PolySim User Manual

A large scale efficient forward-in-time simulator for polyploid genome evolution

Abdulqader Jighly

AgriBio - Center for AgriBioscience

Department of Economic Development, Jobs, Transport and Resources

Victoria, Australia

abdulqader.jighly@ecodev.vic.gov.au

Table of Contents

Introduction

Polyploids are a result of whole genome duplication and have been classified into autopolyploids, allopolyploids, and segmental allopolyploids (Stebbins 1947). Autopolyploids are a consequence of a genome duplication(s) while allopolyploids evolve from hybridization of two or more different diploid genomes. Figure 1 shows an example of the evolution of an allohexaploid organism. In segmental allopolyploids, some chromosomes can act as allopolyploids, while others act as autopolyploids. Thus, the evolutionary history of polyploid organisms differ from diploids as they can have multiple ancestor species.

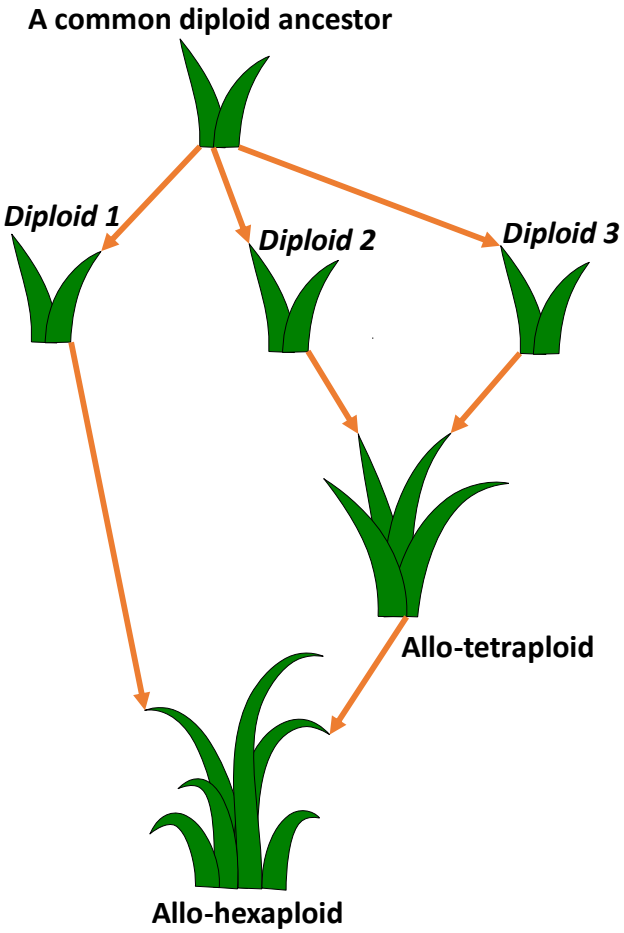


Figure 1. An example of the evolution of an allohexaploid organism

Because autopolyploids/segmental allopolyploids have multiple copies of the same chromosome, multivalent pairings can be formed during meiosis. When multivalent formations occurs, two sister chromatids can segregate into the same gamete in a process called the double reduction (Figure 2) (Mather 1935). However, preferential pairing between genomically related genomes

can reduce double reduction (Sybenga 1996). Preferential pairing can be described using the multivalent to bivalents ratio (τ) which is equal to zero for 100% preferential pairing, such as in allopolyploids (Voorrips and Maliepaard 2012). Interestingly, a mathematical model for tetraploid meiosis suggested that the double reduction coefficient is explicitly dependent on τ, which makes this value important to simulate both double reduction as well as preferential pairing (Rehmsmeier 2013).

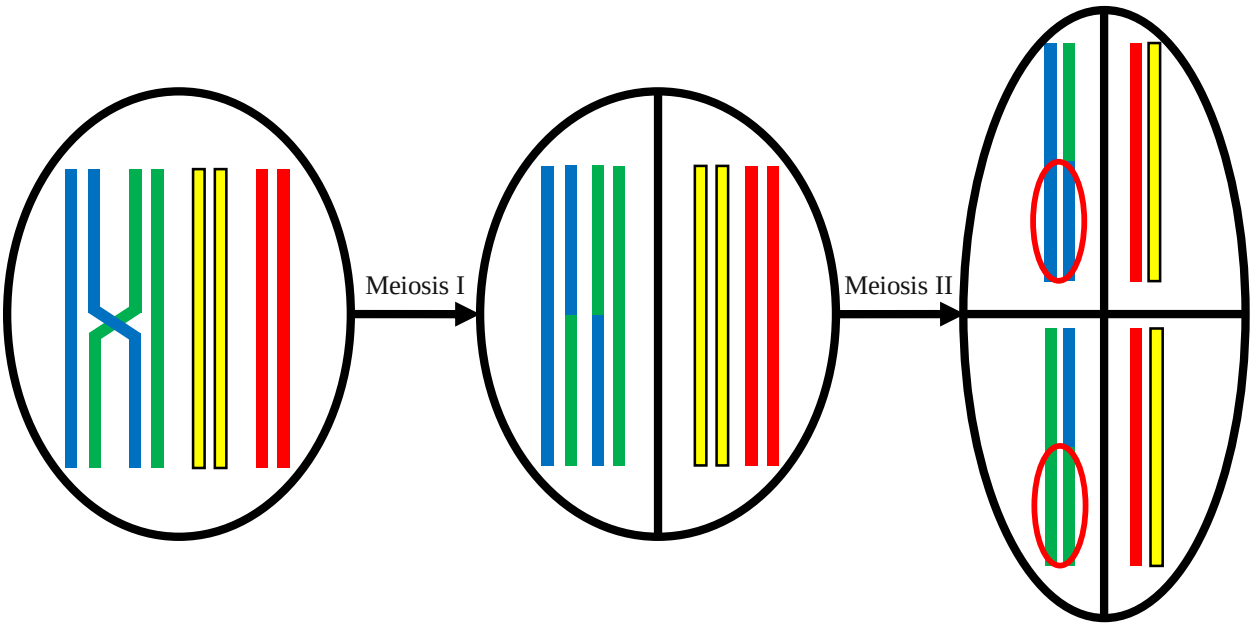


Figure 2. The double reduction where two sister chromatids may end at the same gamete (the two red eclipses).

The special features of multiple origins, preferential pairing and double reduction add extra challenges for simulating polyploid genomes and make diploid simulators inadequate to accurately simulate polyploid genomes. There are currently over 40 simulators available with variable features using both forwards and backwards approaches (reviewed in Habon et al. 2012). However, none of them can simulate polyploid genomes (Dufresne et al. 2014; Habon 2014). The only simulator that has no restriction on the polyploid level is SimuPOP (Peng et al. 2008), but SimuPOP does not consider the special evolutionary or meiotic features of polyploids, which creates a need for a new simulator. Moreover, the developers stated that SimuPOP is largely untested for polyploids.

PolySim Algorithm

PolySim uses the following algorithm to simulate polyploid genomes:

```
Read parameter file
For each generation {
    If (Generation == speciation generation) {
        Evolve new organism
    }
    For each organism {
        Add random mutations
        For each mating {
            Select parents
            For each individual (offspring per mate) {
                For each chromosome {
                    Count number of meiotic formations
                    For each meiotic formation {
                        Count and locate crossovers
                        Do crossovers
                    }
                }
            }
        }
    }
    Speciate with speciation rates
}
Print
```

Parameters

PolySim reads all parameters except the parameter "*Repeat*" from a single file named '*Parameters.txt*' This file should contain a list from the below parameters, depending on the final population requirements. The parameter values should be described within square brackets [ ] after the parameter name.

1- Repeat

Repeat is the only parameter that should be set when running the PolySim command. The parameter ensure using different random seeds when sending multiple jobs with the same parameter set to a server at the same time. Thus, the user can avoid getting the same random seed in more than one replication of the simulation.

2- outFile

Default 'Out.txt'

A string to describe the suffix of the output file name.

3- PrintProgress

Default 'No'

Setting this parameter to 'Yes' will allow printing the number of the current generation on the screen frequently every "*PrintEvery*" value.

4- PrintEvery

Default = '1000'

If the parameter "*PrintProgress*" is set to 'Yes', the user will get a notification every "*PrintEvery*" generation on the console in order to track the progress of the simulation.

5- MateSystem

Default = 1

Range between 0 (fully selfed organism) and 1 (fully crossed organism). A value between 0 and 1 represent partially crossed organisms or the percentage of crossed matings.

6- NoGenotypes

The population sizes for the diploid and polyploid evolved organisms. E.g. for a tetraploid organism the common diploid ancestor should speciate into two different diploids (A and B) then A and B will form the tetraploid AB. The total evolved organisms is three. Thus, the parameter "*NoGenotypes*" can be set as: [A-PopSize B-PopSize AB-Popsize] e.g. [1000 1000 1000]. Each number should be separated with space or tab characters and the order of the numbers ***MUST*** be the same as the parameter "*DirectHybrid*"

7- OffPerMate

The number of offspring per mate. The same randomly selected parents can develop "*OffPerMate*" genotypes in the population (i.e. have 1 to many offspring per mating). Thus the number of mating to generate the final population will equal to "*NoGenotypes*" divided by "*OffPerMate*", as the population will have a constant size regardless the value of this parameter.

8- NoSnps

The number of simulated SNPs per chromosome.

9- Centromere

To define the position of the centromere (ranged between 0 and "*NoSnps*"). This is important mainly for autopolyploids where the centromere position is important to structure and position the crossovers for multivalent formations.

10- CommonAncestorNoGenotypes

The population size for the common diploid ancestor. All evolved diploid organisms will be speciated from this common diploid ancestor.

11- CommonAncestorNoChr

The number of chromosomes for the common diploid ancestor.

12- NoChr

The number of chromosomes for the diploid and polyploid evolved organisms. E.g. for a tetraploid organism the common diploid ancestor should speciate into two different diploids (A and B) then A and B will form the tetraploid AB. The total evolved organisms is three. Thus, the parameter "*NoChr*" can be set as: [A-NoChr B-NoChr AB-NoChr] e.g. [7 7 14]. Each number

should be separated with space or tab characters and the order of the numbers ***MUST*** be the same as parameter "*DirectHybrid*", see below

13- PloidyType

This parameter can be used to differentiate among autopolyploids, allopolyploids and segmental autopolyploids by setting the number of chromosomes that can form multivalent formation for each evolved polyploid organism. Thus, it should range from 0 (fully allopolyploid) to "*CommonAncestorNoChr*" (fully autopolyploid). E.g. when simulating seven chromosomes of a hexaploid organism the common diploid ancestor should speciate into three different diploids (A, B and C), A and B will form the tetraploid AB, and AB will combine with C genome to form the hexaploid ABC. Here, we have two polyploid organisms. Thus, this parameter can be set as [0 0] for allopolyploids, e.g. [1 2] for segmental-allopolyploids and [7 7] for autopolyploids in the case of organism with 7 haploid chromosomes. The segmental polyploid case means that the first chromosome in the tetraploid organism AB and the first two chromosomes of the hexaploid ABC will be treated as autopolyploids (depending on the "*Multivalent*" parameter), while the rest of the chromosomes will be treated as allopolyploid. It is important to note that first chromosomes in order will be the auto-chromosomes in segmental allopolyploids. Each number should be separated with space or tab characters and the order of the numbers ***MUST*** be the same to that of the polyploid organisms in the parameter "*DirectHybrid*". When printing the results for segmental autopolyploids, the first chromosomes in the output file will be the autopolyploid-like chromosomes.

14- MutationRate

The random mutation rate sampled from Poisson distribution. The distribution mean is equal to: "*MutationRate* × *NoSnp* × *NoChr*"

15- StopMutation

To stop the random mutation for the final "*StopMutation*" generations in order to prevent further mutation. For example, if "*StopMutation*" is equal to 10, then the last ten generations will run without mutations.

16- PolySpeciation

To simulate the rate of gene flow from the lower to higher ploidy after the first speciation event.

17- DiSpeciation

To simulate the rate of gene flow from the common diploid ancestor to the other evolved diploids after their first occurrence.

18- NoGenerations

The total number of generations for the full simulation.

19- SaveGeneration

To save the SNP matrix every "*SaveGeneration*"

20- NoGeneration2PopSize

Default=10. When new organism evolve, it is assumed to start with a single genotype and eventually reaches the targeted population size. PolySim applies the equation below to exponentially increase the number of genotypes during "*NoGeneration2PopSize*" (*G*).

$$\underset{i=1\rightarrow G}{P_i} = \left[e^{\left(\frac{\ln(S)}{G}\right)} \right]^i$$

S: Targeted population size

G: Number of generations to reach the targeted population size

i: Generation number *i* from 1 → *G*

P_i: Number of individuals in the generation *i*

Figure 3 shows the increase in a population size during 100 generations to reach 10,000 individuals.

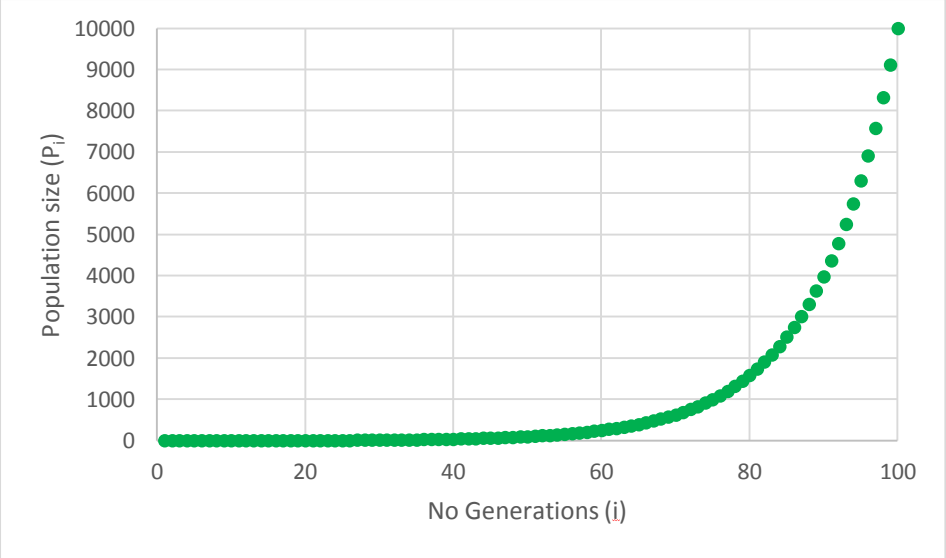


Figure 3. The exponential increase in population size

21- SpeciationGenerations

The number of generations until the occurrence of the first speciation for each organism. E.g. for a tetraploid organism the common diploid ancestor should speciate into two different diploids (A and B) then A and B will form the tetraploid AB. The total evolved organisms is three. Thus, the parameter "*SpeciationGenerations*" can be set as: e.g. [1000 5000 8000]. Each number should be separated with space or tab characters and the order of the numbers ***MUST*** be the same to that of the parameter "*DirectHybrid*"

22- Multivalent

The percentage of the multivalent formation for each combination of genomes (two or more). This parameter can simulate both multivalent formations and homoeologues chromosome pairing. For autopolyploids, any ploidy/2 alleles can be found in the same gamete while passing chromosomes of the same genome in allopolyploids to the same gametes can reduce the fertility as they can lead to aneuploidy (Ramsey and Schemske 2002; Stebbins 1971; Kovarik et al. 2012). For this reason, PolySim allows multivalent formation among homoeologues chromosomes in allopolyploids but only one copy from each genome will pass to each gamete.

For example, when simulating a hexaploid organism, the common diploid ancestor should speciate into three different diploids (A, B and C) then A and B will form the tetraploid AB and this AB with the C genome will form the hexaploid ABC. Thus, this parameter can be set as [AB 0.004; AC 0.005; BC 0.006]. Each genome combination and multivalent rate should be separated with space or tab characters and separating different genome combinations should be done using semicolon. For autopolyploids the user should define the genome combination by doubling the genome character according to the ploidy type for example [AA 0.2] for autotetraploids and [AAA 0.1] for autohexaploids.

23- DirectHybrid

This parameter describes the evolution of organisms starting from the common diploid organism. For example, when simulating a hexaploid organism, the common diploid ancestor should speciate into three different diploids (A, B and C) then A and B will form the tetraploid AB and this AB with the C genome will form the hexaploid ABC. The parameter "*DirectHybrid*" can interpret the described process as [A B C A_B AB_C]. The underscore represent the cross between two genomes to form the higher polyploidy. Each genome on both sides of this under dash must be present in the previous genomes. The order of the evolved genomes describes the sequence of their evolution through generations and is very critical as many other parameters will depend on it.

Let us get the following auto-octaploid example where only a single genome formed the final organism: [A A_A AA_AA]. In such case, it will not be necessary to simulate a common

diploid organism. Thus, the user can set the parameter "*CommonAncestorNoGenotype*" to '1' and the first value of the parameter "*SpeciationGenerations*" to '1' as well.

24- Poisson

Default 'No'

To simulate the recombination rate from Poisson distribution. If the user set this parameter to 'No' then the parameter "*RecombinRate*" should be used. Otherwise, the parameter "*MeanCO*" should be set (detailed below).

25- RecombinRate

If "*Poisson*" = 'No', then allow the user to enter a specific rate for each number of crossover(s) per chromosome. For example, setting the parameter as [0.05 0.8 0.14 0.01] means that the probability of no crossover per chromosome is equal to 5%, one crossover is 80%, two crossovers is 14% and three crossovers is 1%. The parameter is open and the user can enter probabilities up to "*NoSnps*" – 1. It is important to note that the sum of all probabilities in this parameter should equal to one.

26- MeanCO

If "*Poisson*" = 'Yes', then simulate the recombination rate from Poisson distribution that has a mean equal to the parameter "*MeanCO*"

27- RecombinPRate

A recent report (Pecinka et al. 2011) showed that recombination rate can increase in polyploids over their ancestral diploids. PolySim takes this feature into account using this parameter. If "*RecombinPRate*" > 1 then polyploid recombination will increase based on this rate while selecting a value < 1 will result in decreasing the recombination rate in polyploids.

If "*Poisson*" in parameter 25 is set to 'Yes', the Poisson distribution mean for polyploid organisms will equal to: "*MeanCO* × *RecombinPRate*" (Figure 4).

If "*Poisson*" were set to 'No', let us consider the "*RecombinPRate*" = 1.2. The user is required to specify a rate for each number of crossovers based on the "*RecombinRate*" parameter, for example [0.05 0.8 0.14 0.01] probabilities for 0, 1, 2, 3+ crossovers, respectively. Those recombination values are based on the user input and will not follow any distribution in most cases. Thus, in order to fit them with Poisson distribution, we have to set the distribution mean (step 1) and a range of Poisson simulated values for each number of possible crossovers with similar probabilities (step 2). Then we can adjust this distribution according to the parameter "*RecombinPRate*" (step 3 and 4). To do that we did the following four steps as in figure 4:

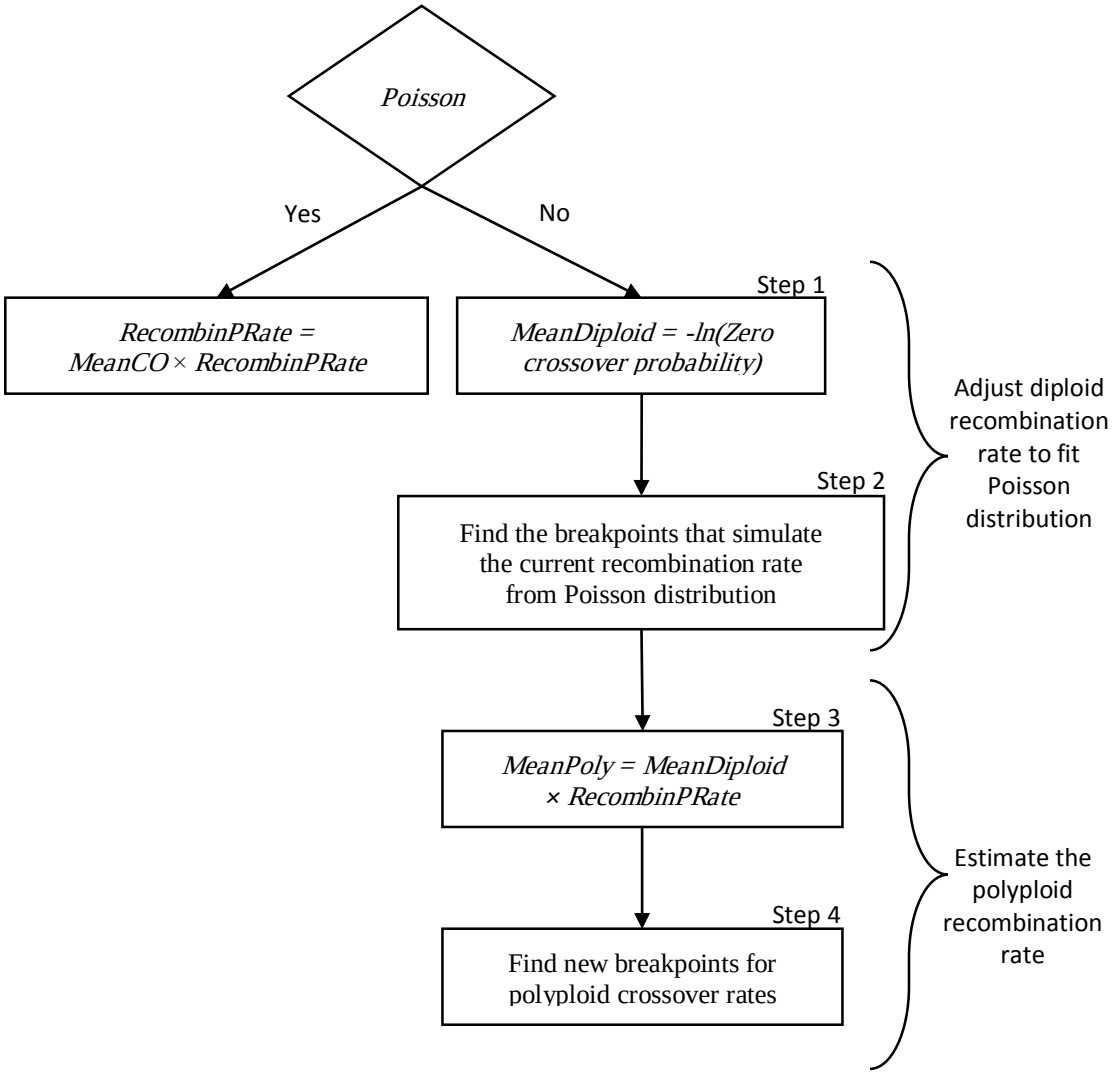


Figure 4. Diagram explaining how PolySim estimate recombination rate in polyploids

1- Consider a Poisson distribution were the probability of getting '0' crossovers is equal to our '0' crossover probability (Figure 5a). Then the Poission distribution mean will equal to:

$$\text{MeanDiploid} = -\ln(\text{Zero crossover probability}) = -\ln(0.05) \approx 3$$

2- Find the sum of probabilities (a stepwise probability) that can describe the user input of the "*RecombinRate*" parameter values. In our case {the sum of probabilities for getting values ranging from 1 to 5 in a Poisson distribution with mean equal to 3 will equal to ~80% (1 crossover), 6 - 8 will equal to ~14% (2 crossovers), and > 8 will equal to ~1% (3 crossovers)} (Figure 5a). The breakpoints of 1, 5 and 8 will be used to estimate the recombination rate in polyploids.

3- Consider a new Poisson distribution with mean = "*MeanDiploid* × *RecombinPRate*" (Figure 5b)

$$\text{MeanPoly} = 3 \times 1.2 = 3.6$$

4- Estimate the new sum of probabilities with the previous breakpoints (at 1-5; 6-8; and > 8 in our example) (Figure 5b).

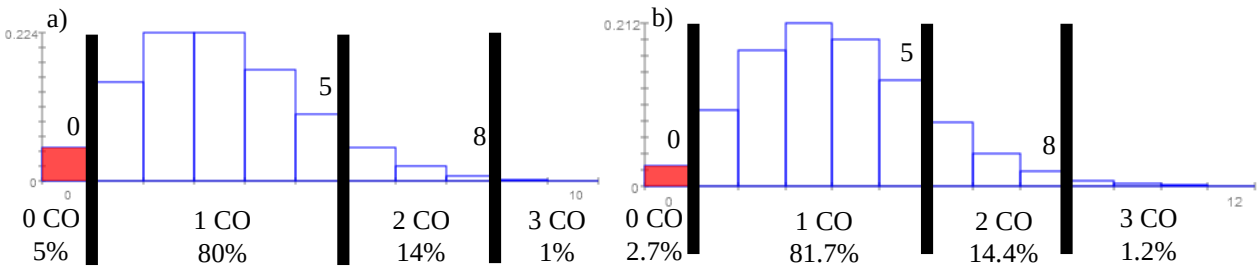


Figure 5. Simulating recombination rate increase in polyploids with the parameter "*RecombinRate*". a) the most probable Poisson distribution for the user specified *RecombinRate* [0.05 0.8 0.14 0.01] b) the adjusted distribution with *RecombinPRate* = 1.2

Notice:

If "*RecombinPRate*" will have a value other than one and the parameter "*Poisson*" will be set to "No", it is important to note that the probability of having no crossovers (the first element of "*RecombinRate*") ***MUST*** be higher than 0.

28- NoSpeciations

Default = 0

To set a number of extra speciation events during the period of "*NoGeneration2PopSize*". When speciation occur, a severe bottleneck happens as the new organism population will start with only one ancestral genotype. Thus, if we have more than one speciation event, the bottleneck effect may be less. This parameter is helpful to simulate the effect of multiple speciation events during the newly evolved population growth.

29- OverLap

Default 'No'

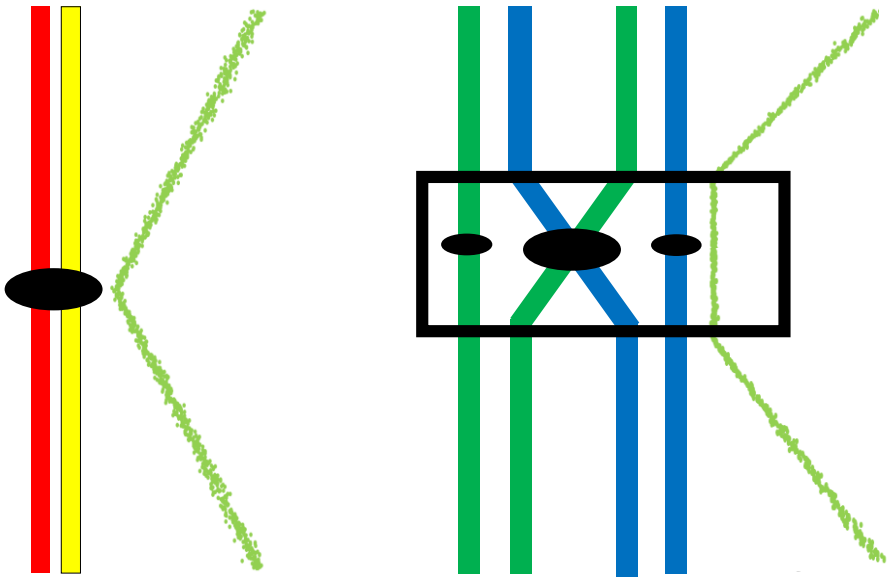
If 'Yes', then random speciation with "*DiSpeciation*" and "*PolySpeciation*" rates will be allowed during the newly evolved organism population growth. In other words, speciation during the "*NoGeneration2PopSize*" will not be allowed when "*OverLap*" is equal to 'No'.

30- PWLRecombination

Default 'No'

If 'Yes', then the position of the crossover will be simulated from 'piecewise linear distribution' according to the user inputs in the following parameters for both bivalent and multivalent formations (parameters 31 to 34; three examples are shown in figure 6).

The importance of this parameter is that recombination rate may not be equal along chromosomes. In diploids (or in bivalents in general), recombination rate tends to be minimal near the centromere, while it increases toward the telomeres as in figure 6a (Bailey et al. 2004). In some cases, specific regions on the chromosome are considered as hotspots where recombination rate goes higher than hotspot flanking regions (Myers et al. 2005). Moreover, in multivalent formations, there can be large genomic regions with a very low recombination rate (Bourke et al. 2015), as those regions will not pair with any homologous copy in a way allowing for recombination (the black box in figure 6b).



a) Recombination rate in bivalent

b) Recombination rate in multivalent

Figure 6. Recombination rate variation across chromosome in case of bivalent and multivalent.

31- BiSnpIntervals

The position of the hotspot/coldspot recombination regions for bivalent formations. The parameter should include a vector of SNP positions (sorted from smallest to largest) starting with '0' and ending with "*NoSnps*". Those position should describe the breakpoint of the hotspot

regions. For each SNP position and the following one, the number of recombination will change linearly depending on the correspondence values in the parameter "*BiSnpWeights*"

32- MvSnpIntervals

The position of the hotspot/coldspot regions for recombination for multivalent formations. Similar to "*BiSnpIntervals*"

33- BiSnpWeights

The rank (weights) of the positions described in the "*BiSnpIntervals*" parameter. The number of values entered in this parameter should equal to the number of positions in the parameter "*BiSnpIntervals*" including the '0' and the "*NoSnps*" positions. The weights can have any positive values and their sum can equal any number (not necessarily 1 as the below examples in figure 6; the methodology is fully described in the C++ reference library <random> http://www.cplusplus.com/reference/random/piecewise_linear_distribution/). The differences between the adjacent values describe the trend for the number of recombination between the correspondence positions in the parameter "*BiSnpIntervals*"

34- MvSnpWeights

The rank (weights) of the positions described in the "*MultiSnpIntervals*" parameter. Similar to "Multi*SnpIntervals*". Figure 6 describes three examples for different values for the previous parameters.

35- Seed

This parameter can be used to set a specific random seed for the simulation.

Simulating the positions of crossover events

If the parameter "*PWLRecombination*" was set to 'Yes', the recombination positions will be simulated from 'piecewise linear distribution'; otherwise all markers will have equal chance for recombination. The supplementary file "*PWLD*" can help users to test their piecewise linear distribution previous parameters through generating 10^6 random number based on their parameters. The file "*PWLD*" can read the parameters "*BiSnpIntervals*", "*BiSnpWeights*", "*MvSnpIntervals*" and "*MvSnpWeights*" from the parameter file. Below are some examples for random number generated using different parameter values for the piecewise linear distribution. In each simulation we generated 10^6 crossover positions for a chromosome with 1000 markers

(from 0 to 999). The first case (Figure 7a) is similar to a chromosome with low recombination near the centromere (marker ID 500 with weight 0.1) and high recombination near the telomeres (marker IDs 0 and 999). This case can be similar to the recombination hotspot regions for the bivalent formation in many organisms. The number of recombination events increased linearly toward the telomeres.

The second case (Figure 7b) can describe standard hotspot regions for multivalent formation in many organisms where we have a large block with very low recombination rate surrounding the centromere (between the marker IDs 300 and 700) and high recombination rate near the telomere (marker IDs 0 and 999). The number of recombination events increased linearly toward the telomeres while the number of recombination events was consistence and low in a big region around the centromere in concordance to the descriptive parameters below. The last case (Figure 7c) is similar to the second case with a hypothetical hotspot region in the middle of the coldspot region near the centromere (marker ID 500).

To run "*PWLD*" file, just put this file in the same folder where the "*Parameters.txt*" file exists and type:

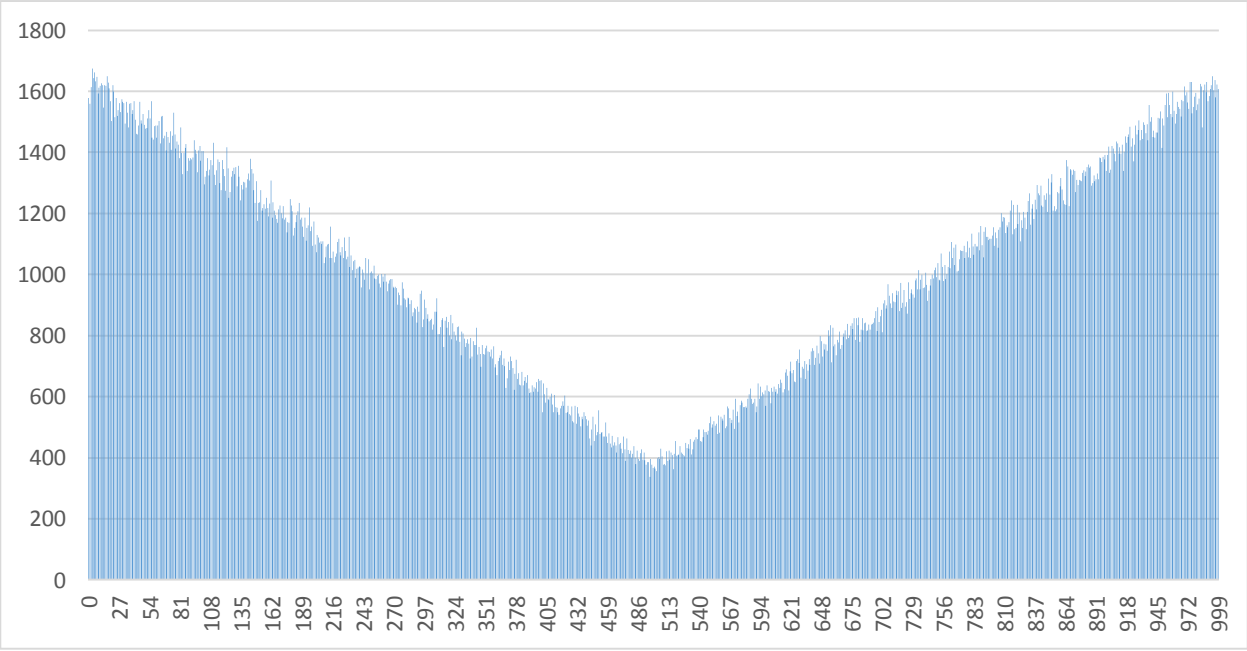
```
./PWLD
```

This will automatically generate 1×10^6 random numbers based on your "*BiSnpIntervals*", "*BiSnpWeights*", "*MvSnpIntervals*" and "*MvSnpWeights*" parameters.

a) Case 1:

Intervals: [0 500 999]

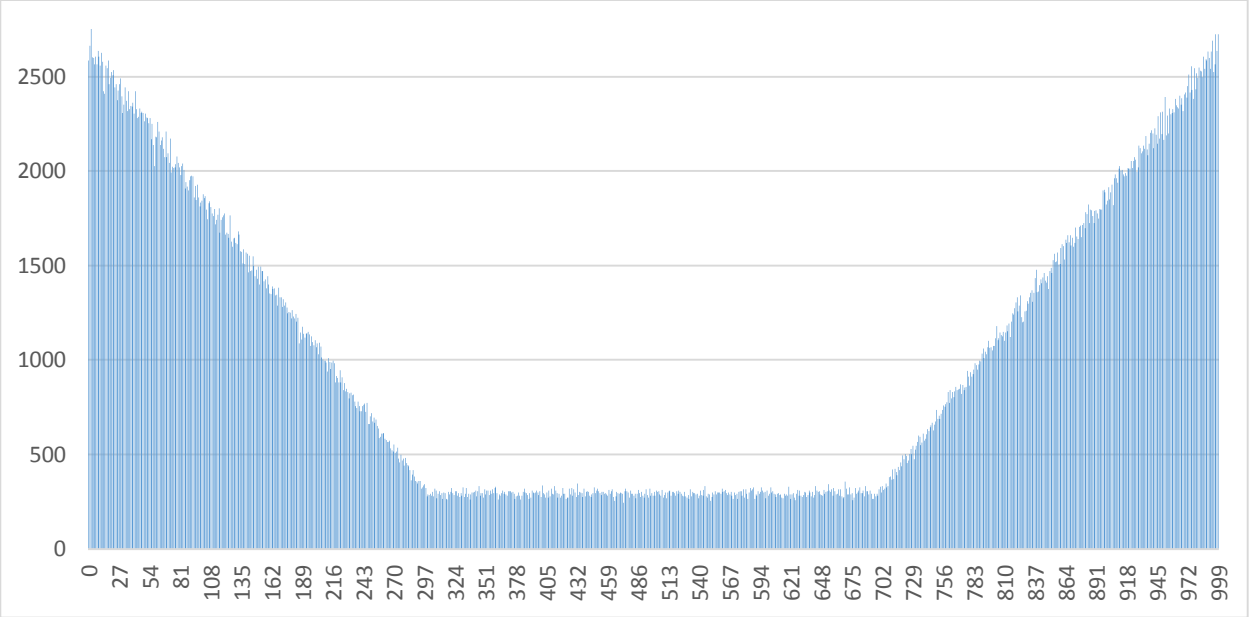
Weight: [0.45 0.1 0.45]



b) Case 2:

Intervals: [0 300 700 999]

Weight: [0.45 0.05 0.05 0.45]



c) Case 3:

Intervals: [0 300 499 500 501 700 999]

Weight: [0.32 0.01 0.01 0.32 0.01 0.01 0.32]

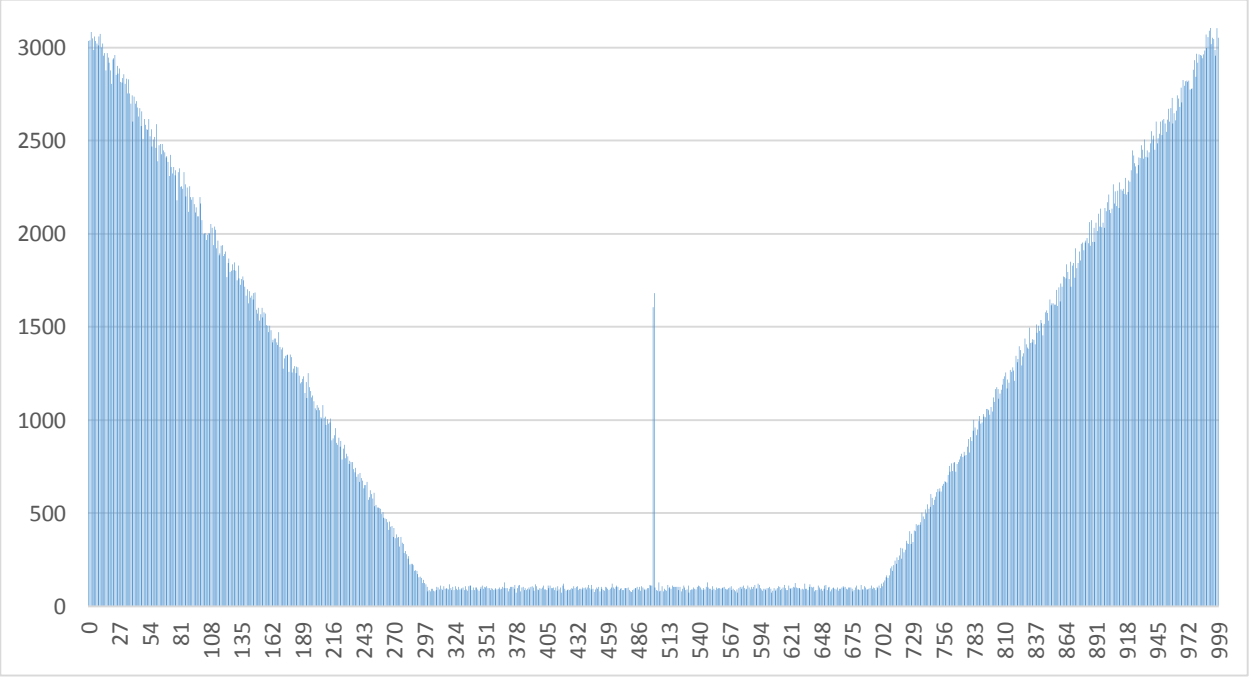


Figure 7. Three different cases for simulating 10^6 positions of recombination events with PWL

Options during new organism evolution

As previously stated, the parameter "*SpeciationGenerations*" (*X*) structures the time frame for the evolution of new organisms through specifying the generation of the first raise for each organism. Let *Xi* the speciation generation for the organism (*i*). The parameter "*NoGeneration2PopSize*" (*G*) describe the number of generations to reach the targeted population size starting from *Xi*. After the generation *Xi*, "*DiSpeciation*" and "*PolySpeciation*" rates can be applied for the organism (*i*) (conditional to the parameter "*OverLap*"). Moreover, the parameter "*NoSpeciations*" (with value = *n*) can be applied for the organism (*i*).

Thus, the user can have one of four options during *Xi* → *Xi* + G:

1. Prohibit any speciation (*NoSpeciations* = 0 & *OverLap* = *No*)
2. Fully random (*NoSpeciations* = 0 & *OverLap* = *Yes*)
3. User-specific number of speciation events (*NoSpeciations* = *n* & *OverLap* = *No*)
4. 2 + 3 (*NoSpeciations* = *n* & *OverLap* = *Yes*)

Examples

Four examples are provided below and they also can be found in PolySim "*Examples*" folder. The command line to run all following examples is:

```
./PolySim $Rep
E.g.:          ./PolySim 1
```

Where $Rep is the replicate number e.g. 1. It is important to give unique replicate ID when running a number of replicates with the same parameter file at the same time to ensure having different random seeds.

- Example 1: Simulating a single diploid population

This is the same simulated data used in Jighly et al. 2017 to generate Figure 2 and Figure S2. The population has 1,000 diploid individuals, with 10,000 SNPs and 1 x 10^{-5} mutation rate. The simulation was run for 15,000 generations of random mating using PolySim.

The content of the parameter file ("*Parameters.txt*") is:

```
MateSystem [1]
OffPerMate [1]
NoSnps [10000]
```

```
CommonAncestorNoGenotypes [1000]
CommonAncestorNoChr [1]
PloidyType [0]
MutationRate [0.00001]
NoGenerations [15000]
Poisson [Yes]
MeanCO [1]
```

- Example 2: Simulating an allohexaploid wheat population

This is the same parameters used in Jighly et al. 2017 (Figure 3) to simulate the allohexaploid species, wheat (2n=6x = 42) using SNP markers adapted from Manickavelu et al. (2014; 2016). The simulation was performed for 1,000 individuals, for each of the allohexaploid genome and its ancestors, with a mutation rate of 1 x 10^{-5} and a 90% selfing rate for 30,000 generations, during which the allohexaploid genome appeared at generation 18,000.

The content of the parameter file ("*Parameters.txt*") is:

```
MateSystem [0.1]
OffPerMate [1]
NoSnps [10000]
CommonAncestorNoGenotypes [1000]
NoGenotypes [1000 1000 1000 1000 1000]
CommonAncestorNoChr [7]
NoChr [7 7 7 14 21]
PloidyType [0]
MutationRate [0.00001]
NoGenerations [30000]
NoGeneration2PopSize [100]
SpeciationGenerations [5000 5200 5400 10000 18000]
DirectHybrid [A B D A_B AB_D]
Poisson [Yes]
MeanCO [1]
```

- Example 3: Simulating the reference scenarios in Jighly et al. 2017

Six reference populations were simulated (three auto- and three allotetraploids) in Jighly et al. (2017) that differed from each other only in terms of outcrossing rate (mating system, MS) with values 0.1, 0.5 and 1, respectively. The reference populations assume no change in recombination rate in polyploids relative to the ancestral diploids (i.e. polyploid recombination change = 1) with a constant population size of 1000 for all organisms. MV rates for reference populations were 0 for allotetraploids and 0.5 for autotetraploids.

The content of the parameter files ("*Parameters.txt*") for MS=1 are:

Reference Allo

```
outFile [out.txt]
PrintProgress [No]
MateSystem [1]
OffPerMate [1]
NoSnps [10000]
Centromere [5000]
CommonAncestorNoGenotypes [1000]
NoGenotypes [1000 1000 1000]
CommonAncestorNoChr [1]
NoChr [1 1 2]
PloidyType [0]
MutationRate [0.00001]
PolySpeciation [0]
DiSpeciation [0]
NoGenerations [25000]
SaveGeneration [1000]
NoGeneration2PopSize [100]
SpeciationGenerations [6000 6200 12000]
DirectHybrid [1 2 1_2]
RecombinPRate [1]
NoSpeciations [0]
OverLap [No]
Poisson [Yes]
MeanCO [1]
```

Reference Auto

```
outFile [out.txt]
PrintProgress [No]
MateSystem [1]
OffPerMate [1]
NoSnps [10000]
Centromere [5000]
CommonAncestorNoGenotypes [1]
NoGenotypes [1000 1000]
CommonAncestorNoChr [1]
NoChr [1 2]
PloidyType [1]
MutationRate [0.00001]
PolySpeciation [0]
DiSpeciation [0]
NoGenerations [20000]
SaveGeneration [1000]
NoGeneration2PopSize [100]
SpeciationGenerations [1 6000]
DirectHybrid [1 1_1]
RecombinPRate [1]
NoSpeciations [0]
OverLap [No]
Poisson [Yes]
MeanCO [1]
Multivalent [11 0.5]
```

- Example 4: Simulating recombination variation along chromosomes in autotetraploids

The below parameter file is the same used by Jighly et al. (2017) to simulate the position of crossover from PWL distribution considering an autotetraploid population with MS=0.1.

The content of the parameter file ("*Parameters.txt*") is:

```
outFile [out.txt]
PrintProgress [No]
MateSystem [0.1]
OffPerMate [1]
NoSnps [10000]
Centromere [5000]
CommonAncestorNoGenotypes [1]
NoGenotypes [1000 1000]
CommonAncestorNoChr [1]
NoChr [1 2]
PloidyType [1]
MutationRate [0.00001]
NoGenerations [20000]
SaveGeneration [1000]
NoGeneration2PopSize [100]
SpeciationGenerations [1 6000]
DirectHybrid [1 1_1]
Multivalent [11 0.5]
OverLap [No]
Poisson [Yes]
MeanCO [1]
PWLRecombination [Yes]
BiSnpIntervals [0 4999 9999]
BiSnpWeights [0.475 0.05 0.475]
MvSnpIntervals [0 2999 6999 9999]
MvSnpWeights [0.475 0.05 0.05 0.475]
```

Output files

If the parameter "*SaveGeneration*" was not specified, PolySim will produce a single file containing all simulated organisms. The name of the file will be:

"Rep_" + "*Repeat*" parameter + "_GenerationNo_" + "*NoGenerations*" paramter + "_" + outFile parameter

For example if we have a total of 15,000 generations:

"Rep_1_GenerationNo_15000_out.txt"

If the parameter "*SaveGeneration*" was specified, PolySim will produce multiple files containing all simulated organisms in different generations. The names of the files will differ only with generation number. For instance, if we set "*SaveGeneration*" to 5,000 in the previous example we will get three different files with the same format:

"Rep_1_GenerationNo_5000_out.txt"
"Rep_1_GenerationNo_10000_out.txt"
"Rep_1_GenerationNo_15000_out.txt"

Output format

The format of PolySim output files is a matrix [M × N] where N = the number of haploid chromosomes or the value of the parameter "*CommonAncestorNoChr*" and each column contains all simulated loci without separators coded as 0 and 1. In other words, each columns contains the haplotype of the whole chromosome.

The number of rows (M) can be calculated with the equation:

$$M = \sum_{i=1}^{x} NoG_i \times P_i$$

Where: x is the total number of simulated organisms; NoG_i is the population size for the organism i and P_i is the ploidy of the organism i.

For instance, if we took the previous example number 3:

1. For the allotetraploid case, we should expect 10,000 rows as we have one tetraploid organism with 1,000 individuals and 3 diploid organisms with 1,000 individuals each.

$$3\times2\times1{,}000 + 4\times1{,}000=10{,}000$$

2. For the autotetraploid case, we should expect 6,002 rows as we have one diploid individual in the common ancestor population, 1,000 diploid individuals and 1,000 tetraploid individuals

$$2\times1 + 2\times1{,}000 + 4\times1{,}000=6{,}002$$

References

Bailey SM, Brenneman MA, Goodwin EH (2004) Frequent recombination in telomeric DNA may extend the proliferative life of telomerase-negative cells. Nucleic acids research 32(12):3743-3751

Bourke PM, Voorrips RE, Visser RG, Maliepaard C (2015) The Double-Reduction Landscape in Tetraploid Potato as Revealed by a High-Density Linkage Map. Genetics 201(3):853-863.

Dufresne F, Stift M, Vergilino R, Mable BK (2014) Recent progress and challenges in population genetics of polyploid organisms: an overview of current state-of-the-art molecular and statistical tools. Molecular Ecology 23:40-69

Hoban S (2014) An overview of the utility of population simulation software in molecular ecology. Molecular Ecology 23:2383-2401

Hoban S, Bertorelle G, Gaggiotti OE (2012) Computer simulation: tools for population and evolutionary genetics. Nature Reviews Genetics 13:110-122

Jighly A, Lin Z, Forster JW, Spangenberg G, Hayes B, Daetwyler HD (2017) Insights into population genetics and evolution of polyploids and their ancestors. Molecular Ecology Resources.

Kovarik A, Renny-Byfield S, Grandbastien MA, Leitch A (2012) Evolutionary implications of genome and karyotype restructuring in *Nicotiana tabacum* L. In Polyploidy and Genome Evolution (pp. 209-224). Springer Berlin Heidelberg.

Mather K (1935) Reductional and equational separation of the chromosomes in bivalents and multivalents. J Genet 30(1):35-78

Manickavelu A, Jighly A, Ban T (2014) Molecular evaluation of orphan Afghan common wheat (*Triticum aestivum* L.) landraces collected by Dr. Kihara using single nucleotide polymorphic markers. BMC Plant Biology 14:320

Manickavelu A, Joukhadar R, Jighly A, Lan C, Huerta-Espino J, Stanikzai AS, et al. (2016) Genome wide association mapping of stripe rust resistance in Afghan wheat landraces. Plant Science 252:222-229

Myers S, Bottolo L, Freeman C, McVean G, Donnelly P (2005) A fine-scale map of recombination rates and hotspots across the human genome. Science 310(5746):321-324.

Pecinka A, Fang W, Rehmsmeier M, Levy AA, Scheid OM (2011) Polyploidization increases meiotic recombination frequency in Arabidopsis. Bmc Biology 9:24

Peng B, Kimmel M, Amos CI (2008) Forward-time simulations of non-random mating populations using simuPOP. Bioinformatics 24(11):1408–1409

Ramsey J, Schemske DW (2002) Neopolyploidy in flowering plants. Annu Rev Ecol Syst 33(1):589–639

Rehmsmeier M (2013) A Computational Approach to Developing Mathematical Models of Polyploid Meiosis. Genetics 193:1083-1094

Stebbins GL (1947) Types of Polyploids: their classification and significance. Adv Genet 1:403-429

Stebbins GL (1971) Chromosomal evolution in higher plants [by] G. Ledyard Stebbins. Contemporary biology, vol Accessed from http://nla.gov.au/nla.cat-vn1859678. Edward Arnold, London

Sybenga A (1996) Chromosome pairing affinity and quadrivalent formation in Polyploids: Do segmental allopolyploids exist? Genome 39:11776-1184

Voorrips RE, Maliepaard CA (2012) The simulation of meiosis in diploid and tetraploid organisms using various genetic models. BMC Bioinformatics 13:248