Introduction

Module: Plant Genetic Resources (3502-470)

Karl Schmid, University of Hohenheim

23 May 2025

Table of contents

The purpose of this computer lab is to introduce the use of coalescent theory for the analysis of genetic diversity. In particular we focus on using simulations to introduce the key concepts. Coalescent simulations make use of the coalescent theory introduced in the lecture. They are frequently used to simulate data under certain models and to compare the simulated data to results from empirical data. We will use the software `ms` (Hudson, 2002) as implemented in the R package `phyclust` to conduct coalescent simulations and to compare the results to theoretical expectations.

R packages for this computer lab

For this computer lab the following R packages are required:

- `phyclust`
- `gap`
- `pegas`

```
library(phyclust)
library(gap)
library(pegas)
```

Makes sure that the packages are installed and loaded.

Simulation of coalescent trees

All simulations will be conducted by using the command `ms` from the package `phyclust`. It has three arguments

`ms (number of alleles, number of simulated trees, further ms arguments)`

Additional arguments are always provided as a single text string (i.e., "text"). For more information check `?ms`.

We start with looking at several coalescent trees. Since coalescent trees are randomly generated, there might be huge differences between trees. The `ms` option to generate trees is `"-T"`.

```
mstrees <- ms(6, 6, "-T")                    # produces trees in Newick format
par(mfrow = c(2,3))
mstrees <- read.tree(text = mstrees, skip = 3) # Reads in trees, produces a list of trees
```

```
for (tr in mstrees) {                          # Function to plot trees
  plot.phylo(tr)
  add.scale.bar()
}
par(mfrow=c(1,1))
```

> **Exercise**
>
> 1. Use the `?ms` function to find out what the parameter `-T` means.
> 2. Compare the trees. Do they all have the same height (time back to the MRCA)? Hint: Look at the scales of the trees
> 3. Do you observe differences between the internal and the terminal branches? How can they be explained with coalescent theory?

In class we discussed the effect of different processes such as population growth, selection, etc. on the shape of a gene genealogy. To study the effect of gene genealogies, we run a few simulations.

> **Exercise**
>
> 1. Check out the parameter `-G` in the `ms` command. How well is it defined?
> 2. Copy and paste the above function, add the `-G` option.
> 3. Choose different values of G and check how the shape of the gene genealogy is changing.

```
mstrees <- ms(6, 6, "-T -G 1000")                    # produces trees in Newick format
par(mfrow = c(2,3))
mstrees <- read.tree(text = mstrees, skip = 3) # Reads in trees, produces a list of trees
for (tr in mstrees) {                          # Function to plot trees
  plot.phylo(tr)
  add.scale.bar()
}
par(mfrow=c(1,1))
```

Coalescent times

By assuming a population of size N, the time (in generations) to the most recent common ancestor (MRCA) for a sample of n individuals can be calculated as:

$$E(T_{MRCA}) = 4N(1 - \frac{1}{n}).$$

We can calculate the expected total length of the sample's genealogy as:

$$E(T_c) = 4N \sum_{i=2}^{n} \frac{1}{i-1}.$$

We will now compare the theoretical mean values with simulations under the coalescent model. This can be done with the option `-L`.

Note that we also specify a mutation rate, although `ms` doesn't need it to compute the genealogies and the mutation rate/structure does not have any

influence on the simulated genealogical trees. This is just done so that we can easily transform the data using `read.ms.output`.

The parameter `-t` gives the value of $\theta = 4N_e\mu$ used in the simulation.

```
mslengths <- ms(10, 1000, "-t 1 -L") # Generate coalescent trees and compute their lengths
mslengths <- read.ms.output(mslengths, is.file=FALSE) # Transform the data
mslengths <- mslengths$times      # Just keep the lengths
par(mfrow = c(1,2))
hist(mslengths[,1], xlab="Time back to MRCA",main="")
abline(v=mean(mslengths[,1]),col="blue",lwd=2)
hist(mslengths[,2], xlab="Total length of the genealogy",main="")
abline(v=mean(mslengths[,2]),col="blue",lwd=2)
par(mfrow=c(1,1))
```

For interpretation, the time is expressed in units of $4N_0$ generations, where N_0 is the population size at point $t = 0$ in the model. Therefore, the estimated coalescent time has to be multiplied by two.

> **Exercise**
>
> Increase the number of alleles. How does the average time to the MRCA and the average total length of the genealogy change?

Coalescent simulation

Next we will simulate data sets with a given $\theta_W = 15$:

```
sim <- ms(25, 1000, paste("-t", 15))        # Simulate 1000 data sets
seq <- read.ms.output(sim, is.file = FALSE) # Split different types of information
gametes <- seq$gametes                      # Take only the gamete information
```

We need to recode the output into a `DNAbin` object (convert the simulations into DNA sequence alignments).

TODO: Give info about length of simulated sequence

```
f.recode <- function(m){ # Function to recode sequence as mock DNA sequence
  m[m==0] <- "a"
  m[m==1] <- "c"
  as.DNAbin(t(m))
}

gametesseq <- lapply(gametes, f.recode) # Recode the sequences produced by ms
```

Now we can calculate various statistics for the simulated data sets and plot their distributions.

We first plot the number of segregating sites in each simulation.

```
library(ggplot2)
# Compute segregating-site counts from each simulation
seg_counts <- sapply(gametesseq, function(y) length(seg.sites(y)))
df <- data.frame(seg_sites = seg_counts)
```

```
# ggplot2 histogram
ggplot(df, aes(x = seg_sites)) +
  geom_histogram(bins = 20, fill = "gray", color = "black") +
  labs(
    title = "Histogram of segregating sites per simulation",
    x = "Segregating sites",
    y = "Frequency"
  ) +
  theme_minimal()
```

When we plot the distribution of the Watterson estimator θ_W for each data set, we can also add the value of θ_W that we used as input parameter:

```
# Compute Watterson's theta for each simulation
theta_vals <- sapply(gametesseq, theta.s)
df_theta <- data.frame(theta = theta_vals)

# ggplot2 histogram with vertical line at theta = 15
ggplot(df_theta, aes(x = theta)) +
  geom_histogram(bins = 20, fill = "gray", color = "black") +
  geom_vline(xintercept = 15, color = "blue", linewidth = 0.2) +
  geom_vline(xintercept = mean(df_theta$theta, color= "red", linewidth=0.2))
  labs(
    title = "Histogram of Watterson's estimator",
    x = expression(theta),
    y = "Frequency"
  ) +
  theme_minimal()
```

And the same for the nucleotide diversity π:

```
hist(sapply(gametesseq,function(x){nuc.div(x)*ncol(x)}),breaks=20,
     main="Histogram of nucleotide diversity, pi", xlab = expression(pi))
abline(v=15,col="blue")
```

```
# Compute nucleotide diversity times sequence length for each simulation
pi_values <- sapply(gametesseq, function(x) nuc.div(x) * ncol(x))
df_pi <- data.frame(pi = pi_values)

# ggplot2 histogram with a vertical line at pi = 15
ggplot(df_pi, aes(x = pi)) +
  geom_histogram(bins = 20, fill = "gray", color = "black") +
  geom_vline(xintercept = 15, color = "blue", size = 1) +
  labs(
    title = "Histogram of nucleotide diversity, pi",
    x = expression(pi),
    y = "Frequency"
  ) +
  theme_minimal()
```

Exercise

1. [Difficult] The values of θ_W and π are rather high. Do you have an explanation why this could be the case? Hint: Inspect the `DNAbin` object and check for example the length of the simulated DNA sequence.
2. Does the input value of θ_W fit with the distributions? Why? Why not?

References

Hudson RR. 2002. Generating samples under a Wright–Fisher neutral model of genetic variation. *Bioinformatics* **18**:337–338. doi:10.1093/bioinformatics/18.2.337