Use Dmel DIMs - Tutorial

DEST

1/13/2021

#Description This R markdown documents a pipeline to implement the Demography Informative Markers DAPC model

Load Packages

```
library(SeqArray)
library(tidyverse)
library(magrittr)
library(reshape2)
library(data.table)
library(zoo)
library(adegenet)
```

Load data

```
### open GDS file
  genofile <-
seqOpen("/project/berglandlab/DEST/gds/dest.all.PoolSNP.001.50.10Nov2020.ann.gds"
)

### get target populations
  samps <- fread("/scratch/yey2sn/DEST/populationInfo/samps.csv")

### Load DIM Loci
  load("/scratch/yey2sn/DEST_DAPC/AIM_SNPs.Rdata")
```

Prepare SNP files

```
  samps <- rbind(samps[set=="DrosRTEC"],
                 samps[set=="DrosEU"],
                 samps[set=="dgn"]
                 )

### get subsample of data to work on
  seqResetFilter(genofile)
  seqSetFilter(genofile, sample.id=samps$sampleId)

  snps.dt <- data.table(chr=seqGetData(genofile, "chromosome"),
                        pos=seqGetData(genofile, "position"),
                        variant.id=seqGetData(genofile, "variant.id"),
                        nAlleles=seqNumAllele(genofile),
                        missing=seqMissing(genofile, .progress=T))

  ## choose number of alleles
 snps.dt <- snps.dt[nAlleles==2]
```

```r
seqSetFilter(genofile, sample.id=samps$sampleId, variant.id=snps.dt$variant.id)

snps.dt[,af:=seqGetData(genofile, "annotation/info/AF")$data]

### select sites
  seqSetFilter(genofile, sample.id=samps$sampleId,
               snps.dt[chr%in%c("2L", "2R", "3L",
"3R")][missing<.05][af>.2]$variant.id)

### get allele frequency data
  ad <- seqGetData(genofile, "annotation/format/AD")
  dp <- seqGetData(genofile, "annotation/format/DP")

  dat <- ad$data/dp
  dim(dat)

## Add metadata
    colnames(dat) <- paste(seqGetData(genofile, "chromosome"),
seqGetData(genofile, "position") , paste("snp", seqGetData(genofile,
"variant.id"), sep = ""), sep="_")

rownames(dat) <- seqGetData(genofile, "sample.id")

samples_to_remove = c(
                      "SIM",
                      "B",
                      "T"
                      )

dat_filt = dat[-which(rownames(dat) %in% samples_to_remove),]

left_join(data.frame(sampleId=rownames(dat_filt)), as.data.frame(samps)) ->
DEST_DGN_metadata
```

Generate metadata file

```r
AIMS_Subset %<>% separate(SNPid, into = c("chr","pos","variantID"), sep = "_",
remove = F) %>%   mutate(chr_pos = paste(chr, pos, sep = "_"))


#write.table( AIMS_Subset,
#             file = "~/Desktop/AIMS_Subset.txt",
#             sep = "\t",quote = F ,row.names = F, col.names = T, append = F)
```

Extract the DIM loci

```r
dat_filt %>%
  t() %>%
  as.data.frame() -> dat_filt_t

names(dat_filt_t) -> Sample_names


dat_filt_t %<>%
```

```
  mutate(SNP_id = rownames(.)) %>%
  separate(SNP_id, into = c("chr","pos","variantID"), sep = "_") %>%
  mutate(chr_pos = paste(chr, pos, sep = "_"))

dat_filt_t$chr_pos %in% AIMS_Subset$chr_pos %>% table

dat_filt_t %>%
  .[which(.$chr_pos %in% AIMS_Subset$chr_pos),] -> DIMS_loc_t

DIMS_loc = t(DIMS_loc_t[Sample_names]) %>% as.data.frame()
names(DIMS_loc) = dat_filt_t$chr_pos[which(dat_filt_t$chr_pos %in%
AIMS_Subset$chr_pos)]
```

Impute missing loci — as means

```
DIMS_loc_naimp = na.aggregate(DIMS_loc)

#save(DIMS_loc_naimp,
#     file="./DIMS_loc_naimp.Rdata")
```

Do DAPC analysis

```
#load("~/Desktop/DIMS_loc_naimp.Rdata")

samps$country = gsub("USA", "United States", samps$country)
samps$country = gsub("w501", "United States", samps$country)

rownames(DIMS_loc_naimp) ->  samples_in_DIMS

samps$country %>%
  table %>%
  .[which(. > 1)] %>%
  names -> count_to_use

samps %>%
  .[which(.$country %in% count_to_use),] -> samps_filt

samples=DIMS_loc_naimp %>%
  .[which(rownames(.) %in% samps_filt$sampleId),]

grps=samps_filt$country[which(samps_filt$sampleId %in% samples_in_DIMS)]


DAPC_model <- xvalDapc(samples,
                       grps,
     n.pca.max = 300,
     training.set = 0.9,
     result = "groupMean",
     center = TRUE,
     scale = FALSE,
     n.pca = NULL,
     n.rep = 30,
     xval.plot = TRUE)
```

Predict new samples

```
new_samples="SAMPLES TO PREDICT AS DATAFRAME"

#             chr_pos1 chr_pos2 ...
#Sample       0.0000    0.0000  ...

# The SNP positions must be same on the DAPC model and the samples to be
predicted

#               2L_5762  2L_10610   2L_19802   2L_27181  2L_28813
#AT_Mau_14_01 0.2105263 0.9696970 0.03333333 0.07894737 0.9210526
#AT_Mau_14_02 0.4615385 0.9333333 0.29411765 0.40384615 0.9791667
#AT_Mau_15_50 0.1807229 0.7575758 0.26785714 0.21782178 0.8300000
#AT_Mau_15_51 0.3333333 0.8235294 0.21052632 0.17757009 0.9154930
#AT_See_14_44 0.3030303 0.9074074 0.32000000 0.22500000 0.9375000

predict.dapc(DAPC_model$DAPC, newdata=new_samples)
```