# Example analyses with **polyfreqs**

*PD Blischak, LS Kubatko and AD Wolfe*

## *Simulation study of population differentiation*

To study population differentiation we simulated two hypothetical populations of tetraploids. Sequencing reads were simulated as follows:

- Population 1: allele frequencies were drawn for 2000 loci from a Beta(1.25, 1.25) followed by simulated reference and total read data for 100 individuals with 15x coverage per locus using the `sim_reads` function in **polyfreqs**. The files are named `tot_reads_pop1.txt` and `ref_reads_pop1.txt`. They are formatted with column headers with a name for each locus (loc1, ..., loc2000) and have row names for each individual (ind1, ..., ind100). This is the format that the data need to be in for running through **polyfreqs**.
- Population 2: same as population 1 but allele frequencies were drawn from a Beta(5, 5). The files are named in the same fashion: `tot_reads_pop2.txt` and `ref_reads_pop2.txt`.

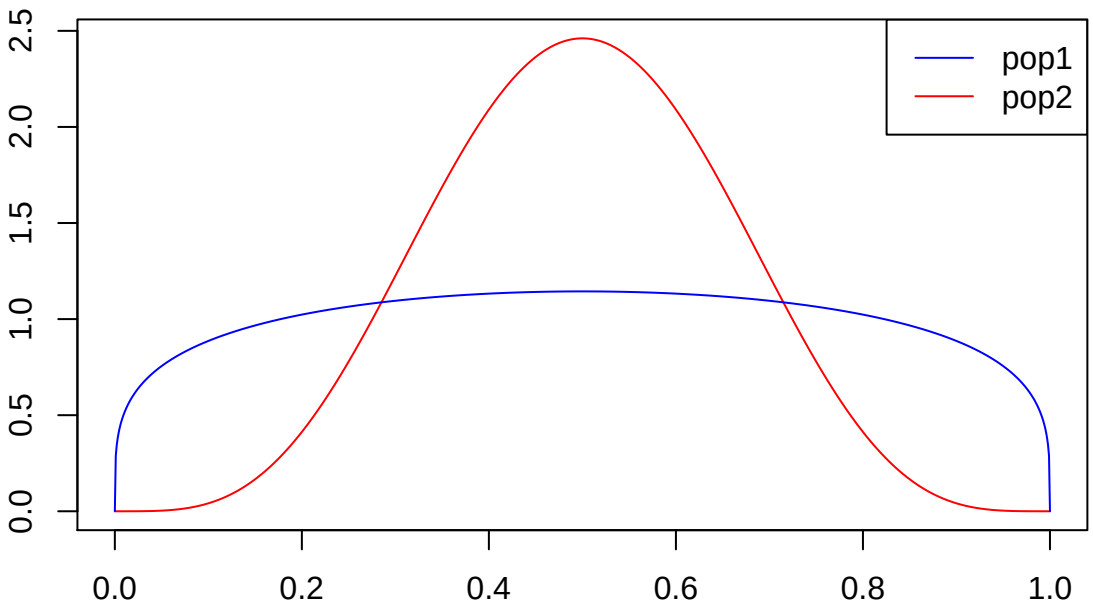
If you wish to re-simulate similar data, use the following code:

```
# Load polyfreqs
library(polyfreqs)

# Simulate allele frequencies from the Beta distributions
freqs_pop1 <- rbeta(2000, 1.25, 1.25)
freqs_pop2 <- rbeta(2000, 5, 5)


# Simulate the read data
pop1_dat <- sim_reads(freqs_pop1, 100, coverage=15, ploidy=4, error=0.01)
pop2_dat <- sim_reads(freqs_pop2, 100, coverage=15, ploidy=4, error=0.01)
```

```
# Vizualize the difference in the distribution of allelic diversity
# btwn pop1 and pop2
plot(x=seq(0, 1, length=1000), dbeta(seq(0, 1, length=1000), 5, 5), type="l", col="red", ylab="", xlab=
     main="Distribution of allelic diversity btwn pop1 (blue) and pop2 (red)")
lines(x=seq(0, 1, length=1000), dbeta(seq(0, 1, length=1000), 1.25, 1.25), col="blue")
legend(x="topright",legend=c("pop1","pop2"), col=c("blue", "red"), lty=1)
```




Next we will run **polyfreqs** for population 1. First, we'll read in the data, convert the tables into matrices and then run them with `genotype=TRUE` to get the genotype samples and provide it with a name for the folder into which the genotype files will be output (`geno_dir="./pop1_genotypes/"`). It's important that you remember to specify the directory name in this way (with the `./` and `/`). This tells the program to make the folder in the current working directory and then to make all new files within that directory that it created. With 2000 loci and 100 individuals, this analysis will take some time (on the order of a few hours).

```
# Read in data using read.table. Remember the row.names and header
pop1_tot_table <- read.table("tot_reads_pop1.txt", row.names=1, header=T)
pop1_ref_table <- read.table("ref_reads_pop1.txt", row.names=1, header=T)

#Convert the tables to matrices
pop1_tot_mat <- as.matrix(pop1_tot_table)
pop1_ref_mat <- as.matrix(pop1_ref_table)

# Run through polyfreqs with genotypes=T
# and geno_dir="pop1_genotypes"
#
# load polyfreqs if you haven'r done so already
# library(polyfreqs)
polyfreqs(pop1_tot_mat, pop1_ref_mat, ploidy=4,
          iter=100000, genotypes=T, geno_dir="./pop1_genotypes/", outfile="pop1_mcmc.out")
```

With the analysis for population 1 completed, we can analyze population 2.

```
# Read in data using read.table. Remember the row.names and header
pop2_tot_table <- read.table("tot_reads_pop2.txt", row.names=1, header=T)
pop2_ref_table <- read.table("ref_reads_pop2.txt", row.names=1, header=T)

#Convert the tables to matrices
pop1_tot_mat <- as.matrix(pop2_tot_table)
```

```r
pop1_ref_mat <- as.matrix(pop2_ref_table)

# Run through polyfreqs with genotypes=T
# and geno_dir="pop2_genotypes"
polyfreqs(pop2_tot_mat, pop2_ref_mat, ploidy=4,
          iter=100000, genotypes=T, geno_dir="./pop2_genotypes/", outfile="pop2_mcmc.out")
```

### Downstream analyses

With

## *Evaluating model adequacy in autotetraploid potato* (Solanum tuberosum)

```r
# Using autetraploid potato data from the fitTetra package
#
# If not installed, install it using:
# install.packages("fitTetra")
#
# Then load the data
library(fitTetra)
data(tetra.potato.SNP)

# Get the names of the individuals and loci
samples <- unique(tetra.potato.SNP$SampleName)
markers <- unique(tetra.potato.SNP$MarkerName)

# Initialize x and y matrices -- x will be the reference allele
potato_mat_x <- matrix(NA, nrow=length(unique(tetra.potato.SNP$SampleName)),
                   ncol=length(unique(tetra.potato.SNP$MarkerName)))
rownames(potato_mat_x) <- samples
colnames(potato_mat_x) <- markers

potato_mat_y <- matrix(NA, nrow=length(unique(tetra.potato.SNP$SampleName)),
                   ncol=length(unique(tetra.potato.SNP$MarkerName)))

# Get the counts from the data frame
for(i in 1:dim(potato_mat_x)[1]){
    tmp <- subset(tetra.potato.SNP, SampleName==samples[i])
    potato_mat_x[i,] <- tmp$X_Raw
    potato_mat_y[i,] <- tmp$Y_Raw
}

# Get the total counts as the sum of x and y and give row and column names
potato_mat_tot <- potato_mat_x + potato_mat_y
rownames(potato_mat_tot) <- samples
colnames(potato_mat_tot) <- markers

# print the tables to file in a format suitable for polyfreqs
```

```
write.table(potato_mat_x, file="potato_ref_reads.txt",quote=F,sep="\t")
write.table(potato_mat_tot, file="potato_tot_reads.txt",quote=F,sep="\t")
```

With the data in the correct format we can go ahead and run **polyfreqs**. We will again save the genotypes as we go and will put them in the folder `potato_genotypes/`.

```
# Read in data using read.table. Remember the row.names and header
potato_tot_table <- read.table("potato_tot_reads.txt", row.names=1, header=T)
potato_ref_table <- read.table("potato_ref_reads.txt", row.names=1, header=T)

#Convert the tables to matrices
potato_tot_mat <- as.matrix(potato_tot_table)
potato_ref_mat <- as.matrix(potato_ref_table)

# Run through polyfreqs with genotypes=T
# and geno_dir="./pop2_genotypes/"
polyfreqs(potato_tot_mat, potato_ref_mat, ploidy=4, iter=100000,
          genotypes=T, geno_dir="./potato_genotypes/", outfile="potato_mcmc.out")
```