

Petras1967_Transformation

Manisha Munasinghe

2024-03-27

I. Mathematical Derivation

In Munasinghe and Brandvain 2024 (in Revision), we extend Charlesworth 1994's adaptive model of lethal *t*-haplotype evolution with reproductive compensation to consider the influence of inbreeding via sib-mating on his results. We compare our results to Petras 1967. Here, Petras considered the evolution of a lethal *t*-haplotype into a naive population with inbreeding. The coefficient of inbreeding used was a measure of the Wahlund effect (i.e., the numerical reduction of heterozygotes due to population subdivision). To compare our results, we must translate our parameter representing the proportion of sib-mating (p_{sib}) into his coefficient of inbreeding (F).

Using standard path diagrams, we know that the probability that alleles in an individual who is a product of brother-sister mating are immediately IBD is $\frac{1}{4}$ and the probability that they are not is $\frac{3}{4}$. Thus, the overall probability of IBD in this mating equals $\frac{1}{4} * 1 + \frac{3}{4} * f$, where f is the coefficient of inbreeding. Consequently, after a single generation of brother-sister mating with probability p_{sib} , the deviation from HWE is $f' = p_{sib} (\frac{1}{4} * 1 + \frac{3}{4} * f)$. At equilibrium, $f' - f = 0 = p_{sib} (\frac{1}{4} * 1 + \frac{3}{4} * f) - f$. Solving this for f , we get the following.

$$f' - f = 0 = p_{sib} (\frac{1}{4} * 1 + \frac{3}{4} * f) - f$$

$$f = (\frac{p_{sib}}{4} + \frac{3fp_{sib}}{4})$$

$$f - \frac{3fp_{sib}}{4} = f(1 - \frac{3p_{sib}}{4}) = f(\frac{4-3p_{sib}}{4}) = \frac{p_{sib}}{4}$$

$$f = \frac{p_{sib}}{4} * \frac{4}{4-3p_{sib}} = \frac{p_{sib}}{4-3p_{sib}}$$

Therefore, we show that we can approximate the inbreeding coefficient from the proportion of sib-mating as $f = \frac{p_{sib}}{4-3p_{sib}}$

II. Comptuer Simulation

We simulated a single population with under variable proportions of sib-mating following the method used in our paper. From this, we can calculate observed genotype frequencies and calculate the deviation from the expected genotype frequencies under HWE. This deviation is analogous to the inbreeding coefficient used in Petras 1967. We did this for a fixed wild type allele frequency, but the trend of our results does not depend on this allele frequency. We confirm with our simulated results that our analytical equation for obtain f from p_{sib} perfectly aligns with what we calculate from our simulated results.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.4.4      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(viridis)
```

```
## Loading required package: viridisLite
```

```
fams <- crossing(mat = c("AA", "Aa", "aA", "aa"),
                 pat = c("AA", "Aa", "aA", "aa")) %>%
  mutate(p_A_mat = str_count(mat, "A")/2,
         p_A_pat = str_count(pat, "A")/2,
         kid_aa = (1-p_A_pat) * (1-p_A_mat),
         kid_aA = p_A_pat * (1-p_A_mat),
         kid_Aa = p_A_mat * (1-p_A_pat),
         kid_AA = p_A_mat * p_A_pat) %>%
  select(-p_A_mat, -p_A_pat)

inbreeding <- function(pA, psib, ngen = 10){
  geno.freqs <- c(aa = (1-pA)^2, aA = pA*(1-pA), Aa = pA*(1-pA), AA = pA^2)
  last_fams <- fams %>%
    mutate(fam_freqs = geno.freqs[pull(fams,mat)] * geno.freqs[pull(fams,pat)] )
  for(g in 1:ngen){

    pfam_given_RM <- geno.freqs[pull(fams,mat)] * geno.freqs[pull(fams,pat)]

    ### INBREEDING ###
    # find p fam given geno
    p_fam_given_genos <- last_fams %>%
      pivot_longer(contains("kid"),
                   values_to = "geno_freq",
                   names_to = "kid_genos",
                   names_prefix = "kid_") %>%
      mutate(geno_freq = fam_freqs * geno_freq) %>%
      group_by(kid_genos) %>%
      mutate(rel_freq = geno_freq / sum(geno_freq)) %>%
      mutate(fam = paste(mat, pat, sep="_")) %>%
      select(rel_freq, fam, kid_genos) %>%
      pivot_wider(names_from=fam, values_from = rel_freq) %>% ungroup()

    pfam_given_SM <- (select(p_fam_given_genos, -kid_genos) %>% as.matrix()) %*%
      (select(last_fams, contains("kid")) %>% as.matrix()) %>%
      data.frame() %>%
      mutate(mat = c("aa", "aA", "aA", "AA")) %>%
      pivot_longer(-mat, values_to = "pat_freq_given_genos") %>%
      mutate(mat_freq_given_genos = geno.freqs[pull(fams,mat)]) %>%
      mutate(p = pat_freq_given_genos * mat_freq_given_genos) %>%
      pull(p)
```

```

last_fams <- last_fams %>%
  mutate(fam_freqs = pfam_given_RM * (1-psib) + pfam_given_SM * psib)

new_genotype_freqs <- last_fams%>%
  select(-mat,-pat )%>%
  pivot_longer(contains("kid"),values_to = "genotype_freqs",names_to = "genotype",names_prefix = "kid_")%>%
  group_by(genotype)%>%
  summarise(freqs = zapsmall(sum(fam_freqs*genotype_freqs)))

genotype_freqs <- pull( new_genotype_freqs)
names(genotype_freqs) <- pull( new_genotype_freqs,genotype)
genotype_freqs <- genotype_freqs/ sum( genotype_freqs )
#print(c(genotype_freqs , sum(genotype_freqs)))
}
return(genotype_freqs)
}

fs <- sapply( seq(0,1,.01), function(psib){
  #print(psib)
  tmp <- round(inbreeding(pA=.3,psib = psib,ngen = 100), digits = 4)
  return( data.frame(f= (tmp[[1]] - .3^2) / (.3*.7), psib))
},simplify = FALSE
)

bind_rows(fs)%>%
  mutate(maths = (psib)/(4-3*psib))%>%
  pivot_longer(-psib)%>%
  mutate(name = ifelse(name == "f" , "simulation",name))%>%
  dplyr::mutate(name=case_when(
    name=='maths' ~ 'Analytical Expectation',
    TRUE ~ 'Simulated Result'
  )) %>%
  ggplot(aes(x = psib, y = value, color=name))+
  geom_line(data = . %>% filter(name == "Analytical Expectation"))+
  geom_point(data = . %>% filter(name == "Simulated Result"),size = .3)+
  scale_color_manual(values=c('#BDC696','#13140f')) +
  labs(x='psib: Proportion of Sib-Mating',y='F: Inbreeding Coefficient',color='Type')

```

