

SIMHYB 2: a software tool to explore and illustrate evolutionary forces in Population Genetics teaching and research

Álvaro Soto de Viana¹

David Rodríguez Martínez

Unai López de Heredia Larrea¹

¹GIE “Arboreto de Montes”

Dpto. Sistemas y Recursos Naturales.

ETSI Montes, Forestal y del Medio Natural

Universidad Politécnica de Madrid.

USER MANUAL

<http://github.com/GGFHF/SIMHYB2>

July, 2023

CONTENTS

Overview

Simulation process

Getting started

Input files and parameters

Neutral loci allelic frequencies

Self-incompatibility loci

Specific classes and fitness coefficient

Fertility table

Fitness inheritance

Ageing

Output files

References

How to cite this program

OVERVIEW

SIMHYB is a Java-based software for the simulation of mixed hybridizing populations. The program is intended for the analysis of the effect of the different demographic, adaptive and reproductive factors on the evolution of these populations. Census size of each species, immigrants, number of intermediate specific classes, directional fertility among them and fitness coefficient of each class can be defined by the user. Inheritance of fitness and ageing effect are also taken into account. The software generates individuals of known pedigree, allowing their traceability throughout the generations. SIMHYB 2 yields for each simulated generation an output file easily convertible to inputs for GENALEX (Peakall & Smouse, 2012) or STRUCTURE (Pritchard *et al.*, 2000), some of the most popular softwares for population genetics and Bayesian analysis.

Initially, SIMHYB 2 was intended for hermaphrodite long-living plants, such as trees (so it considers features such as the presence of chloroplast, overlapping generations or the possibility of self-incompatibility processes), but it can be also applied for other organisms, taking into account certain considerations.

SIMULATION PROCESS

SIMHYB 2 simulates the evolution of a population of constant size with up to two different diploid genetic groups (we will refer to them as “species”, for simplicity), which may hybridize, depending on the conditions defined by the user. The population consists of a fixed number of individuals, each of them identified by a vector including pedigree, a so-called “specific coefficient” (representing the expected contribution of each species to the individual genome), the complete genotype for a user-defined number of loci and other information.

Initial population

The user can define the initial population size for each species, considering only adult individuals. The size of the global population will remain constant throughout the simulation. Genotypes of

the individuals of the initial population are generated drafting alleles according to their frequencies in each species, from the input files described above. A chloroplast and a mitochondria are assigned to each species, and recorded for each individual. “Null” values are recorded as the parents of the initial population individuals. Individual fitness values are assigned randomly from the continuous interval $(f_{sp}-\varepsilon_{sp}, f_{sp}+\varepsilon_{sp})$, being f_{sp} the average value for the specific category and ε_{sp} a *within category variability* defined by the user.

Reproductive events

Each cycle or generation includes five phases: 1) reproduction, 2) migration, 3) ageing, 4) selection, and 5) standardization. For **1) reproduction**, no spatial restriction is considered in the current version of SIMHYB 2, i.e., pollen from any individual (if applied to trees) can reach the flowers of any other tree with the same probability. The spatial positions of the individuals are not reckoned in SIMHYB 2 and, accordingly, there is no isolation by distance. The probability of obtaining a viable cross is firstly determined by the fertility coefficients defined in the input file `fertility.txt`, according to the specific classes of the mother tree and the pollen donor. After that, effective pollination is limited by self-incompatibility, if this option is selected. SIMHYB 2 considers gametophytic self-incompatibility driven by a single locus. This way, at pollination, one of the alleles at this locus from the pollen donor is randomly selected and compared with both alleles of the mother tree. After passing these barriers, the new individual will carry at each locus one of the alleles of each parent, randomly selected, and the chloroplast and mitochondria of the corresponding parent, as specified by the user. The species coefficient of the new individual will be the average of the parents’ values, so it becomes a precise estimation of the contribution of each genetic pool (pure species) to the genome of the individual. A new fitness coefficient will be assigned to the new individual according to the following formula:

$$f = w_h f_h + w_{sp} f_{sp}$$

$$f_h = \frac{f_{mother} + f_{father}}{2} + \delta$$

where f_{sp} is the fitness coefficient corresponding to the specific class of the new individual, f_h is the fitness inherited from the parent trees; δ is drafted from the interval $(-\varepsilon_h, \varepsilon_h)$, being ε_h the *within sibling variability* parameter, to include variability among full-sibs, and w_h and w_{sp} are weights, established by the user, so that $w_h + w_{sp} = 1$.

At this point, **2) immigration** can take place, according to a user-defined probability (between 0 and 1). The user also defines the number of individuals in the immigrant pool (they can be different number for each species). The current version of SIMHYB 2 only considers the *continent-island* migration model. Thus, the evolving population would be the *island*, which can incorporate immigrants from a *continent* whose allelic frequencies do not vary during the simulation. Immigrants are created in the same way that the initial population, but a .csv input file with the allelic frequencies for the *continent* population, with the same loci and alleles (it can be the same file as for the *island*).

3) Ageing affects each individual fitness every cycle, according to this formula:

$$f_t = f_{t-1} (1 - a)b + f_0 (1 - a)^t (1 - b)$$

where f_t is the fitness coefficient t cycles after birth, f_0 is the initial fitness, at the time of birth, a is the ageing coefficient, and b is the linearity coefficient, which varies between 0 and 1.

4) Selection takes place at this moment: the N (population size, defined by the user) individuals with the highest fitness coefficients are selected and remain in the reproductive population, while the other individuals die. Finally, **5) standardization** is performed, so that fitness coefficients vary between 0 and 1 (this way, if high b and w_h have been selected, certain individuals can keep very high fitness values along the generations).

End of simulation

Three alternative criteria can be followed to finish the simulation: (1) a user-defined number of cycles; (2) the presence of individuals of a user-defined number of specific categories in the simulation; and (3) fixation of a chloroplast type in the population.

GETTING STARTED

SIMHYB 2 is programmed in Java 16, and runs in any computer with an OS that allows for Java (<https://www.java.com/>), or OpenJDK (<http://openjdk.java.net/>): Linux/Unix, Microsoft Windows, Mac OS X, and other platforms, including cloud virtual desktops that operate in academic institutions to facilitate e-learning (Moser et al., 2014). The application provides a user-friendly front-end for ease of use by students. Download and unzip the package and copy the executable file, `SIMHYB.jar`, and the `simhyb.properties` file in the desired location of your computer. Every time you launch a simulation, this file will be overwritten with the parameters and instructions of the current simulation. It is therefore recommended to save this file together with the results for your own records. You don't have to worry about the content of the first copy of the file, included in `InstallSIMHYB.rar` since it will be overwritten, but the executable needs to have this file to begin the simulations. If your computer has a Linux based OS of your computer, we recommend modifying the path of the input files by editing the following lines in the `simhyb.properties` file:

```
specificCategoryFile=/YOURPATH/spcat_fitness.txt  
alleleFrequenciesFile=/YOURPATH/allelefrequencies.csv  
immigrantAlleleFrequenciesFile=/YOURPATH/allelefrequenciesContinent.csv  
selfIncompatibilityLocusFile=/YOURPATH/selfincompatibility.csv  
fertilityTableFile=/YOURPATH/fertility.txt
```

Before launching the program, create a folder named **logs** at the same directory where the executable is located.

It is also recommended to place all the needed input files in the same directory where the executable is located.

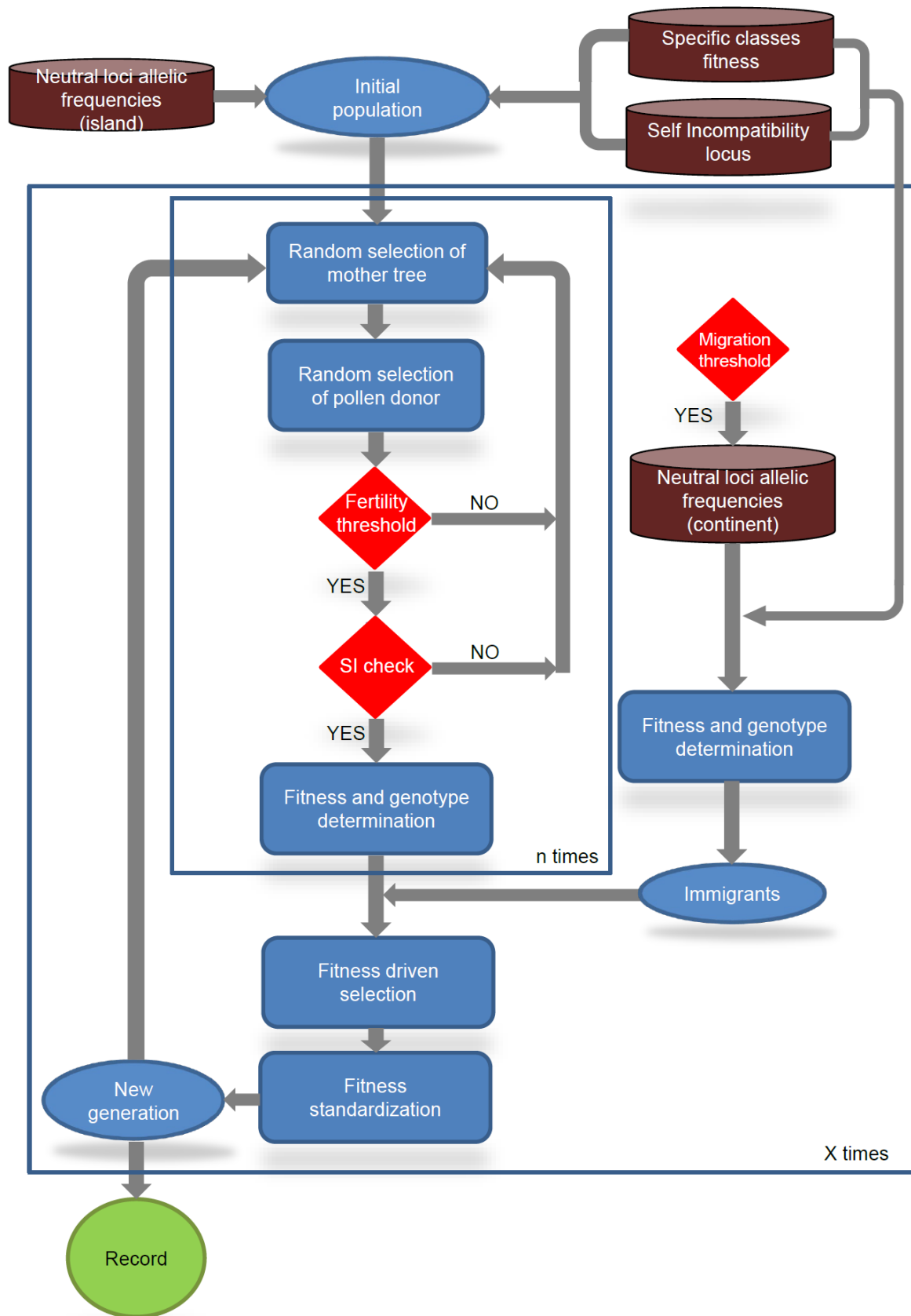


Figure 1. SimHyb 2 flowchart. **X** represents the number of reproductive cycles and **n** is the number of reproductive events (matings or offspring) per cycle.

INPUT FILES AND PARAMETERS

When `SIMHYB2.jar` is executed, and before starting the simulation, an interface window appears where the user must define the simulation parameters and input files (Figure 2).

The SimHyb interface window is titled "SimHyb" and contains several sections for defining simulation parameters:

- Population size:** Fields for Species 1 and Species 2.
- Allele frequencies:** Fields for File and Number of loci.
- Reproductive events:** A dropdown menu for Type (set to "Matings") and a field for Number per cycle.
- Fertility table:** A large text area for input.
- Self-incompatibility:** A checkbox.
- Self-incompatibility locus file:** A text field.
- Plastidial inheritance:** A dropdown menu set to "maternal".
- Mitochondrial inheritance:** A dropdown menu set to "maternal".
- Immigrant population:** Fields for Probability of immigration per cycle, Number of immigrants per cycle, Species 1, Species 2, and Allele frequencies of immigrant population.
- Fitness inheritance:** Fields for Specific category fitness table, Within category variability, Sp. category weight, Within sibling variability, Sibling weight, Ageing coefficient, and Ageing lineality.
- Output:** Fields for Output directory and Snapshot frequency.
- End of simulation:** A dropdown menu for Criterion (set to "# reproductive cycles") and a field for Number.

At the bottom right, there are "Run" and "Cancel" buttons.

Figure 2. The SimHyb interface. When the simulation is launched, after selecting the input files and parameters, a pop-up window appears.

Population size

The user can define the population size for each species, considering only adult individuals. The size of the global population will be fixed throughout the simulation.

Neutral loci

The user must also provide an `allelefrequencies.csv` file with the loci and the allele frequencies of each species. The format is exemplified in Figure 3. The interface allows considering just a subsample of the loci in the simulations, using the same input file.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	197	0.04348	209	0	206	0	197	0.02055	209	0.00685	206	0.02055	
2	199	0.02174	219	0.85507	210	0	199	0	219	0.86986	210	0.0137	
3	201	0.63768	223	0.14493	212	0.39855	201	0.67808	223	0.12329	212	0.62329	
4	211	0.04348	0	0	214	0.2971	211	0	0	0	214	0.21918	
5	213	0.05797	0	0	216	0.00725	213	0	0	0	216	0.00685	
6	215	0.11594	0	0	218	0.02899	215	0.30137	0	0	218	0	
7	221	0.07971	0	0	220	0.10145	221	0	0	0	220	0.0137	
8	0	0	0	0	222	0.02899	0	0	0	0	222	0.0137	
9	0	0	0	0	224	0.13768	0	0	0	0	224	0.03425	
10	0	0	0	0	226	0	0	0	0	0	226	0.05479	
11													
12													
13													
14													
15													
16													
17													
18													
19													

Figure 3. An example of the `allelefrequencies.csv` file, opened in Windows Excel. Each loci (just three, in this example) is represented by two columns. The first one includes the allele names and the second one its frequencies in the species. All the loci for the first species are presented in consecutive columns and then, in the same order, they appear for the other species. Please notice the columns do not have headings.

Specific classes

The user can define as many intermediate specific classes as desired, intermediate between the two pure species included in the original population, and can define the limits of each class, according to the so called “species coefficient” in a `spcat_fitness.txt` file. In this file, each row corresponds to a specific class, defined by a number, a name for the class, its

boundaries in terms of the species coefficient, and the fitness coefficient corresponding to the specific class (f_{sp}). The route must be specified in the line “Specific category fitness table”, in the interface. See an example in Figure 4.

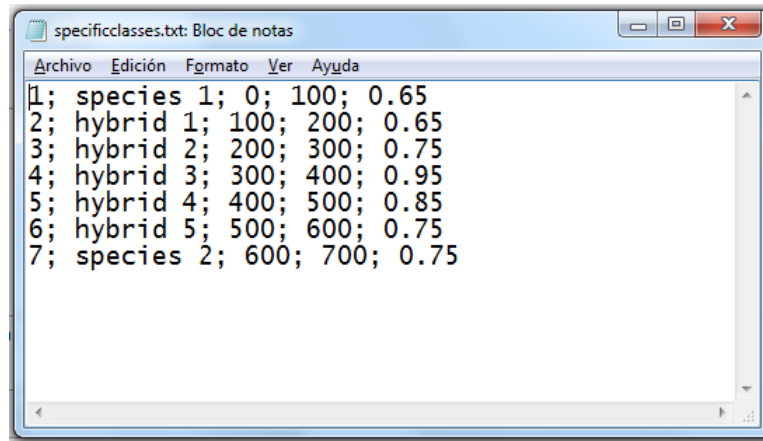


Figure 4. An example of the `spcat_fitness.txt` file, opened here in Windows Notepad. In each row, the first figure is the identification of the specific category, followed by its description; the next two numbers correspond to the category boundaries, according to the specific coefficient; the last number is the fitness coefficient associated to the specific category. In this example, seven categories are defined, two corresponding to the parental species (1 and 7) and with 5 intermediate, hybrid categories.

Fertility

Fertility within and between specific classes takes into account the pollination direction. Figure 5 shows an example of the `fertility.txt` file. Each row corresponds to a specific class, acting as mother tree, while each column corresponds to a specific class acting as pollen donor.

Organelles inheritance

Chloroplast and mitochondria can be inherited, independently, from the mother or from the father, according to user specifications.

	A	B	C	D	E	F	G	H	I	J
1	#	1 (father)	2	3	4	5	6	7		
2	1	0.5	0	0	0	0	0	1		
3	2	0	0	0	0	0	0	0		
4	3	0	0	0	0	0	0	0		
5	4	0	0	0	0.5	0.7	0.7	1		
6	5	0	0	0	0.7	0.7	0.7	1		
7	6	0	0	0	0.7	0.7	0.7	1		
8	7	1	0	0	1	1	1	1		
9										
10										
11										

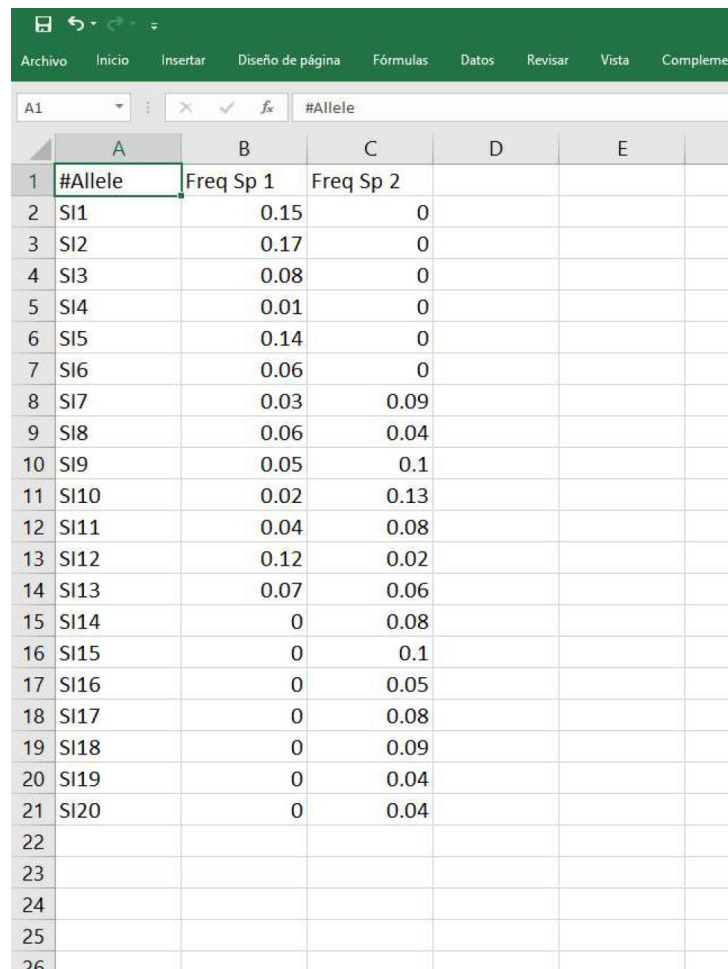
Figure 5. An example of the fertility.txt file, opened here in Windows Excel. This example corresponds to 7 specific classes (two parental species, 1 and 7, and 5 intermediate, hybrid categories). In columns, pollen donors, and in rows, mother trees. Please notice asymmetric fertility is allowed.

Self-incompatibility

The user can select the option of considering self-incompatibility in the simulations. The current version of SIMHYB 2 only takes into account gametophytic self-incompatibility. Figure 6 shows an example of the `selfincompatibility.csv` file.

End of simulation

The user can select to finish the simulation when one of the chloroplasts is completely replaced by the other one, when a specified number of specific categories remain in the population or after a desired number of cycles. When the simulation finishes, a pop-up window appears.



	A	B	C	D	E
1	#Allele	Freq Sp 1	Freq Sp 2		
2	SI1	0.15	0		
3	SI2	0.17	0		
4	SI3	0.08	0		
5	SI4	0.01	0		
6	SI5	0.14	0		
7	SI6	0.06	0		
8	SI7	0.03	0.09		
9	SI8	0.06	0.04		
10	SI9	0.05	0.1		
11	SI10	0.02	0.13		
12	SI11	0.04	0.08		
13	SI12	0.12	0.02		
14	SI13	0.07	0.06		
15	SI14	0	0.08		
16	SI15	0	0.1		
17	SI16	0	0.05		
18	SI17	0	0.08		
19	SI18	0	0.09		
20	SI19	0	0.04		
21	SI20	0	0.04		
22					
23					
24					
25					
26					

Figure 6. An example of the selfincompatibility.csv file

OUTPUT FILES

SIMHYB 2 provides two output files. The first one includes all the individuals in the population every X cycles, as a sort of “snapshot album” of the evolving population. The user can define the frequency of these “snapshots” and the output can be consulted while the simulation is still running. The second output file is available only when the simulation is finished, and includes all the individuals of the population in each cycle.

Output files are provided as .csv files. The first row register the headings of the first 10 columns, which include different information of each individual (see below). The second row includes the name of the nuclear, diploid loci, starting with the SI locus and followed by the neutral loci. Third

and successive rows include the virtual individuals. The first 10 positions include the following information: (1) Individual ID (integer); (2) Specific coefficient (numeric character); (3) Father individual ID (“null” value for individuals in the first generation and for immigrants); (4) Mother individual ID (“null” value for individuals in the first generation and for immigrants); (5) Chloroplast (A or B); (6) Mitochondria (A or B); (7) Generation (integer; cycle of the simulation); (8) Birth Generation (integer; cycle in which the individual is added to the population); (9) Death Generation (integer; cycle in which the individual is removed from population; “-1” for survivors); and (10) individual fitness value in that generation (number, between 0 and 1). The next positions register the diploid genotype of the individual, starting with the self-incompatibility locus (alphanumeric characters, two alleles) and neutral loci (alphanumeric characters, two alleles per locus) (Figure 7).

Therefore, this output file is easily convertible into the input file of different and widely used population genetics software tools. For instance, just modification of the heading rows and removing some of the first columns would be needed to get an input file for GENALEX (Peakall & Smouse, 2012) or NEWHYBRIDS (Anderson & Thompson, 2002). In a similar way, removing the first row is enough to get an input file for STRUCTURE (Pritchard *et al.*, 2000), where columns 3-10 or 3-12 can be labelled as “extra columns”. Of course, non-desired rows (according to the analysis to be performed), such as repeated individuals, present in the population for several cycles (where the only variation would be registered in the relative fitness value), must also be removed accordingly.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Id	Sp. Coef.	Father	Mother	Chloropl	Mitochor	Generati	Birth Ger	Death Ge	Fitness						
2	Self-inco	locus 1	locus 2	locus 3	locus 4	locus 5	locus 6	locus 7	locus 8	locus 9						
306	272	700	132	185	B	B	3	2	4	0.0678	SI12	SI2	197	197	203	207
307	262	525	137	128	B	B	3	2	4	0.07	SI3	SI1	197	215	207	211
308	269	700	167	128	B	B	3	2	4	0.1177	SI2	SI13	197	197	207	205
309	286	350	165	162	B	B	3	2	4	0.1413	SI6	SI10	197	197	219	219
310	297	525	172	167	B	B	3	2	4	0.1476	SI9	SI12	197	201	211	219
311	247	700	134	176	B	B	3	2	4	0.1838	SI7	SI13	197	197	203	207
312	210	700	170	139	B	B	3	2	4	0.1853	SI3	SI12	197	197	205	207
313	242	700	169	128	B	B	3	2	4	0.2024	SI2	SI13	197	197	205	205
314	338	350	159	144	B	B	3	3	4	0.2029	SI5	SI3	215	211	211	203
315	248	700	170	104	B	B	3	2	4	0.2031	SI2	SI3	197	197	203	207
316	305	525	107	168	B	B	3	3	4	0.2065	SI2	SI12	197	213	209	219
317	395	350	173	144	B	B	3	3	4	0.2279	SI7	SI3	215	201	219	219
318	399	525	173	287	B	B	3	3	4	0.2286	SI5	SI1	197	197	215	207
319	352	437	150	233	B	B	3	3	4	0.24	SI8	SI9	197	197	219	219
320	396	525	173	243	B	B	3	3	4	0.2415	SI7	SI12	197	197	203	207
321	288	350	192	193	B	B	3	2	4	0.2496	SI11	SI5	197	197	219	219
322	315	437	236	129	B	B	3	3	4	0.2791	SI1	SI12	215	197	203	207
323	118	700	93	56	B	B	3	1	4	0.2825	SI2	SI5	197	197	191	211
324	386	612	262	147	B	B	3	3	4	0.2871	SI3	SI1	197	197	207	207
325	356	525	179	243	B	B	3	3	4	0.2883	SI2	SI1	197	197	211	219
326	187	700	58	71	B	B	3	1	4	0.2891	SI11	SI2	197	197	203	211
327	202	525	173	199	B	B	3	2	4	0.2892	SI7	SI3	197	197	207	219
328	320	612	236	199	B	B	3	3	4	0.2935	SI5	SI3	197	201	201	207
329	254	525	123	159	B	B	3	2	4	0.3003	SI2	SI5	211	197	219	199

Figure 7. An example of SimHyb output file.

REFERENCES

- Anderson EC, Thompson EA (2002). A model-based method for identifying species hybrids using multilocus genetic data. *Genetics* 160: 1217-1229. DOI: 10.1093/genetics/160.3.1217
- Peakall R, Smouse PE (2012). GenAlEx 6.5: genetic analysis in Excel. Population genetic software for teaching and research. *Bioinformatics* 28: 2537-2539. DOI: 10.1093/bioinformatics/bts460
- Pritchard JK, Stephens M, Donnelly P (2000). Inference of population structure using multilocus genotype data. *Genetics* 155(2): 945-959. DOI: 10.1093/genetics/155.2.945

HOW TO CITE THIS PROGRAM

Soto A, Rodríguez-Martínez D, López de Heredia U (2023) SIMHYB 2: a software tool to explore and illustrate evolutionary forces in Population Genetics teaching and research