Evolution of the recombination rate

Frederic Michaud

2018-04-25

Introduction

It is generally believed that, due to Sex-antagonist selection, the recombination on the sex-chromosom 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.

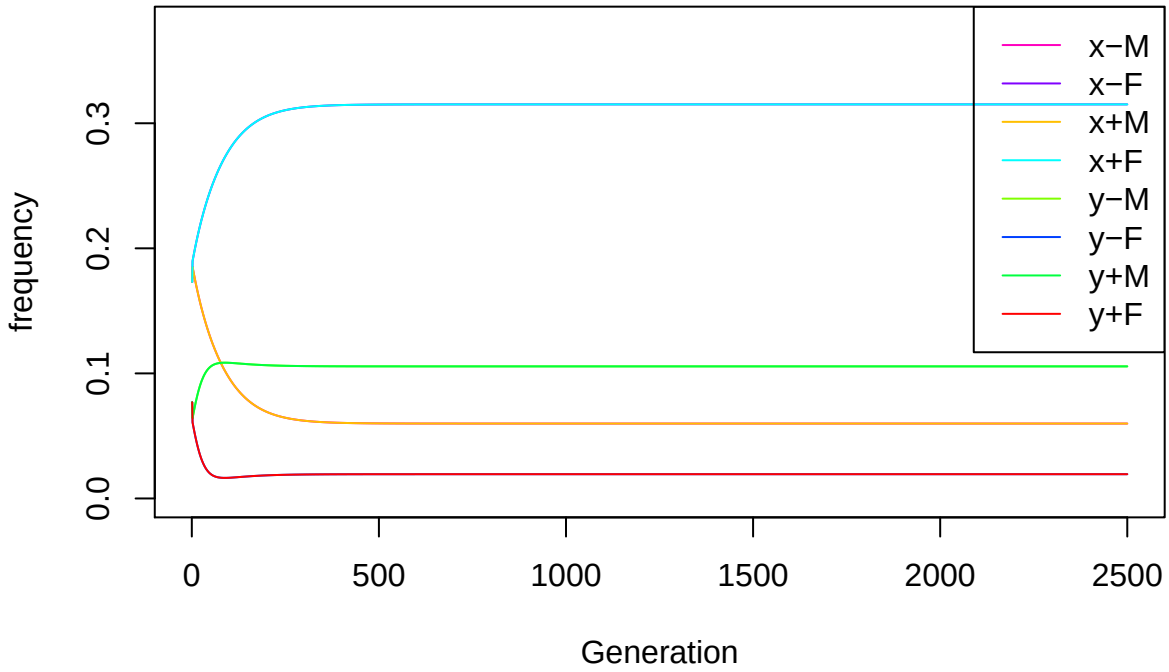
Sex-antogonistic selection

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 benefitial one.

Otto results

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.

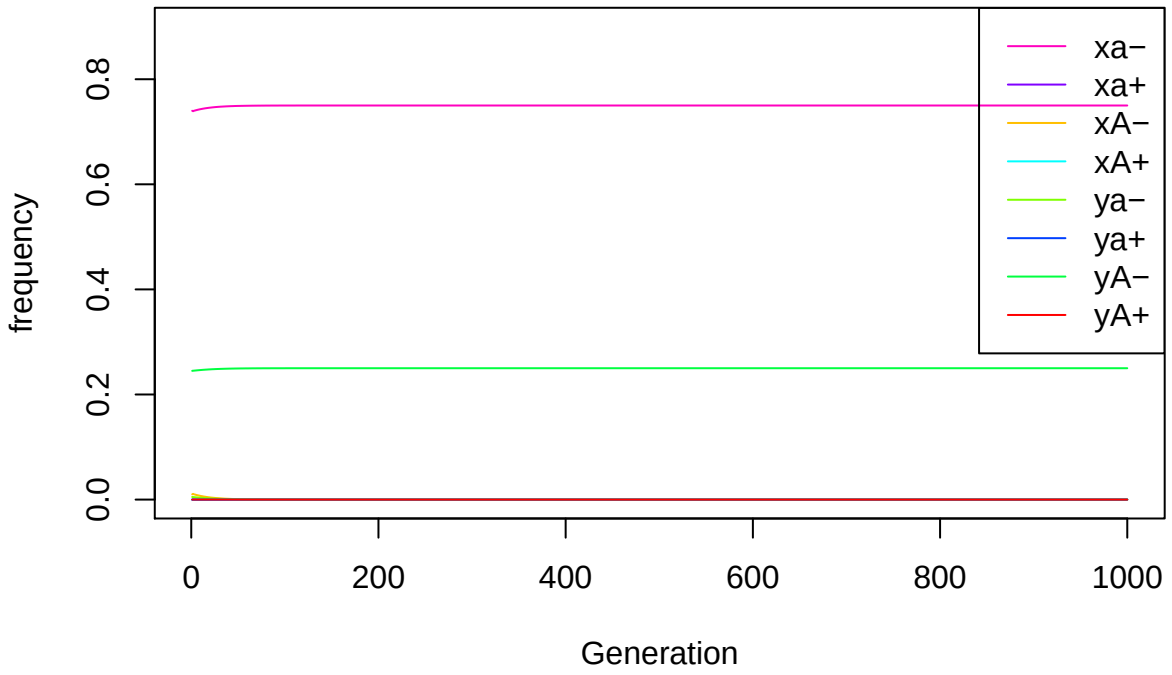
In the next section, we will proceed as follow. We start from an equilibrium without any recombination. We set the initial frequency of gamete and the fitness of male and female so that we are in orange region of Figure 1 of previously mentionned paper, in an equilibrium B'. We then run the simulation without recombination for a 1000 generations to go to the exact equilibrium value. We then add a mutation which add recombination, and check if this mutation invade the system or not.

setting the genome

```
locussa = create.locus(allele1=  c(1,1,2),
                       allele2 = c(1,2,2),
                       fitness.female = c(0.75,1,0.75),
                       fitness.male = c(0.95,1,0.4),
                       allele.name = c("a","A"))
genome = create.genome(list(locussd,locussa,locusm),
                        male.recombination = c(0.01,0.5),
                        female.recombination = c(0.01,0.5))
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         0.95           0.75        1
#> 2       a       A         1.00           1.00        1
#> 3       A       A         0.40           0.75        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
#> * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
get.haplotype.names(genome)
#> [1] "xa-" "xa+" "xA-" "xA+" "ya-" "ya+" "yA-" "yA+"
```

Computing the frequency at equilibrium

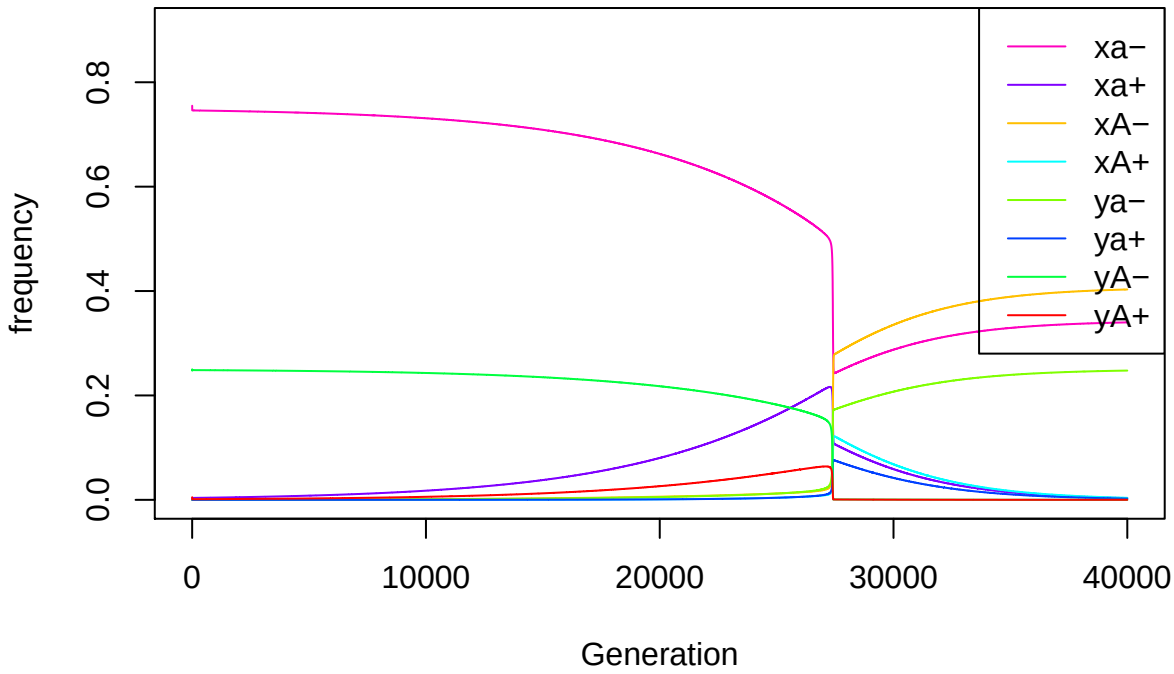
```
initial.frequency <- get.frequency.from.gamete.frequency(genome,
                                                          male.gamete.frequency = c(0.49,0,0.01,0,0.01,0,0.49,0)
                                                          female.gamete.frequency = c(0.99,0,0.01,0,0,0,0,0))
freq <- compute.frequency.evolution(genome,initial.frequency, generations = 1000)
plot.haplotype.frequency(genome,freq)
```



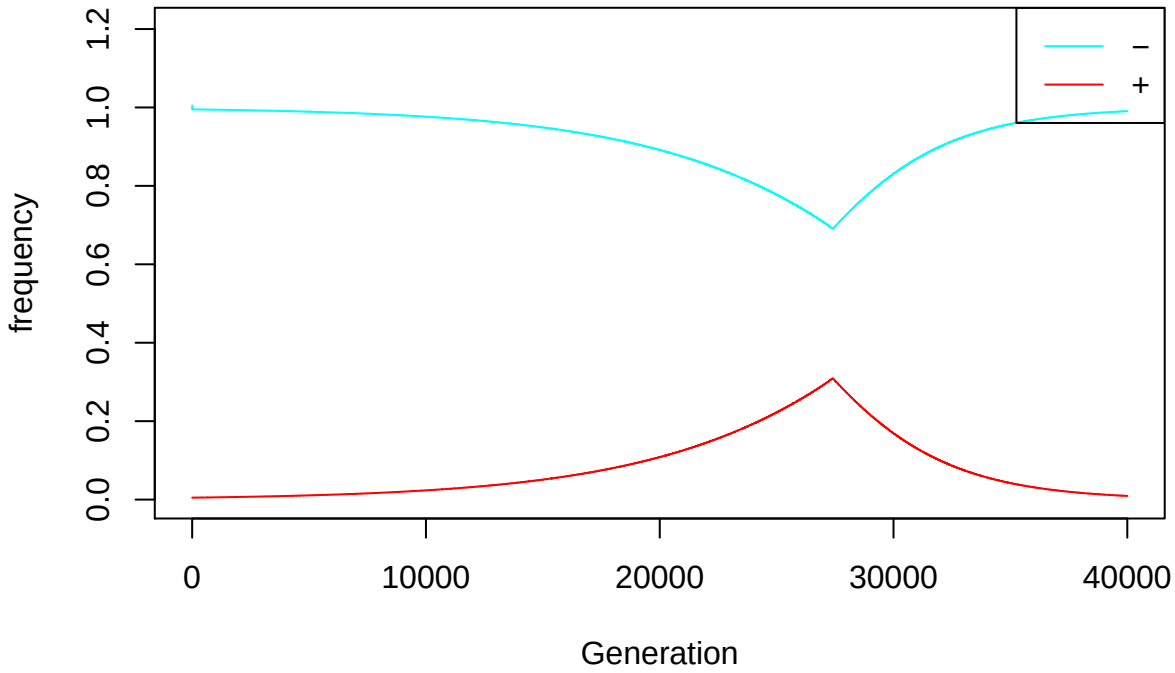

```
equilibrium.frequency <- freq[,ncol(freq)]
equilibrium.frequency[8] <- 0.01 #addinf a mutation
```

simulation starting from equilibrium with added mutation for recombination

```
freq <- compute.frequency.evolution(genome,equilibrium.frequency, generations = 40000)
plot.haplotype.frequency(genome,freq)
```




```
plot.allele.frequency(genome,freq,3)
```



As we can see, the mutation causing recombination (called +) actually invade the system. However, if we wait long enough, we switch to a different equilibrium, and the recombination disapear again.

going from A to A'

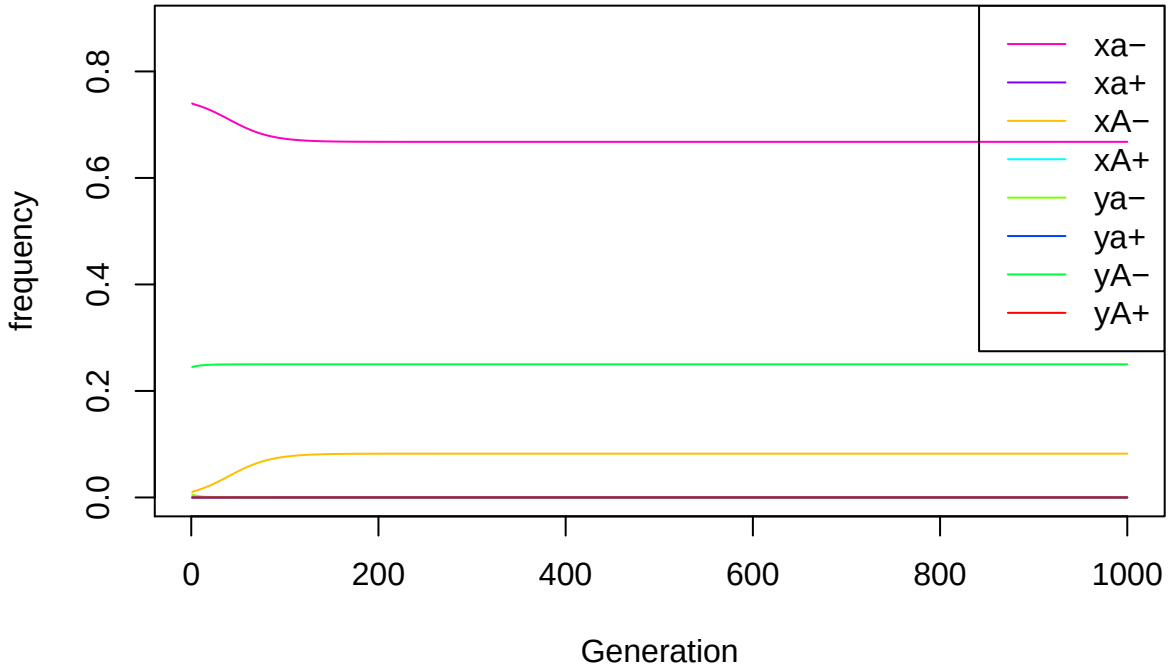
We now consider a similar problem but starting from a different equilibrium. ### setting the genome

```
locussa = create.locus(allele1=  c(1,1,2),
                       allele2 = c(1,2,2),
                       fitness.female = c(0.75,1,0.75),
                       fitness.male = c(0.9,1,0.6),
                       allele.name = c("a","A"))
genome = create.genome(list(locussd,locussa,locusm),
                        male.recombination = c(0.01,0.5),
                        female.recombination = c(0.01,0.5))
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          0.9           0.75        1
#> 2       a       A          1.0           1.00        1
#> 3       A       A          0.6           0.75        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
```

```
#> * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
get.genotype.names(genome)
#>  [1] "xa-/xa-" "xa-/xa+" "xa-/xA-" "xa-/xA+" "xa-/ya-" "xa-/ya+" "xa-/yA-"
#>  [8] "xa-/yA+" "xa+/xa+" "xa+/xA-" "xa+/xA+" "xa+/ya-" "xa+/ya+" "xa+/yA-"
#> [15] "xa+/yA+" "xA-/xA-" "xA-/xA+" "xA-/ya-" "xA-/ya+" "xA-/yA-" "xA-/yA+"
#> [22] "xA+/xA+" "xA+/ya-" "xA+/ya+" "xA+/yA-" "xA+/yA+"
```

Computing the frequency at equilibrium

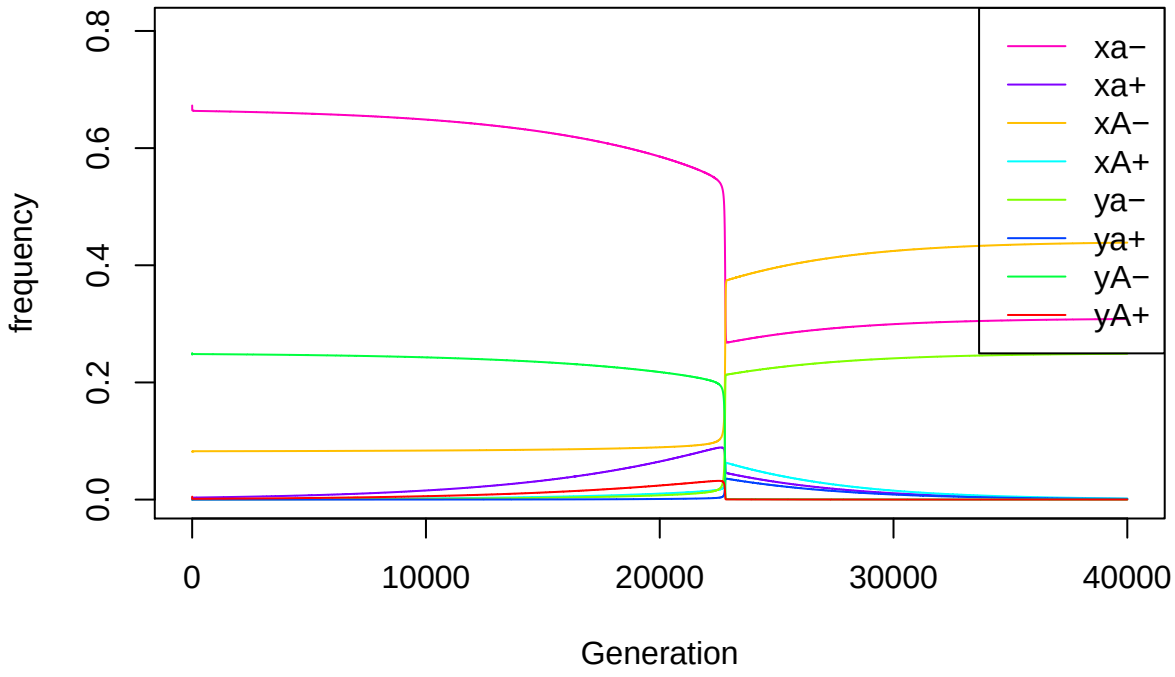
```
initial.frequency <- get.frequency.from.gamete.frequency(genome,
                                                          male.gamete.frequency = c(0.49,0,0.01,0,0.01,0,0.49,0)
                                                          female.gamete.frequency = c(0.99,0,0.01,0,0,0,0,0))
#initial.frequency <- get.frequency.from.gamete.frequency(genome,
#                                                          male.gamete.frequency = c(0.49,0,0.01,0,0.49,0,0.01,0)
#                                                         female.gamete.frequency = c(0.99,0,0.01,0,0,0,0,0))
freq <- compute.frequency.evolution(genome,initial.frequency, generations = 1000)
plot.haplotype.frequency(genome,freq)
```



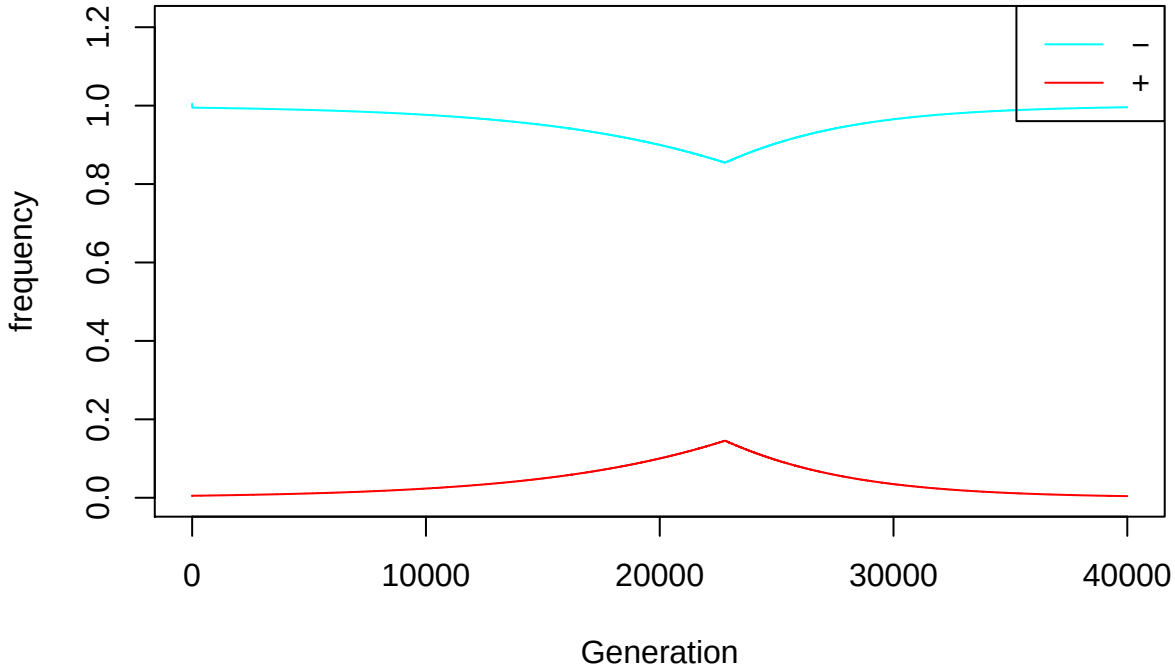

```
equilibrium.frequency <- freq[,ncol(freq)]
equilibrium.frequency[8] <- 0.01 #adding a mutation
```

simulation starting from equilibrium with added mutation for recombination

```
freq <- compute.frequency.evolution(genome,equilibrium.frequency, generations = 40000)
plot.haplotype.frequency(genome,freq)
```




```
plot.allele.frequency(genome,freq,3)
```



As we can see, the mutation causing recombination also invade the system, but also disapear after sometime.

Stable recombination region

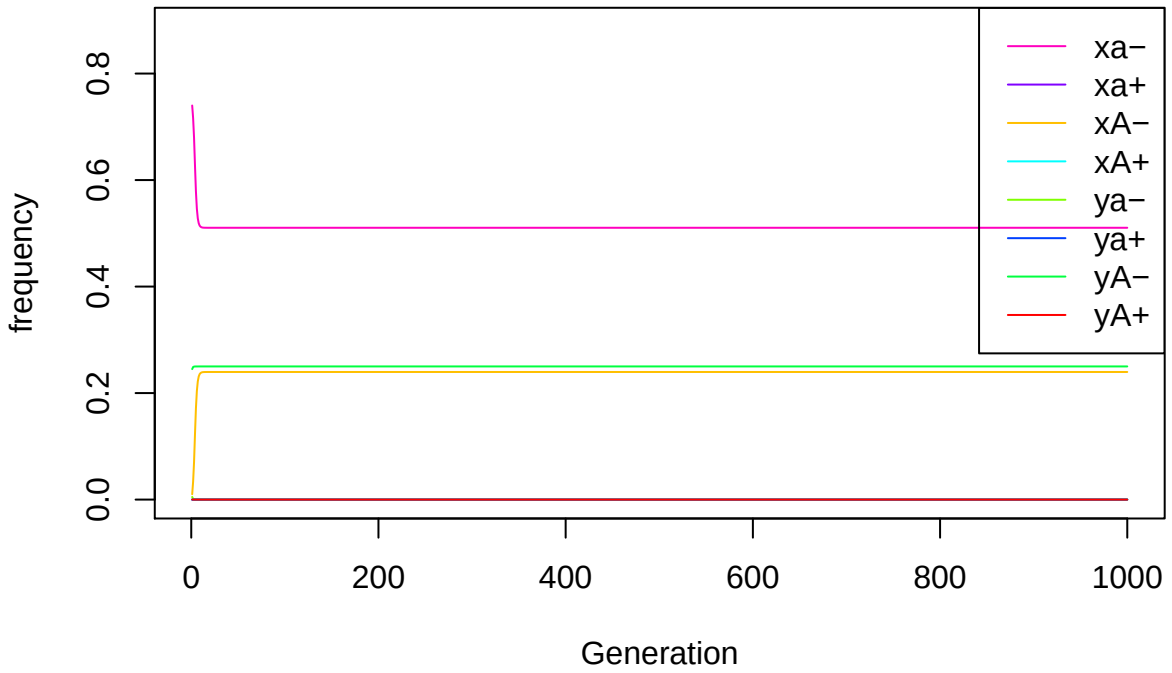
We now consider a similar problem but starting from a different equilibrium. ### setting the genome

```
locussa = create.locus(allele1=  c(1,1,2),
                       allele2 = c(1,2,2),
                       fitness.female = c(0.2,1,0.2),
```

```
                      fitness.male = c(0.2,1,0.2),
                      allele.name = c("a","A"))
genome = create.genome(list(locussd,locussa,locusm),
                       male.recombination = c(0.01,0.5),
                       female.recombination = c(0.01,0.5))
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          0.2            0.2        1
#> 2       a       A          1.0            1.0        1
#> 3       A       A          0.2            0.2        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
#> * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
get.genotype.names(genome)
#>  [1] "xa-/xa-" "xa-/xa+" "xa-/xA-" "xa-/xA+" "xa-/ya-" "xa-/ya+" "xa-/yA-"
#>  [8] "xa-/yA+" "xa+/xa+" "xa+/xA-" "xa+/xA+" "xa+/ya-" "xa+/ya+" "xa+/yA-"
#> [15] "xa+/yA+" "xA-/xA-" "xA-/xA+" "xA-/ya-" "xA-/ya+" "xA-/yA-" "xA-/yA+"
#> [22] "xA+/xA+" "xA+/ya-" "xA+/ya+" "xA+/yA-" "xA+/yA+"
```

Computing the frequency at equilibrium

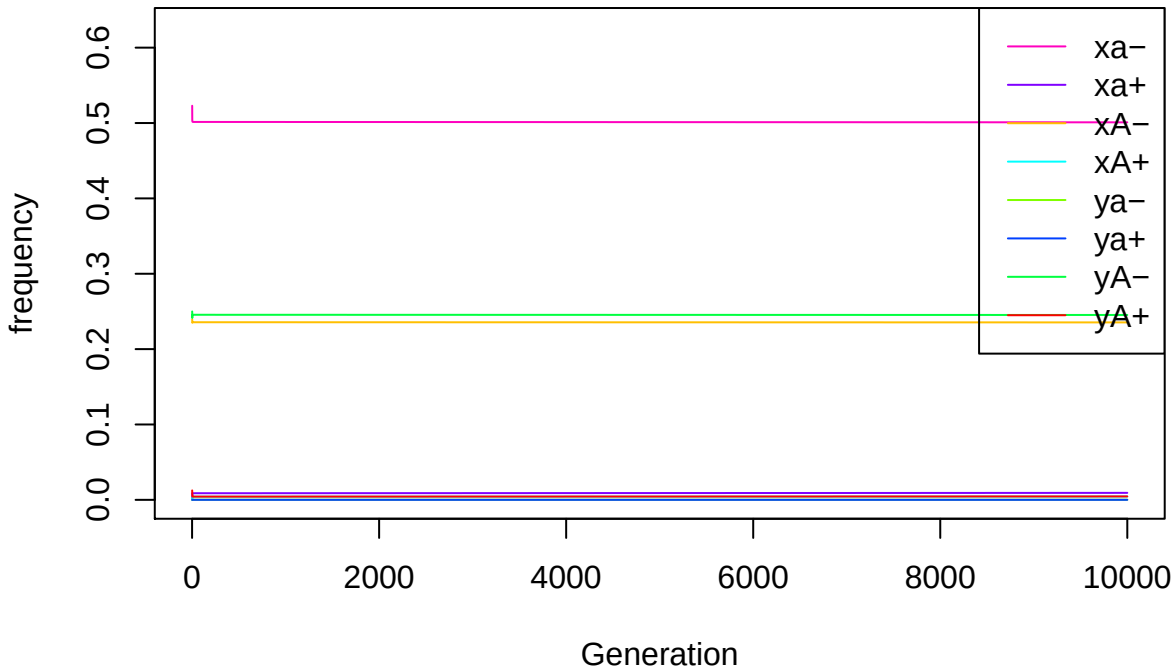
```
initial.frequency <- get.frequency.from.gamete.frequency(genome,
                                                          male.gamete.frequency = c(0.49,0,0.01,0,0.01,0,0.49,0)
                                                          female.gamete.frequency = c(0.99,0,0.01,0,0,0,0,0))
#initial.frequency <- get.frequency.from.gamete.frequency(genome,
#                                                          male.gamete.frequency = c(0.49,0,0.01,0,0.49,0,0.01,0)
#                                                         female.gamete.frequency = c(0.99,0,0.01,0,0,0,0,0))
freq <- compute.frequency.evolution(genome,initial.frequency, generations = 1000)
plot.haplotype.frequency(genome,freq)
```



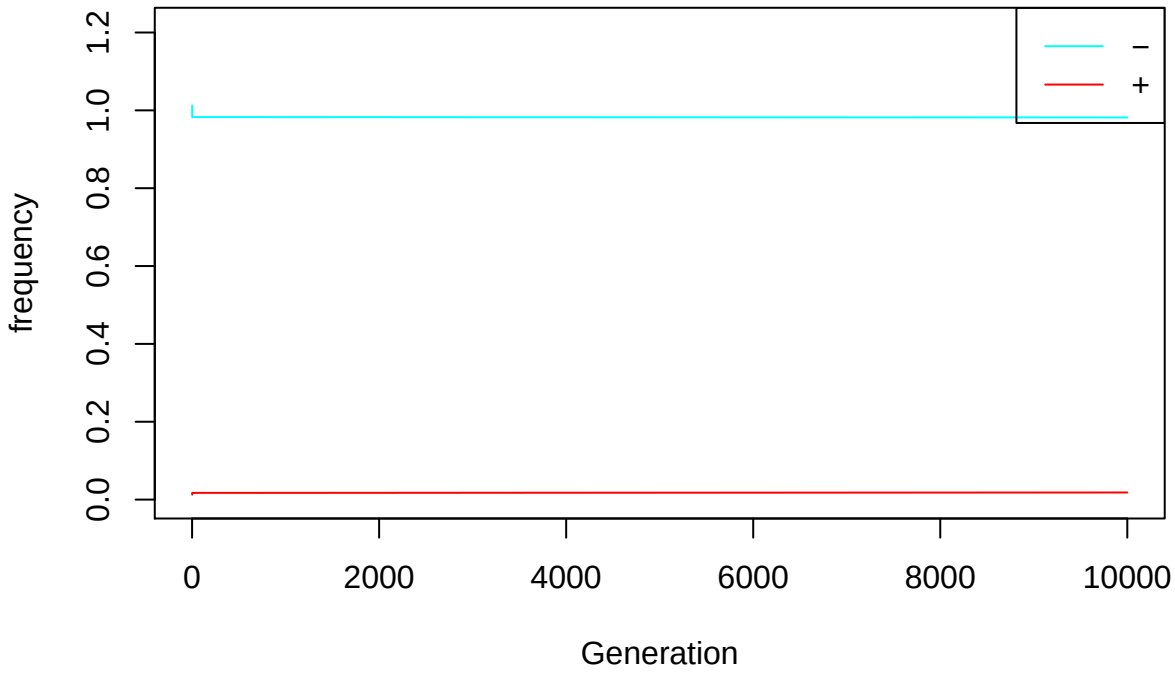

```
equilibrium.frequency <- freq[,ncol(freq)]
equilibrium.frequency[8] <- 0.025 #adding a mutation
```

simulation starting from equilibrium with added mutation for recombination

```
freq <- compute.frequency.evolution(genome,equilibrium.frequency, generations = 10000)
plot.haplotype.frequency(genome,freq)
```




```
plot.allele.frequency(genome,freq,3)
```



In this case, the recombination is selected, but very very slowly. It should not disapear even if we wait long enough.