

Evolution of the recombination rate

Frederic Michaud

2018-04-23

Introduction

It is generally believed that, due to Sex-antagonist selection, the recombination on the sex-chromosome tends to decrease. However, in a numerous number of species, recombination is actually kept on sexual chromosome. Several explanation have been given for this maintenance. One scenario it that this is due to finite population effect like the muller-ratchet effect to purge deleterious mutation or the Hill–Robertson effect. But even in infinite population, overdominance in male can produce under some circonstances a increase in recombination. In this vignette, we will explore such a scenario.

Sex-antagonistic selection

We will start by exploring a case where recombination is actually removed from the population. We will just set a simple model, with a sd locus linked to a sex-antagonistic locus. The distance can be modified by a locus, fully linked to the sd locus. We call this locus the modifier. This is done by specifying the vector `recombination.modifier` for each genotype of the modifier locus. This vector gives a pre-factor by which the distance specified in the genome will be multiply. Notice that only the overall scaling can be changed. Here, we see that a “-” genotype of the modifier leads to a stop in recombination, while a “++” leave the recombination unchanged.

```
locus1 = create.locus(allele1=c(1,1),
                      allele2 = c(1,2),
                      sd = c(0,1),
                      fitness.male=c(1,1),
                      fitness.female=c(1,1),
                      allele.name = c("x","y"))
locus2 = create.locus(allele1 = c(1,1,2),
                      allele2 = c(1,2,2),
                      recombination.modifier = c(0,0.5,1),
                      allele.name = c("-", "+")
                      )
locus3 = create.locus(allele1= c(1,1,2),
                      allele2 = c(1,2,2),
                      fitness.female = c(0.9,0.95,1),
                      fitness.male = c(1,0.95,0.9),
                      allele.name = c("M","F"))
genome = create.genome(list(locus1,locus2,locus3),
                        male.recombination = c(0,0.01),
                        female.recombination = c(0,0.01))

print(genome)
#> locus 1
#> allele1 allele2 fitness.male fitness.female modifier sd
#> 1      x      x           1           1           1 0
#> 2      x      y           1           1           1 1
#> * * * * * * * * * * * * * * * * * * * * * * * *
```

```

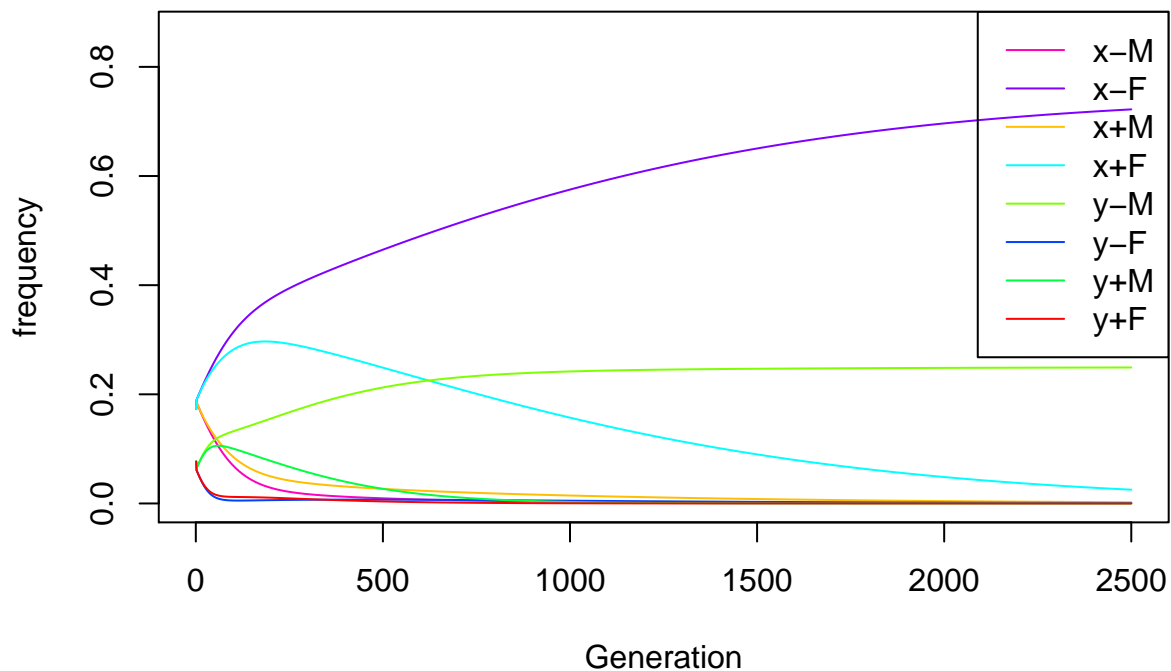
#> locus 2
#> allele1 allele2 fitness.male fitness.female modifier
#> 1      -      -          1          1          0.0
#> 2      -      +          1          1          0.5
#> 3      +      +          1          1          1.0
#> * * * * *
#> locus 3
#> allele1 allele2 fitness.male fitness.female modifier
#> 1      M      M          1.00         0.90         1
#> 2      M      F          0.95         0.95         1
#> 3      F      F          0.90         1.00         1
#> * * * * *

```

```

freq <- compute.frequency.evolution(genome,generations = 2500)
plot.haplotype.frequency(genome,freq)

```



We see that in this case, and as expected, the recombination is halted, so that y can fix the male beneficial allele while x can fix the female beneficial one.

overdominance in male

According to S.P. Otto, <https://onlinelibrary.wiley.com/doi/pdf/10.1111/jeb.12324> , in a scenario where there is overdominance in male, with different alleles being selected on the X chromosom carried by male and by female, selection would actually be selected. We can check if this is true using the same scenario as before changing only the strength of the fitness in male and in female. Notice that in male, heterozygothe are favoured.

```

locusssd = create.locus(allele1=c(1,1),
                        allele2 = c(1,2),
                        sd = c(0,1),
                        fitness.male=c(1,1),
                        fitness.female=c(1,1),

```

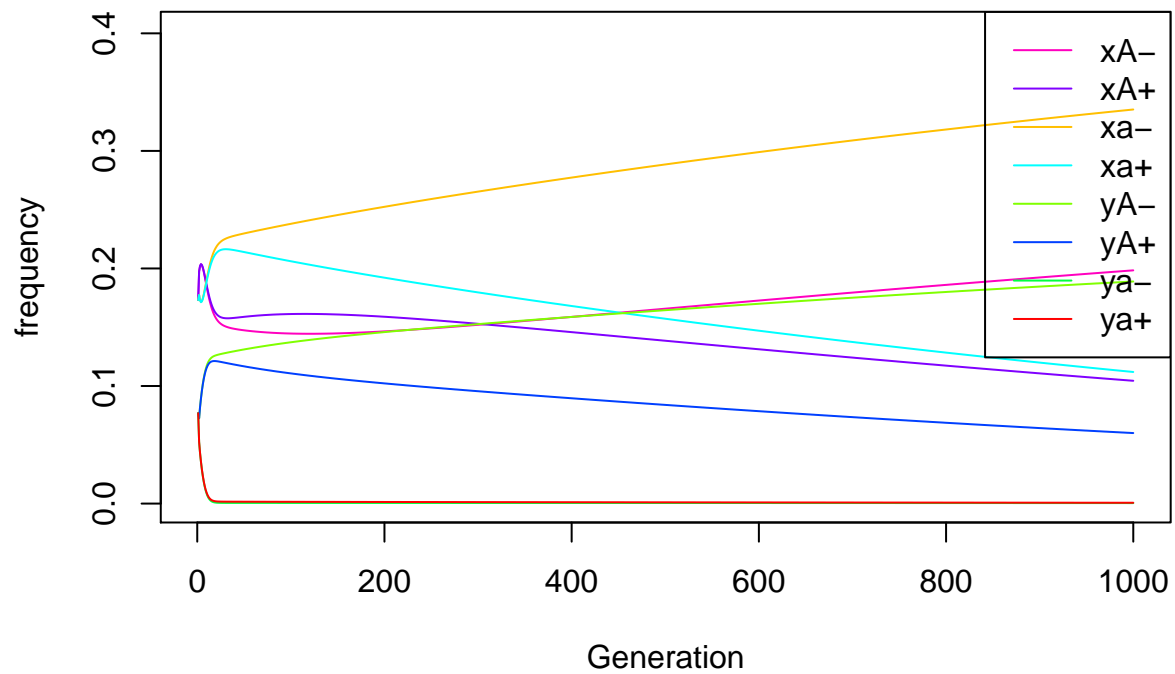
```

        allele.name = c("x","y"))
locusm = create.locus(allele1 = c(1,1,2),
                      allele2 = c(1,2,2),
                      recombination.modifier = c(0,0.5,1),
                      allele.name = c("-", "+"))
)
locussa = create.locus(allele1= c(1,1,2),
                      allele2 = c(1,2,2),
                      fitness.female = c(0.6,1,0.6),
                      fitness.male = c(0.8,1,0.3),
                      allele.name = c("A","a"))
genome = create.genome(list(locussd,locussa,locusm),
                      male.recombination = c(0.01,0.01),
                      female.recombination = c(0.01,0.01))

print(genome)
#> locus 1
#>   allele1 allele2 fitness.male fitness.female modifier sd
#> 1      x      x           1           1           1 0
#> 2      x      y           1           1           1 1
#> * * * * *
#> locus 2
#>   allele1 allele2 fitness.male fitness.female modifier
#> 1      A      A           0.8           0.6           1
#> 2      A      a           1.0           1.0           1
#> 3      a      a           0.3           0.6           1
#> * * * * *
#> locus 3
#>   allele1 allele2 fitness.male fitness.female modifier
#> 1      -      -           1           1           0.0
#> 2      -      +           1           1           0.5
#> 3      +      +           1           1           1.0
#> * * * * *

initial.frequency <- get.frequency.from.one.allele.frequency(genome,3,2,1.99)
freq <- compute.frequency.evolution(genome, generations = 1000)
plot.haplotype.frequency(genome,freq)

```



Unfortunately, we are not able to reproduce Otto's result. The reason why is still an open question.

Unfor-