chapter3 R answers

Samuel

28/08/2019

Discussion questions

1. The Hardy-Weinberg model hardly applies to any natural population. Why is it nevertheless useful?
2. In a small population, genetic drift will lead to a reduction in the frequency of heterozygotes relative the Hardy-Weinberg expectation - is this true or false? Discuss why.
3. Why is it that when an allele goes to fixation in a population, there are no heterozygotes but there is also no deviation from the Hardy-Weinberg expectation?

R coding questions

4. Write a simple R function called square that will return the square of any numeric variable it is given.

```
square = function(numvar) {
  res2 = numvar * numvar
  return(res2)
}
```

5. You genotype a species of grasshoppers along a north south transect across the European Alps. Near Munich, Germany, north of the Alps you sample 120 individuals; near Innsbruck, Austria, within the Alps you sample 122 individuals; near Verona, Italy, south of the Alps you sample 118 individuals. You find the following number of each genotype.

   Munich - 6 (A1A1), 33 (A1A2), 81 (A2A2)

   Innsbruck - 20 (A1A1), 59 (A1A2), 43 (A2A2)

   Verona - 65 (A1A1), 39 (A1A2), 14 (A2A2)

   Using the R code we learnt during the tutorial, calculate the allele frequencies in each population. Then test whether there is a deviation from Hardy-Weinberg Equilibrium in each of them.

```
Ng_obs = list(Mun=c(6,33,81),
              Inn=c(20,59,43),
              Ver=c(65,39,14))
obs_freq = lapply(Ng_obs, function(x) {
  A1 = (2*x[1]+x[2])/(2*sum(x))
  A2 = (2*x[3]+x[2])/(2*sum(x))
  return(c(A1,A2))
})
obs_freq
```

```
## $Mun
## [1] 0.1875 0.8125
##
## $Inn
## [1] 0.4057377 0.5942623
##
## $Ver
```

```
## [1] 0.7161017 0.2838983
```

```
exp_freq = lapply(obs_freq, function(x) {
  homo1 = x[1]^2
  het = 2*x[1]*x[2]
  homo2 = x[2]^2
  return(c(homo1,het,homo2))
})
exp_freq
```

```
## $Mun
## [1] 0.03515625 0.30468750 0.66015625
##
## $Inn
## [1] 0.1646231 0.4822292 0.3531477
##
## $Ver
## [1] 0.51280164 0.40660011 0.08059825
```

```
lapply(seq_along(Ng_obs), function(x) {
  chisq.test(Ng_obs[[x]], p = exp_freq[[x]])
})
```

```
## Warning in chisq.test(Ng_obs[[x]], p = exp_freq[[x]]): Chi-squared
## approximation may be incorrect
```

```
## [[1]]
##
##  Chi-squared test for given probabilities
##
## data:  Ng_obs[[x]]
## X-squared = 1.1393, df = 2, p-value = 0.5657
##
##
## [[2]]
##
##  Chi-squared test for given probabilities
##
## data:  Ng_obs[[x]]
## X-squared = 0.00099522, df = 2, p-value = 0.9995
##
##
## [[3]]
##
##  Chi-squared test for given probabilities
##
## data:  Ng_obs[[x]]
## X-squared = 4.1326, df = 2, p-value = 0.1267
```

```
sapply(seq_along(exp_freq), function(y) {
  exp_freq[[y]]*sum(Ng_obs[[y]])
})
```

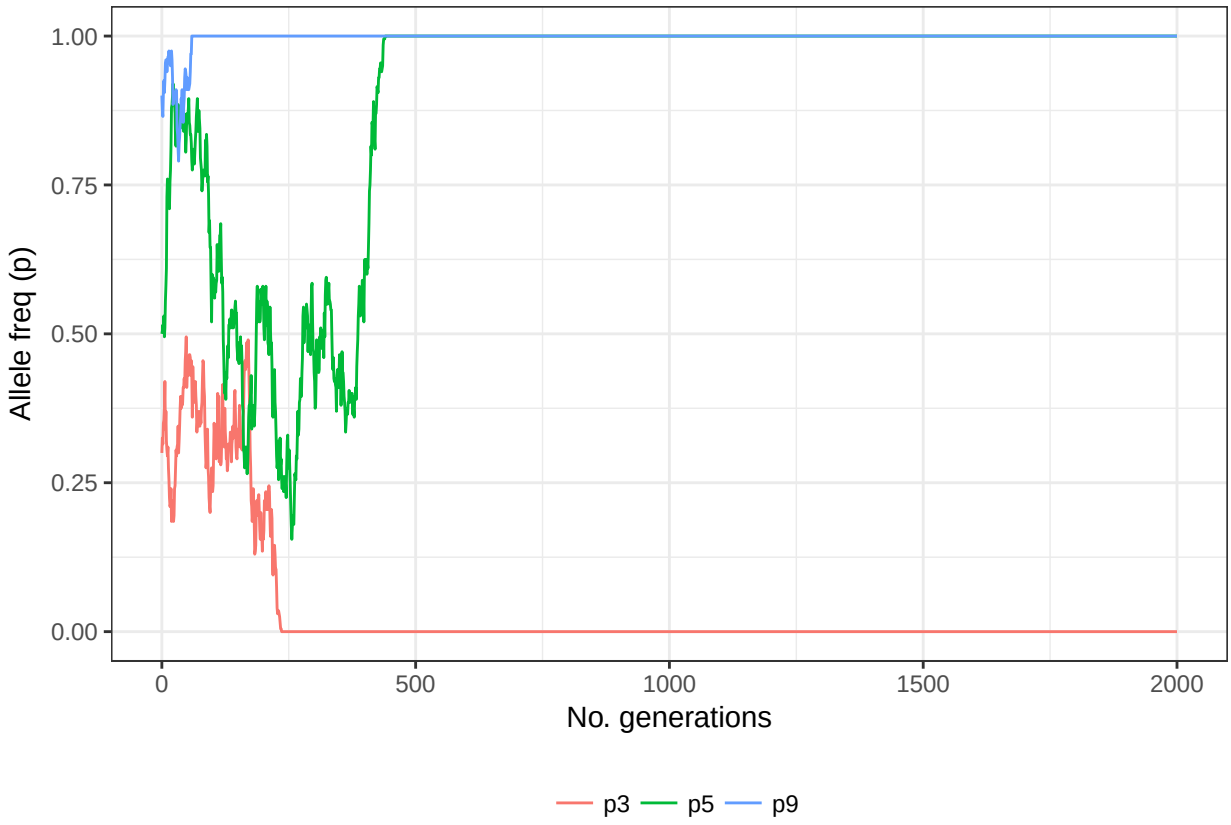
```
##          [,1]     [,2]      [,3]
## [1,]  4.21875 20.08402 60.510593
## [2,] 36.56250 58.83197 47.978814
## [3,] 79.21875 43.08402  9.510593
```

6. Using the functions we wrote during the tutorial, simulate drift over 2000 generations for a population of 100 individuals. How does altering p from 0.3, 0.5 and 0.9 alter the outcome of the simulations? Which is more likely to go to fixation? a custom function for simulating drift across multiple generations

```
drift_sim <- function(N, p, ngen){
  # initialise p
  p_init <- p
  # sample across all the generations
  pvec <- sapply(1:ngen, function(x){
    pA <- rbinom(1, 2*N, p)
    p <<- pA/(2*N)
  })
  # create a vector of p over time
  p <- c(p_init, pvec)
  # write out
  return(p)
}
# simulate drift for three different p
a <- sapply(c(0.3, 0.5, 0.9), drift_sim, N = 100, ngen = 2000)
# rename matrix
colnames(a) <- c("p3", "p5", "p9")
# get number of generations
g <- seq(0, 2000, 1)
# combine into a tibble
library(tibble)
library(tidyr)
mydrift <- as.tibble(cbind(g, a))
head(mydrift)
```

```
## # A tibble: 6 x 4
##       g    p3    p5    p9
##   <dbl> <dbl> <dbl> <dbl>
## 1    0. 0.300 0.500 0.900
## 2    1. 0.325 0.505 0.875
## 3    2. 0.315 0.515 0.865
## 4    3. 0.350 0.505 0.905
## 5    4. 0.355 0.530 0.925
## 6    5. 0.375 0.495 0.905
```

```
mydrift <- gather(mydrift, key = "init_p", value = "p", -g)
# plot data
library(ggplot2)
p <- ggplot(mydrift, aes(g, p, colour = init_p))
p <- p + geom_line()
p <- p + xlab("No. generations") + ylab("Allele freq (p)")
p + theme_bw() + theme(legend.position = "bottom", legend.title = element_blank())
```


1.00
0.75
0.50
0.25
0.00
Allele freq (p)
0
500
1000
1500
2000
No. generations
p3
p5
p9