qpAdm

Getting Started

Reading materials: Haak et al 2015 ; Harney et al 2021 ; ADMIXTOOLS 2 Tutorial

Prerequisites

```
#dataset
data=/mnt/NEOGENE1/toTransfer/workshopfiles/qp/data/qpadmdatCay

#software
qpAdm=/usr/local/sw/AdmixTools-7.0.2/bin/qpAdm
qpfstats=/usr/local/sw/AdmixTools-7.0.2/bin/qpfstats

#working folder
workfold=/PATH/TO/YOUR/FOLDER
```

Preparing files

Firstly, we need to prepare input files. Right pops should include reference populations (one pop per line) whereas the first population is the outgroup (An African for human populations).

```
cat > rightpops <<EOF
Mbuti
Ust_Ishim
Kostenki14.SG
MA1
Han
Papuan
Dai
Chukchi
Mixe
CHG
Natufian
WHG
AfontovaGora3
Iberomaurusian
EOF
```

Left pops consist of target (first line) and sources, again in the form of one pop per line.

```
cat > leftpops <<EOF
Anatolia_Catalhoyuk
Asikli
Levant_N
EOF
```

```
cat > workshop.par <<EOF
genotypename: ${data}.geno
snpname: ${data}.snp
indivname: ${data}.ind
popleft: leftpops
popright: rightpops
summary: YES
details: YES
allsnps: YES
inbreed: NO
EOF
```

```
cat > Catalqpadm.sh <<EOF
#!/bin/bash -l
#SBATCH -J qpAdm
#SBATCH -p chimp
#SBATCH -n 1
#SBATCH -t 5-00:00:00
#SBATCH -D $workfold
#SBATCH -o slurm-%j-%N-%u.out
#SBATCH -e slurm-%J-%N-%u.err
export PATH=$PATH:/usr/local/sw/AdmixTools-7.0.2/bin/
$qpAdm -p workshop.par > Catal2Way.log
EOF
```

```
sbatch Catalqpadm.sh
```

qpfstats

In the recent version of Admixtools (v7.0.2), another software called `qpfstats` calculates base statistics for qpAdm using the populations given. It is faster to calculate all these statistics first, then use the output as input for `qpAdm`. Firstly, you should prepare a file including one population per line, here we called `poplist`. Check the directory for the list of populations.

```
cat > allvsall.par <<EOF
indivname: ${data}.ind
snpname: ${data}.snp
genotypename: ${data}.geno
poplistname: poplist
fstatsoutname: fstats.txt
allsnps: YES
inbreed: NO
outpop: Mbuti
EOF
```

```
cat > allvsallqpfstats.sh <<EOF
#!/bin/bash -l
#SBATCH -J qpfstats
#SBATCH -p chimp
#SBATCH -n 1
#SBATCH -t 5-00:00:00
#SBATCH -D $workfold
```

```
#SBATCH -o slurm-%j-%N-%u.out
#SBATCH -e slurm-%J-%N-%u.err
export PATH=$PATH:/usr/local/sw/AdmixTools-7.0.2/bin/
$qpfstats -p allvsall.par > allvsall.log
EOF
```

PS: It takes long time, so please don't run this step by yourself now. We have the output files in the directory.

```
cat > workshopqpfstat.par <<EOF
popleft: leftpops
popright: rightpops
summary: YES
details: YES
allsnps: YES
inbreed: NO
fstatsname: /mnt/NEOGENE1/toTransfer/workshopfiles/qp/results/fstats.txt
EOF
```

```
cat > Catalqpfstat.sh <<EOF
#!/bin/bash -l
#SBATCH -J qpAdmFstat
#SBATCH -p chimp
#SBATCH -n 1
#SBATCH -t 5-00:00:00
#SBATCH -D $workfold
#SBATCH -o slurm-%j-%N-%u.out
#SBATCH -e slurm-%J-%N-%u.err
export PATH=$PATH:/usr/local/sw/AdmixTools-7.0.2/bin/
$qpAdm -p workshopqpfstat.par > Catal2Wayqpfstat.log
EOF
```

```
sbatch Catalqpfstat.sh
```

Inspect Output Files

There several options in R to parse `qpAdm` outputs. I modified output parser from ADMIXTOOLS 2 since we used `details: YES` option. Let's go to description of the output first, then inspect and visualize it using R. Supplementary File of Harney et al 2021 described the output well.

```
##create function for detailed qpadm output
parse_qpadm_output_detail = function(outfile, detailed = T) {
  # reads qpAdm output file

  dat = read_lines(outfile)

  lstart = str_which(dat, 'left pops:')[1]+1
  rstart = str_which(dat, 'right pops:')[1]+1
  lend = str_which(dat, '^$') %>% magrittr::extract(. > lstart) %>% head(1)-1
  rend = str_which(dat, '^$') %>% magrittr::extract(. > rstart) %>% head(1)-1
  coefstart = str_which(dat, '^best coefficients:')[1]
  sigstart = str_which(dat, 'fixed pat')[1]+1
  sigend = str_which(dat, '^best pat:')[1]-1
  target = dat[lstart]
  left = dat[lstart:lend][-1]
  right = dat[rstart:rend]
  if (detailed){
    coefs = dat[c(coefstart:(coefstart+1),coefstart+3)]
  } else {
    coefs = dat[c(coefstart:(coefstart+2))]
  }
  coefs %<>% str_split(’ +’) %>%
    map(~tail(., length(left)+1) %>% head(-1) %>% as.numeric %>% set_names(left)) %>%
    set_names(c(’weight’, ’mean’, ’se’)) %>% as_tibble %>% mutate(z = mean/se)
  weights = tibble(target, left) %>% bind_cols(coefs)

  popdrop = do.call(rbind, str_split(dat[sigstart:sigend], ’ +’)) %>%
    as.data.frame(stringsAsFactors=F) %>% select(-1) %>%
    magrittr::set_colnames(c(’pat’, ’wt’, ’dof’, ’chisq’, ’p’, left, ’feasible’)) %>%
    mutate(feasible = feasible != ’infeasible’, across(!c(’pat’, ’feasible’), as.numeric)) %>% as_tibble

  namedList(weights, popdrop)
}
```

Generally we have multiple results to inspect. The following R script reads multiple outputs of qpAdm and visualizes them.

```
library(tidyverse)
library(admixtools)
parse_qpadm_output_detail("/mnt/NEOGENE1/toTransfer/workshopfiles/qp/results/Catal2way.log")
```

```
##List files ending with .log in a certain folder
logfiles <- list.files("/mnt/NEOGENE1/toTransfer/workshopfiles/qp/results",
                       pattern = "*.log", full.names = T)

##Parse listed files
logread <- logfiles %>% map(parse_qpadm_output_detail)

namevec <- str_remove(basename(logfiles), ".log")

logread <- logread %>%
  setNames(namevec)

##Arrange p-value table
allp <- logread %>%
  # extract second tibble from each nested list
  map(‘[[‘, 2) %>%
  # slice the first row
  map(slice_head)  %>%
  # bind rows together and select needed columns
  bind_rows(.id = "model") %>% select(model, p, feasible)

##Arrange weight table and merge with p-values
allweight <- logread %>%
  # extract first tibble from each nested list
  map(‘[[‘, 1) %>%
```

```r
  # bind rows together
  bind_rows(.id = "model") %>%
  left_join(., allp) %>%
  mutate(feasible = ifelse(p > 0.05 & feasible, TRUE,
                           ifelse(p > 0.01 & feasible, "partial", FALSE))) %>%
  mutate_at(c(4:8), as.numeric) %>%
  filter(feasible != FALSE) %>%
  group_by(model) %>%
  filter(all(abs(z)>2)) %>%
  arrange(factor(left, levels = rev(c("Asikli", "Levant_N")))) %>%
  mutate(sdpos = cumsum(weight))

ggplot(allweight, aes(x = weight, y = model, fill = left)) +
  geom_col(position = position_stack(), size = 0.5, color = "black") +
  #scale_fill_manual("", values = MetBrewer::met.brewer("Hokusai3",2)) +
  theme_minimal()+
  geom_errorbarh(aes(xmin = sdpos-se,
                     xmax = sdpos), height = 0.1, size = 0.5) +
  theme(axis.title = element_blank(),
        panel.grid = element_blank(),
        axis.text.x = element_blank(),
        legend.position = "bottom",
        panel.border = element_blank(),
        axis.text.y = element_text(angle = 90, hjust = 0.5)
  )
```