("/home/yanjun/projects/F2_seq/F2_re_seq/data/chr_id.match.txt",stringsAsFactors

chr.match <- read.table("/Volumes/office-home/impute/git/F2_re_seq/data/chr_id.match.txt",stringsAsFacto
length.chr <- ceiling(chr.match$Size.Mb.)
names(length.chr) <- chr.match$Name
mat_3 <- Mat2geno(out = out_3,id = id_all[index.keep],length.chr = length.chr)
```

5. Set bins spanning recombination break point to missing and format to Rqtl input format

```
mat.fam.re_3 <- Remove_het_mat(mat.fam = mat_3)
Export2rqtl(genoFile =mat.fam.re_3,phenoFile = pheFile2,output.name = paste0("/Users/yanjunzan/Document
```