GSpace version 1.0

User manual

December 17, 2020

GSpace is a computer program for the simulation of allelic and sequence data at multiple chromosomes under general isolation by distance models. It is based on a backward "generation by generation" coalescent/recombination algorithm allowing the consideration of various isolation by distance models on a lattice. GSpace can consider a large panel of subdivided population models representing populations with or without a demic structure, the latter case being an approximation for a population of individuals dispersed over a continuous habitat. Many dispersal distributions can be considered as well as heterogeneities in space of the demographic parameters. Typical applications of our program include (1) the study of the effect of various sampling, mutational and demographic factors on the pattern of genomic variation at different spatial scales; (2) the production of test data sets to assess the influence of these factors on any inferential method available to analyze genotypic or sequence data; and (3) the development of simulation-based inference pipelines by coupling the simulation of genetic data sets with the computation of summary statistics and simulation-based inference algorithms (e.g. ABC, approximate Bayesian computation).
GSpace is freely available on the INRAe website for the command-line version that can be compiled under any system using a C++17 ISO compiler. This command-line version of GSpace runs on Windows, MacOS X and Linux distributions. We also provide all source code, including unit and functional test modules that only run on MacOs and Linux, for future developments at the GitHub repository.

1 Requirements

1.1 Source compilation for various OS

The program GSpace is available for download on the INRAe website and is provided as original source code. Recompile the sources using a compiler that handles C++17 using the following command (or equivalent) :

```
compiler -O3 -std=gnu++17 -o GSpace *.cpp
```

This should work on Microsoft OS and most Unix-based systems, including MacOS. For advanced users or developers, a GitHub repository GitHub repository is available. Compilation of the GitHub sources needs to be done trough CMake (minimum version 3.9) using the following commands from the main directory of the downloaded sources :

```
mkdir build
cd build
cmake -DCMAKE_FUNC_TEST=OFF -DCMAKE_BUILD_TYPE=Release ..
make
ctest -j 6 --timeout 160000 (verify integrity of install)
```

Using CMake compilation gives access to all unit tests that check the integrity of various functions of the program, with -DCMAKE_FUNC_TEST=OFF, or to functional tests that check the reliability of the program using using various settings with known expectations, with -DCMAKE_FUNC_TEST=ON. Unit tests are relatively quick but running functional tests can lead to high computation times (e.g. around 12 hours with the current configuration).

1.2 Hardware

GSpace should run on any reasonably recent computer and has limited memory needs for most reasonable settings. The lattice, sub-population, and sample sizes (i.e. number of individuals and loci) are formally limited only by the maximum C++ integer value (2^{32}), however high values will increase memory usage and decrease execution speed. Reasonable simulation times are usually obtained even with reasonably large lattices, population sizes and sample sizes (e.g few minutes for 10 chromosomes of 1000 loci/sites for 1000 individuals evolving on a 100 × 100 lattice with sub-population sizes of 100 individuals). Finally, simulating a large number of loci, or long sequences, with high recombination rate (e.g. $\geq 10^{-7}$ between loci/sites per generation) strongly increase memory usage and execution time. However, millions of loci/sites can be easily simulated on any recent machine as it only requires a few Go's of RAM and few hours of simulation.

2 Your first GSpace session: a simple example

In this section, we describe a simple example, so that you can directly get in touch with the software while starting to read in detail the whole documentation.

For this example, we consider a demographic model of isolation by distance in 2 dimensions with 20 × 20 sub-populations, each sub-population being a panmictic unit with 30 haploid individuals (i.e. isolation by distance with a demic structure). Dispersal is simulated using a stepping stone migration model (i.e. migration only occurs between adjacent populations) with a migration rate of 0.05. Ten data files are simulated for a sample size of 20 haploid individuals taken from 4 populations located on a 2 × 2 square in the center of the lattice. For each sampled individual, we simulate 5 independent chromosomes with 3 linked loci with a $5 \cdot 10^{-5}$ recombination rate and evolving under a SSM mutation model with a mutation rate of $5 \cdot 10^{-4}$ mutation per locus per generation. By default, edge effects are `reflecting`.

The content of the setting file `GSpaceSettings_firstSessionExample.txt` with all the above described parameters is the following:

```
%%%%%%%% SIMULATION SETTINGS %%%%%%%%%%%%%%%
Data_filename=Example
Run_Number=10

%%%%%%%% OUTPUT FILE FORMAT SETTINGS %%%%%%%

Genepop=true

%%%%%%%% MARKERS SETTINGS %%%%%%%%%%%%%%%%%%
Chromosome_number=5
Mutation_Rate=0.0005
Mutation_Model=SMM
Allelic_Lower_Bound=1
Allelic_Upper_Bound=200
Sequence_Size = 3

%%%%%%%% RECOMBINATION SETTINGS %%%%%%%%%%%%
Recombination_Rate=0.0005

%%%%%%%% DEMOGRAPHIC SETTINGS %%%%%%%%%%%%%%
%% LATTICE
Lattice_SizeX=20
Lattice_SizeY=20
Ind_Per_Pop=30

%% DISPERSAL
```

```
Dispersal_Distribution=uniform
Dist_max = 1,1
Total_Emigration_Rate=0.05

%%%%%%%% SAMPLE SETTINGS %%%%%%%%%%%%%%%%%%%
Sample_SizeX=2
Sample_SizeY=2
Min_Sample_CoordinateX=9
Min_Sample_CoordinateY=12
Ind_Per_Pop_Sampled=5
```

First, copy the provided `GSpaceSettings_firstSessionExample.txt` into an empty folder and rename it `GSpaceSettings.txt` (be careful to respect capital letters under Linux; it does not matter for Windows or MacOS). Make the `GSpace` executable accessible from this folder either by copying the executable file or launching it from a distant folder e.g.
`../ExecutableFolder/GSpace` .

Launch the executable and wait for the completion of the computation which should last only a few seconds. The output files generated by `GSpace` are :

1. 10 text files named Example_GP_1.txt to Example_GP_10.txt for the 10 simulated data sets;
2. a file named `Example_GSpace_param_summary.txt` with a summary of the input settings and some information about dispersal distributions;
3. a file named `cmdline_settings.txt` with a copy of the line of command use to launch `GSpace`. This file help to track modifications of `GSpaceSettings.txt` options by command line arguments, more details in section 4

Each of the simulated data sets is written as a `Genepop` input file and has the following format (example for output file Example_GP_1.txt, see section 4.2.2 for more details on `Genepop` file format):

```
This file has been generated by the GSpace program.
locus1_smm
locus2_smm
locus3_smm
locus4_smm
locus5_smm
locus6_smm
locus7_smm
locus8_smm
```

```
locus9_smm
locus10_smm
locus11_smm
locus12_smm
locus13_smm
locus14_smm
locus15_smm
pop
9 12 , 026 200 007 115 108 017 128 084 171 179 091 139 187 063 059
9 12 , 026 200 005 120 105 011 129 084 169 183 093 130 186 070 050
9 12 , 030 199 010 115 108 017 129 084 169 183 092 130 186 070 050
9 12 , 026 200 005 126 114 008 129 084 169 184 089 139 191 072 059
9 12 , 026 200 007 119 105 012 129 084 169 183 092 130 186 070 050
pop
9 13 , 032 198 009 114 108 014 125 084 171 184 089 139 185 068 051
9 13 , 035 199 008 119 108 013 125 084 171 184 089 139 187 070 050
9 13 , 032 198 009 114 108 013 126 086 170 184 089 139 185 068 051
9 13 , 030 199 010 115 108 017 128 084 171 184 089 139 187 066 052
9 13 , 034 200 009 119 108 013 126 085 170 181 085 135 185 069 047
pop
10 12 , 026 200 007 114 108 018 128 085 171 183 092 130 192 064 058
10 12 , 026 200 005 118 105 017 130 084 171 182 086 139 191 072 059
10 12 , 030 199 010 118 105 017 128 086 170 181 088 139 185 070 047
10 12 , 026 200 005 123 108 018 126 086 170 182 086 139 185 070 047
10 12 , 030 199 010 119 106 011 126 086 170 182 086 139 186 070 050
pop
10 13 , 033 198 007 114 108 014 129 086 164 182 090 140 192 064 058
10 13 , 030 199 010 114 108 014 128 085 171 182 085 129 191 072 057
10 13 , 032 199 007 114 108 014 128 086 170 182 085 129 187 066 052
10 13 , 026 200 005 115 108 015 126 085 170 183 092 130 191 072 059
10 13 , 033 198 007 114 108 014 128 085 171 182 090 140 193 064 058
```

3 Principle of the simulation algorithm and implemented models

3.1 Coalescent and recombination algorithm

The GSpace program is based on a backward-in-time coalescent with recombination approach, which is well known for allowing the development of efficient exact simulation tools (Hudson, 1983, 1993). Such an approach allows the generation of large genetic data sets and the consideration of complex migration schemes including those with arbitrary heterogeneities in space and time of the demographic parameters.

For neutral genes, the coalescent process depends solely on the demographic history of the population and is independent from mutational processes. So we first generate the genealogy of the sampled genes going backward in time and then simulate mutations starting from the top of the coalescent tree (i.e. MRCA: Most Recent Common Ancestor) and adding them independently along all branches of the tree.

The coalescent algorithm used to build the genealogical tree is not based on the large-N approximations of the n-coalescent theory (Kingman, 1982; Nordborg, 2007). It is rather an exact algorithm for which coalescence, recombination and migration events are considered generation by generation until all common ancestors has been found, to build the ancestral recombination graph (ARG, Hudson 1983) of the whole simulated sample. The idea of tracing lineages back in time, generation by generation, is fundamental in the coalescence theory, and is well described in Nordborg (2007). Such a generation-by-generation algorithm leads to less efficient simulations in terms of computation time than those based on the n-coalescent theory. However, this algorithm is much more flexible when complex demographic and dispersal features are considered. In particular, it can consider small deme sizes down to 1, which represents a population without a demic structure, also called models of "continuous habitat" (a population of individuals dispersed over a continuous habitat).

`GSpace` combines some parts of the modified Hudson's algorithm for recombination and coalescence implemented in `msprime` (Kelleher *et al.*, 2016) with previous features from `IBDSim` (Leblois *et al.*, 2009). For a single non-recombining locus, the generation-by-generation algorithm that gives the coalescent tree for a sample of n genes evolving under IBD has been detailed in Leblois *et al.* (2003, 2004, 2006) and the main ideas underlying the global algorithm are summarized in Leblois *et al.* (2009).

The original algorithm using continuous-time approximations (i.e. the n-coalescent approximations) described in Kelleher *et al.* (2016) has also been implemented but it does not handle migration yet (parameter `Approximate_time`, see section 4.2).

These algorithms and the program were checked by comparing simulated values of probabilities of identity of two genes under models of isolation by distance on finite lattices with their exact analytically computed values (e.g. Malécot 1975 for the lattice model) adapted to different mutation models following Rousset (1996). Recombination has been specifically tested by comparing simulated values of 2-locus joined probabilities of identity under a Wright-Fisher model with their exact analytically computed values (e.g. Vitalis & Couvet 2001).

3.2 Migration models

Specifying a dispersal model proceeds in several steps. One must first specify a forward dispersal distribution, describing emigration probabilities to different distances (`Dispersal_Distribution` setting). Next one must specify how habitat boundary effects are handled (`Edge_Effects` setting). These will typically affect the immigration probability in each deme; for example, demes at the boundary may receive fewer immigrants than more central demes. However, even the immigration probability in central demes may be reduced compared to the emigration probability of the unbounded forward distribution.

3.2.1 Forward dispersal distributions

Biologically realistic dispersal functions often have a high kurtosis (Endler, 1977; Kot *et al.*, 1996). However, commonly used discrete probability distributions are not the most appropriate ones for isolation by distance because they imply that high kurtosis can be achieved only by assuming a low dispersal probability, i.e. that most offspring reproduce exactly where their parents reproduced (Rousset, 2000). Therefore, we have implemented two less well-known families of dispersal distributions that allow high kurtosis and high migration rates: the discrete version of the Pareto family, and the Sichel family (Chesson & Lee, 2005). The better known uniform, geometric and discrete Gaussian families are also implemented. For two-dimensional models, we assume that dispersal is independent in each direction, so that $f_{dx,dy} = f_{dx} \cdot f_{dy}$.

The first family of distributions are truncated variants of the discrete Pareto, or Zeta, distribution (see e.g. Patil & Joshi, 1968) with the probability of moving k steps (for $-d_{\max} \leq k \leq d_{\max}$, $k \neq 0$) in one direction being of the form:

$$f_k = \frac{M}{2 \cdot |k|^n} \tag{1}$$

with parameters M and n, controlling the total dispersal rate and the kurtosis, respectively and $d_{\max}$ being the maximal dispersal distance.

The second family of dispersal distributions is obtained as mixtures of convolutions of stepping stone steps and is a convenient way to model discrete distributions with various forms (Chesson & Lee, 2005). As detailed in that paper, the Sichel mixture is described by three parameters, ξ, ω and γ. Parameterization of the Sichel mixture distribution is not trivial but details on each parameter and formulas to compute various moments of the distribution as well as its kernel are given in Chesson & Lee (2005). Both the full three-parameter distribution, and the long-tailed variant of this family obtained in the limit case $\omega \to 0$, $\xi \to \infty$ with $\omega\xi \to \kappa$ (reciprocal Gamma mixture) are implemented. In the latter case the two parameters γ and κ then describes a family of distributions which are Gaussian-looking at short distances but have tails proportional to $r^{-2\gamma-1}$ for distance r. The values of γ and κ can be chosen so as to achieve some given second moment (σ) and kurtosis. For more details on the Sichel distribution parametrization, see Watts *et al.* (2007) and Chesson & Lee (2005).

The third family of distributions are uniform dispersal distributions for which the probability of moving k steps (for $-d_{\max} \leq k \leq d_{\max}$, $k \neq 0$) in one direction is :

$$f_k = \frac{m}{d_{\max}}, \tag{2}$$

with m controlling the total emigration rate. The finite island model of dispersal can be simulated by assuming a 1-dimensional lattice with circular boundary condition and a uniform distribution with $d_{\max}$ equals the number of sub-populations minus 1.

The fourth family of distributions are discretized Gaussian dispersal distributions for which the probability of moving k steps (for $-d_{\max} \leq k \leq d_{\max}$, $k \neq 0$) in one dimension is :

$$f_k = d2^{-2d_{\max}} \binom{k + d_{\max}}{2d_{\max}},$$

with m controlling the total emigration rate and

$$d = \frac{m}{1 - 2^{-2d_{\max}} \binom{d_{\max}}{2d_{\max}}}.$$

These discretized Gaussian distributions have mean 0 and axial variance $\sigma^2 = d_{\max}.d/2$. More details on these discretized Gaussian distributions can be found in Annexes of Rousset (1997).

The fifth family of distributions are Geometric dispersal distributions for which the probability of moving k steps (for $-d_{\max} \leq k \leq d_{\max}$, $k \neq 0$) in one direction is :

$$f_k = \frac{M}{2} g^{|k|-1}, \tag{3}$$

with m controlling the total emigration rate and g the shape of the distribution. The stepping stone dispersal is the limit of the geometric distribution with $g \to 0$, and the finite island model of dispersal is the limit of the geometric distribution with $g \to 1$.

The above dispersal distributions can be selected by values of the Dispersal_Distribution setting listed below, where in each case we describe the additional parameters to be specified. Details on the default values and syntax for the additional dispersal parameters are given p.25.

Dispersal_Distribution=u or Uniform: custom uniform distribution with total emigration rate and maximal distance of dispersion set in the input file by the keywords Total_Emigration_Rate and Dist_max respectively.

Dispersal_Distribution=n or Gaussian: custom discretized Gaussian distribution with total emigration rate and shape set in the input file by the keywords Total_Emigration_Rate and Dist_max respectively.

Dispersal_Distribution=g or Geometric: custom geometric distribution with total emigration rate and shape set in the input file by the keywords Total_Emigration_Rate, Geometric_Shape and Dist_max respectively. Note that high kurtosis cannot be achieved with a geometric distribution without small emigration rates.

Dispersal_Distribution=p or Pareto: custom truncated Pareto distribution with parameters M and n set in the input file by the keywords Total_Emigration_Rate and Pareto_Shape, respectively.

Dispersal_Distribution=s or Sichel: custom Sichel mixture distribution with parameters ξ, ω and γ set in the input file by the keywords Sichel_Gamma, Sichel_Xi, Sichel_Omega. Some parameter values which gives biologically realistic dispersal distributions can be found in Watts *et al.* (2007). Detailed description of this distribution is given in Chesson & Lee (2005).

3.2.2 Spatial distribution of individuals and habitat shape

GSpace considers isolation by distance models on a lattice, and not on continuous space, strictly speaking (but see Robledo-Arnuncio & Rousset 2010 for details on continuous and lattice models). GSpace can either consider models with demic structure, i.e. where each lattice node is a panmictic population of size N individuals; or models of so-called “continuous habitat”, a population of individuals dispersed over a continuous habitat, where each lattice node is a single individual.

Mathematical analyzes of isolation by distance models usually consider lattice models without edge effect (i.e. on a circle or a torus in one and two dimensions respectively, Fig. 1) to have complete homogeneity in space, which strongly facilitates analytical developments. However, as torus or circle

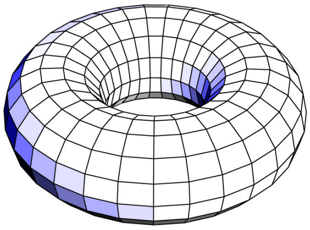

Figure 1: Graphical representation of a torus

models are not generally realistic, we implemented various edge effects in GSpace:

- reflective boundaries (`Edge_effects=reflecting`) : the lattice is represented on a line or plane and trajectories of dispersal events going outside the lattice are reflected on edges as light is reflected on a mirror;

- no edges (`Edge_effects=circular`) : the lattice is represented on a circle or a torus for a one or a two-dimensional model respectively;

- absorbing boundaries (`Edge_effects=absorbing`) : the lattice is represented on a line or plane and all individuals that emigrate (forward) out of the habitat are lost (i.e. the probability mass of coming (backward) outside the lattice is equally shared on all other movements inside the lattice).

3.2.3 Effects of edges and heterogeneous density to gene flow on backward distributions

We used the "backward" dispersal distribution in the coalescent algorithm because the position of the parental gene is chosen conditionally on the position of its descendant gene. This "backward" function is computed using $f_{dx,dy}$, the forward dispersal density function describing where descendants go. In the simplest case, considering that density is homogeneous in space backward dispersal functions are equal to forward dispersal functions, so that $b_{dx} = f_{dx}$ for one-dimension models and $b_{dx,dy} = f_{dx,dy} = f_{dx} \cdot f_{dy}$ for two-dimensional habitat with independent dispersal in each dimension.

However, when density is not homogeneous in space, backward and forward dispersal differ. In this case, each lattice node has a backward distribution that depends on the density of each surrounding node. Those surrounding nodes correspond to all locations from which genes could have come in one generation (forward in time). Since those nodes are occupied by different numbers of individuals and because nodes occupied by more individuals contribute potentially more to the number of immigrants that reach a given node, we have to weight each term of the backward dispersal distribution by the number of individuals of the node where immigrants come from. Then for any node $\mathbf{z}_1$ the probability $b_{\mathbf{z}_1,\mathbf{z}_2}$ that a gene is immigrant from $\mathbf{z}_2$ is equal to

$$b_{\mathbf{z}_1,\mathbf{z}_2} = \frac{N_{\mathbf{z}_2} f_{\mathbf{z}_1-\mathbf{z}_2}}{\sum N_{\mathbf{z}} f_{\mathbf{z}_1-\mathbf{z}}} \tag{4}$$

where the sum is over all possible non-empty (as implied by $N_{\mathbf{z}}$) nodes $\mathbf{z}$ that are defined inside the lattice and within a distance from the focal node smaller than their `Dist_max` .

The realized immigration rate in any lattice node may only be remotely related to the `Total_Emigration_Rate` setting of the forward dispersal distribution, first as the result of edge effects (with absorbing boundaries, the immigration rate will be lower than the forward emigration rate as some emigrants are lost outside the habitat), second (in two dimensions) because `Total_Emigration_Rate` only gives the one-dimensional emigration rate and, without edge effects, the two-dimensional non-dispersal rate will rather be the square of 1−`Total_Emigration_Rate`.

3.2.4 Postdispersal sampling, life cycle

For all simulations, the life cycle is divided into five steps:

(i) each individual gives birth to a great number of recombining gametes, and dies;

(ii) gametes undergo the effect of mutations;

(iii) migration may occur for haploids;

(iv) diploid individuals are formed, if necessary, by considering random union of gametes within each deme, and

(v) competition brings back the number of adults in each deme to N;

(vi) migration may occur for diploids.

Samples simulated by `GSpace` are composed of individuals sampled after the last dispersal step in the life cycle.

3.3 Mutation models

GSpace can simulate allelic (e.g. microsatellite) and DNA sequence data. Currently 10 mutation models are implemented in GSpace and all settings for the genetic markers are given in 4.2.3. The implemented models are the following:

3.3.1 Allelic models

1. IAM: the infinite allele model (IAM, Kimura & Crow, 1964) in which each mutation gives rise to a new allele;

2. KAM: the K-allele model (KAM, Crow & Kimura, 1970) in which a mutation changes the initial allelic state into one of $K - 1$ other possible states;

3. SMM: the strict stepwise mutation model (SMM, Ohta & Kimura, 1973), much applied to microsatellite markers, where each mutation adds or removes a repeated unit to the mutated allele;

4. GSM: the generalized stepwise model (GSM, e.g. Pritchard *et al.*, 1999), where each mutation adds or removes X repeated units to the mutated allele. X is randomly chosen from a geometric distribution with parameter $pGSM$ (pGSM).

3.3.2 Nucleotidic models

6. JCM: the Jukes-Cantor model (JC69, Jukes & Cantor, 1969), is a nucleotide substitution model where the equilibrium frequencies of the purine bases (A and G) and the pyrimidine bases (C and T) are assumed to be in an equal proportion ($i.e.$ $\pi_A = \pi_G = \pi_C = \pi_T = \frac{1}{4}$) and the rate of substitution from any given base to the other three bases is the same irrespective of the ancestral or the mutant base;

7. K80: the two parameter Kimura model (K80 or K2P,Kimura, 1980), is a simple generalisation of the JC69 model where the nucleotide substitutions are categorized as either transitions ($i.e.$ A↔G, C↔T) or as transversions ($i.e.$ A↔C, A↔T, G↔C,G↔T). Real data typically contains some form of a transition-transversion bias which can be explained by the relative rates of these two categories of substitutions. This bias can be specified in GSpace using the ratio of the rate of transition substitutions over transversion substitutions for every substitution (*see also* 4.2.3). In the absence of a transition-transversion bias, (ex. the JC69 model), this value is by default

$\frac{1}{2}$, as a given ancestral base (*say* G) has always three possible mutant states; one of which is always a transition substitution (*in this case* A) and the other two are transversion substitutions (*i.e.* C or T).

8. `F81`: the Felsenstein'81 model (F81,Felsenstein, 1981), is another generalisation of the JC69 model where we do not distinguish between transitions and transversions and instead allow the equilibrium base frequencies to vary (*i.e.* $\pi_A \neq \pi_G \neq \pi_C \neq \pi_T$). In this case the transition-transversion ratio needn't be specified as it is the same as that of the JC69 model (*i.e.* $\frac{1}{2}$);

9. `HKY85`: the Hasegawa, Kishino and Yano model (HKY85, Hasegawa *et al.*, 1985), integrates aspects of both the K80/K2P and F81 models by respectively letting the equilibrium base frequencies to be variable and the transition-transversion bias to be specified;

10. `TN93`: the Tamura-Nei model (TN93, Tamura & Nei, 1993) is the most general of the substitution models available in `GSpace` which further generalizes the HKY85 model by allowing an additional bias to be introduced between purine transitions and pyrimidine transitions. This can be specified in terms of the ratio of the rate of `A`↔`G` substitutions over that of `C`↔`T` substitutions for every substitution (*see also* 4.2.3).

4 All `GSpace` settings

4.1 Input file format

`GSpace` reads a single generic text file (ASCII), whose default name is `"GSpaceSettings.txt"`, which must be in the active folder (respecting capitalization under Linux). The file is read at the beginning of each execution and allows one to control all settings of `GSpace`. It contains lines of the form `keyword=value(s)` or of the form `keyword=value1,value2,value3`, where value(s) and keyword(s) can take various formats as described below. Note that all booleans will be evaluated using a convenient procedure allowing the following symbols/keywords: `T, True, Yes, Y, F, False, No, N`. All settings values and their defaults are explained in details in the following subsections. The default name of the setting file is `GSpaceSettings.txt` but you can change this through the command line (again, caring for capitalization): running `GSpace` using `Setting_Filename=mysettings.txt` will make the program read `mysettings.txt` rather than `GSpaceSettings.txt` (Note that complete path can be included in the filename).

The settings read from the file can be modified by command line settings. For example the command `GSpace Chromosome_number=4 Mutation_Rate=0.001` can be used to provide or modify the settings `Chromosome_number` and `Mutation_Rate`.

4.1.1 A complete example

Here, we present a complete (i.e with almost all keywords) settings file in the `GSpace` format. :

```
%%%%%%%% SIMULATION SETTINGS %%%%%%%%%%%%%%%
Setting_Filename = GSpaceSettings.txt
Random_Seeds = 5
Run_Number = 2

%%%%%%%% OUTPUT FILE FORMAT SETTINGS %%%%%%%
Output_Dir = .
Data_File_Name = ContPop
Data_File_Extension = .txt
Genepop = True
Genepop_Ind_File = False
Group_All_Samples = No
Genepop_No_Coord = F
VCF = No
Coord_File = N
Sequence_Characteristics_File = False
Fasta = N
Fasta_Single_Line_Seq = No
Phylip = N
Approximate_time = False

%%%%%%%% MARKERS SETTINGS %%%%%%%%%%%%%%%%%%
Ploidy = Haploid
Chromosome_number = 5
Mutation_Rate = 0.0005
Mutation_Model = SMM
Allelic_Lower_Bound = 1
Allelic_Upper_Bound = 200
Sequence_Length = 100

%%%%%%%% RECOMBINATION SETTINGS %%%%%%%%%%%%
Recombination_Rate = 0.0005

%%%%%%%% DEMOGRAPHIC SETTINGS %%%%%%%%%%%%%%
%% LATTICE
Node_Size_Matrix = No
Lattice_Size_X = 500
Lattice_Size_Y = 500
Ind_Per_Node = 1
```

```
%% DISPERSAL
Migration_Matrix = No
Dispersal_Distribution = g
Geometric_Shape = 0.2
Edge_Effects = circular
Total_Emigration_Rate = 0.2
Disp_Dist_Max = 100,100

%%%%%%%% SAMPLE SETTINGS %%%%%%%%%%%%%%%%%%
Sample_Size_X = 15
Sample_Size_Y = 15
Min_Sample_Coord_X = 200
Min_Sample_Coord_Y = 200
Void_Sample_Node_X = 1
Void_Sample_Node_Y = 1
Ind_Per_Node_Sampled = 1

%%%%%% VARIOUS COMPUTATION OPTION S%%%%%%%%%
%The code below can be specified in a single line %
Diagnostic_Tables = Prob_Id_2Loc, Prob_Id_1Loc,
Effective_Dispersal, Iterative, Per_Loc, Per_Chrom
Dist_Class_Nbr = 25
```

4.2 Description of all settings

All parameter names, values or keywords specified before the brackets [ ] in this documentation will correspond to the default parameter names/values/ keywords implemented in GSpace. Other possible inputs are given between brackets [ ] after the default values. Alternative parameter names mostly correspond to IBDSim parameter names that have been modified in GSpace for clarity but correspond to the same settings. They are conserved in GSpace for compatibility between the two software.

e.g. Some_Param[Alternative_name]=default_val [or other_val1 or other_val2 or...]

4.2.1 Simulation parameters

All settings in this category are quite straightforward to understand:

- Setting_Filename = GSpaceSettings.txt is the name of the setting file use for the current simulation.

- `Random_Seeds = 3` are the seeds for the random number generator. Different runs with precisely the same parameter values and same seeds will give exactly the same results. Note that two different versions of `GSpace` can output different results with the same seeds and the same version of `GSpace` compiled with different compilers or with the same compiler on a different operating system (Linux, IOS, Windows) can also lead to different result with the same seeds.

- `Run_Number = 0` tells `GSpace` to run a given number of iterations, i.e. a given number of simulated data sets with multiple chromosomes/loci, here 0.

- `Approximate_Time = false` [or `true`] tells `GSpace` to simulate genomics samples with a modified Hudson's algorithm (see section 3.1). At the time of writing, this option is incompatible with more than one population.

- `Pause=Final` [or `Never` or `OnError`] tells `GSpace` when to stop the program and wait for a user intervention (Press a key) to resume. The `Default` behavior under Windows is that `GSpace` pause at the end of the run, letting the terminal/command window open until the user presses any key (i.e. `Pause=Final`). Under Linux/MacOs, the default behavior is `Pause=Never` meanning that `GSpace` will never for user intervention. `OnError` means that `GSpace` will pause on each error/warning. The recommended settings for batch simulations is `Pause=Never`.

4.2.2 Data set format settings

These settings set the different data file formats and names to be generated for each data set simulated by `GSpace`. These data files can be then analyzed by any other programs than can read one of the four following formats.

- `Output_Dir = .` is the directory name for the simulated data sets. Absolute or relative path (relative to the location of the program) can be provided.

- `Data_File_Name` [`Output_Filename`] = `GSpace_Simu` is the generic file name for the simulated data sets. This generic file name will be incremented with the number of the run. Example: simulated output file name number 56 will be named here `GSpace_Simu_56` .

- `Data_file_extension` [`Output_file_extension`] = `txt` tells `GSpace` to add a `.txt` extension to each simulated data file. Example: if not set, simulated data file number 56 will be named here `GSpace_Simu_56.txt` .

- Genepop = false [or true] tells GSpace whether to write each data file in Genepop format (actually the extended input file format of Genepop v.4; Rousset, 2008). Genepop format is compatible with allelic mutation models and not with nucleotidic models (see section 3.3). Here is an example :

  ```
  example of input file for Genepop
  loc1
  loc2
  pop
  0.56 8.67, 0101 0102
  pop
  1.67 8.5, 0101 0102
  ```

  where each line represents the genotype of one individual at different loci, and groups of individuals (samples from different populations/nodes) are separated by pop statements. For each population sample, the values before the coma of the last individual indicates geographic coordinates of the populations. When the setting Genepop_Group_All_Samples is set to True, no pop indicator is placed between geographic samples and all individuals are thus represented as coming from a single population. On the opposite, when the setting Genepop_Ind_File is set to True, a pop indicator is placed between each simulated individual. See the Genepop documentation for details and examples.

- Genepop_ind_file = false [or true] output the Genepop file with a unique individual per pop even if they have the same coordinates. Not compatible with Genepop_group_all_samples.

- Genepop_group_all_samples= false [or true] outputs the Genepop file with all individual in the same pop even if they have the differents coordinates. Not compatible with Genepop_Ind_File.

- Genepop_no_coord = false [or true] outputs the Genepop file with no coordinates for individual. Instead individual are identifies by they rank in GSpace.

- VCF = false [or true] tells GSpace whether to write each data file in the widely-used VCF format (v.4.3). VCF format is compatible with nucleotidic data and allelic data with few alleles, but not for data with a large number of alleles. For tis reason, we only implemented the VCF format for DNA sequence mutation models but not for allelic ones (see section 3.3). Here is an example :

  ```
  ##fileformat=VCFv4.3
  ##FORMAT=<ID=GT, Number=1,Type=String,Description="Genotype">
  ```

```
##SAMPLE=<ID=Indiv1,Assay=WholeGenome,Description="Individual
from pop[1;1]">
##SAMPLE=<ID=Indiv2,Assay=WholeGenome,Description="Individual
from pop[1;2]">
##SAMPLE=<ID=Indiv3,Assay=WholeGenome,Description="Individual
from pop[2;1]">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT INDIV0 INDIV1 INDIV2
1 1 . C A,G . . . GT 1|0 2|1 1|0
1 2 . A C . . . GT 1|1 1|1 1|0
1 3 . A C . . . GT 0|0 1|1 0|0
```

where each line represents a site/locus, and output informations about its ancestral state and all its derived states observed in the sample, followed by the phased genotype of each sampled individual at this site/locus. See the VCF documentation for details and examples.

- `Coordinate_file` [`Coordinate_file`] `= false` [or `true`] outputs, for each simulated sample, a text file with two columns, each line being the two coordinates (X, Y) of each sampled individual, in the same order than in the output data files.

- `Sequence_characteristics_file` [`Seq_char_file`] `= false` [or `true`] outputs, for each simulated sample, some information about each simulated haplotype in a text file with six columns :
  `ind_Nb chrom_Nb phase_Nb X_coord Y_coord mut_nb`,
  each line carrying the information for one simulated haplotype at one chromosome of one sampled individual. This file contains some information that is often needed to complete the genetic information written in the sequence output files, notably for the Phylip format that does not allow to write more information than the DNA sequence itself.

- `Fasta = false` [or `true`] tells `GSpace` whether to write each data file in `Fasta` format. `Fasta` format is compatible with DNA sequence simulation but not with allelic mutation models (see section 3.3).
  Here is an example :

```
>Gspace_TestX0_1_ancestral_sequence_chrom_1_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestX0_1_ancestral_sequence_chrom_2_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestX0_1_ind_1_coord_1_1_chrom_1_1_mutNbr_2
GGGTTAGACGCCACAGTTTCTGCCCTAACCTTTGTTACAGCCTGGGA
>Gspace_TestX0_1_ind_1_coord_1_1_chrom_1_2_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestX0_1_ind_1_coord_1_1_chrom_2_1_mutNbr_0
```

```
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestXO_1_ind_1_coord_1_1_chrom_2_2_mutNbr_1
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCTTTGGA
>Gspace_TestXO_1_ind_2_coord_1_4_chrom_1_1_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestXO_1_ind_2_coord_1_4_chrom_1_2_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestXO_1_ind_2_coord_1_4_chrom_2_1_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestXO_1_ind_2_coord_1_4_chrom_2_2_mutNbr_1
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCTTTGGA
>Gspace_TestXO_1_ind_3_coord_4_1_chrom_1_1_mutNbr_2
GGGTTAGACGCCACAGTTTCTGCCCTAACCTTTGTTACAGCCTGGGA
>Gspace_TestXO_1_ind_3_coord_4_1_chrom_1_2_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestXO_1_ind_3_coord_4_1_chrom_2_1_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestXO_1_ind_3_coord_4_1_chrom_2_2_mutNbr_1
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCTTTGGA
>Gspace_TestXO_1_ind_4_coord_4_4_chrom_1_1_mutNbr_0
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
>Gspace_TestXO_1_ind_4_coord_4_4_chrom_1_2_mutNbr_2
GGGTTAGACGCCACAGTTTCTGCCCTAACCTTTGTTACAGCCTGGGA
>Gspace_TestXO_1_ind_4_coord_4_4_chrom_2_1_mutNbr_1
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCTTTGGA
>Gspace_TestXO_1_ind_4_coord_4_4_chrom_2_2_mutNbr_1
GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCTTTGGA
```

where each haplotype of each chromosome of each simulated individual is written as a block composed of : (i) a first line, beginning with " > ", being the description of the sequence (defline). When simulated with `GSpace` this line contains straightforward information about the simulation and the sequence itself (e.g. simulation name, run number, individual number, coordinates, chromosome number, phase, number of mutation compared to the ancestral MRCA sequence) separated by underscores; followed by (ii) lines of the sequence, each line being by default shorter than 80 characters in length (but the complete sequence can be written on a single line using the setting `Fasta_Single_Line_Seq=True`). See here for details and examples.

- `Fasta_Single_Line_Seq = false` [or `true`] tells `GSpace` to write each DNA sequences in the `Fasta` format as a set of lines of 80 characters at most (as usually required by the `Fasta` format), or if set to `True`, it allows `GSpace` to write the whole sequence on a single line.

- `Phylip = false` [or `true`] tells `GSpace` whether to write the simulated data for all individuals, but for each chromosome separately, in a file in Phylip format. Phylip format is compatible with DNA sequence simulation but not with allelic mutation models (see section 3.3).

Here is an example :

```
9 47
Anc_1     GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
1_1_1     GGGTTAGACGCCACAGTTTCTGCCCTAACCTTTGTTACAGCCTGGGA
1_1_2     GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
2_1_1     GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
2_1_2     GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
3_1_1     GGGTTAGACGCCACAGTTTCTGCCCTAACCTTTGTTACAGCCTGGGA
3_1_2     GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
4_1_1     GGGTTAGACGCCACAGTATCTGCCCTAACCTTTGTTACAGCCTTGGA
4_1_2     GGGTTAGACGCCACAGTTTCTGCCCTAACCTTTGTTACAGCCTGGGA
```

where the first line (Header) describes the dimensions of the alignment: two positive integers separated by one space specify the number of sequences (i.e., the number of rows in the alignment) and the length of the sequences (i.e., the number of columns in the alignment). Then, the alignment section consists of n lines, one (for haploids) or two (for diploids) for each individual for the current chromosome/locus. Each row consists of a 10-character sequence identifier (for GSpace, it represents the individual number, the chromosome number and the "phase" of the sequence, completed with spaces) followed by the sequence itself. See here for details and examples. As this format does not allow to store a large quantity of information for each sequence/individual, additionnal information for each sequence can be output in an other file using the setting `Sequence_characteristics_file = true` described above.

4.2.3 Genetic marker parameters

In this section most of the settings deal with the parametrization of the genetic markers simulated by GSpace (see 3.3).

- `Ploidy  = Haploid` [or `Diploid`] set the level of ploïdy for all the loci/site simulated using the same setting file. Note that the diploid model assumes individual dispersal and the haploid model assumes gametic dispersal (see section 3.2.4).

- `Chromosome_Nbr` [`Locus_Nbr`] `= 1` is the number of chromosomes that will be simulated for each sampled individual in the data set. These chromosomes are independant in the genetic sense (i.e. recombination rate between them is 0.5) but they all disperse together within a gamete or an individual.

- Sequence_Size=50 defines the size of the sequence or the number of loci par chromosome to be simulated by GSpace.

- Mutation_Model = IAM [or KAM or SMM or GSM or JCM or K80 or F81 or HKY or TN93] sets the mutation model for all loci. See 3.3 for a general description of all the models.

- Mutation_Rate=0.0005 is the mutation rate per generation per locus/ per site for all simulated loci/sites.

- Recombination_Rate = 0.005 is the recombination rate per generation between loci/sites within a chromosome for all simulated loci/sites and chromosomes.

- MRCA_Sequence = 0 [or ATGACAGTACAGATTAGAATAGAC or ATGACA,GTAGCAT,AGTTGTA] is the user-specified MRCA sequence for nucleotidic mutation models. It can be constituted of one or more (one for each simulated chromosome) comma separated sequences of ATCG of length Sequence_Size (i.e. one nucleotide for each site). If this setting has not been specified by the user or if MRCA_Sequence = 0 GSpacewill generated random MRCA sequences for each chromosome by drawing from a uniform base frequencies. If MRCA_Sequence=ATGACA,0,TGTGTA, only the nucleotidic states for chromosome 2 are randomly assigned.

- MRCA_Allelic_States [Allelic_States_MRCA] = 0 [or any integer between Allelic_Lower_Bound and Allelic_Upper_Bound or a vector of vectors of integers, e.g. 20,30,40;10,20,30;40,50,60] sets the allelic state of the MRCA for the allelic mutation models (KAM, SMM and GSM). It can be constituted of one or more (one for each simulated chromosome) semicolon separated sequences, each of them being constituted of separated comma integer values (one for each locus) of length Sequence_Size. If this setting has not been specified by the user or if MRCA_Allelic_State = 0, the MRCA allelic states for all loci for all chromosomes are randomly generated by drawing from the uniform equilibrium allelic frequency distribution between Allelic_Lower_Bound and Allelic_Upper_Bound. If MRCA_Allelic_Sate=0;10,20,30;0, the allelic states for all loci of chromosome 1 and 3 are randomly assigned.

- Allelic_Lower_Bound = 1 sets the lowest possible allelic state for the mutation models KAM, SMM and GSM.

- `Allelic_Upper_Bound = 10` sets the largest possible allelic state for the mutation models KAM, SMM, and GSM (*e.g.* for a KAM, the number of possible allelic states K is then given by K=`Allelic_Upper_Bound` - `Allelic_Lower_Bound` +1).

- `P_GSM = 0.22` sets the parameter of the geometric distribution from which X is drawn. The GSM model is a TPM model with `SMM_Probability_In_TPM`=0.

- `Transition_Transversion_ratio = 0.5` (also known as a Ti/Tv bias) is the user-specified probability of a transition substitution against a transversion substitution and applies only to the K80/K2P, HKY85 and TN93 models. (*see* pg. 13). In the absence of a user-specified Ti/Tv value `GSpace` uses 0.5 (the value for the JC69 and F81 models). This value means that for every nucleotidic substitution there are twice as many possible transversions than transitions.

- `Transition1_Transition2_ratio = 1` is the user-specified probability of a purine (A or G) transition substitution against a pyrimidine (C or T) transition substitution (*see* pg. 13). This setting only applies to the TN93 model.

- `Equilibrium_Frequencies=0.2 0.3 0.2 0.3` is the default vector of population base frequencies of A, G, C and T respectively. The frequencies should sum up to unity. This setting only applies to the F81, HKY85 and TN93 models where the equilibrium base frequencies can differ from each other.

4.2.4 Demographic parameters

In this section all settings for the demographic part of the model are specified (e.g. deme sizes, habitat dimensions, dispersal characteristics). Note that `GSpace` simulates gametic or individual dispersal depending on the simulated ploidy level (see section 3.2.4).

- `Lattice_Size_X = 1` is the lattice size in the first dimension X.

- `Lattice_Size_Y = 1` is the lattice size in the second dimension Y.

- `Lattice_Boundaries` [`Edge_Effects`] `= Reflecting` [or `Circular` or
`Absorbing`] sets the habitat (i.e. lattice) boundaries type to be considered for the entire simulation. See section 3.2.2 for details on the different possible habitat boundaries implemented in `GSpace`.

- Ind_Per_Node [Ind_Per_Pop] = 1 is number of individual per lattice node that GSpace will consider. It also correspond to the density in number of individuals per lattice node.

- Node_Size_Matrix [Specific_Density_Design] = false [or true] tells GSpace to consider (1) homogeneous density on the lattice if set to false ; Or (2) a user specific density configuration of the lattice where each node of the lattice has a number of individuals (i.e. deme size) specified in a file named Node_Size_Matrix.txt, if set to true. The default name of the specific density file can be changed this through :

  - Node_Size_Matrix_Filename [Density_Filename] = Node_Size_Matrix.txt [or any textfile name]. Node_Size_Matrix_Filename = My_Matrix_File.txt will make the program read My_Matrix_File.txt rather than the default Node_Size_Matrix.txt (Note that complete path can be included in the filename).

  The format of Node_Size_Matrix.txt is a matrix of integers separated by spaces, tabs, semi-clons or comma with Lattice_Size_Y rows and Y=Lattice_Size_X columns. The file begins with coordinate x=1 and y=Lattice_Size_Y in the upper left corner, and ends with coordinate x=Lattice_Size_X and y=1 in the lower right corner, so that the matrix is a right side up image of the lattice (be careful, it is not the case in IBDSim).

- Dispersal_Distribution = none [or Uniform or Gaussian or Geometric or Pareto or Sichel or ss or u or n or g or p or s] Its argument is a character, either a letter, referring to one of the implemented dispersal distributions. This setting tells GSpace to consider one of the custom distribution. Detailed descriptions of all implemented dispersal distribution and parameters of these distributions are given in section 3.2.1. The following settings are thus only described here in terms of keywords and default values.

- Geometric_Shape=0.5 is the shape parameter value of the geometric distribution (see p.9).

- Pareto_Shape=5 is the shape parameter value of the custom truncated Pareto distribution (see the description of truncated Pareto distribution on p.8).

- Sichel_Gamma=-2.15 is the first parameter of the Sichel distribution (see the complete Sichel distribution description p.9).

- `Sichel_Xi=100` is the second parameter of the Sichel distribution.
- `Sichel_Omega=-1` is the third parameter of the Sichel distribution.
- `Total_Emigration_Rate` [`Emigration_Rate`] `= 0` is the total emigration rate (i.e. probability to disperse) for all migrations models.
- `Disp_Dist_Max` [`Dist_Max`] `= 0` is the $d_{\max}$ maximum distance parameter for all migrations models.
- `Migration_Matrix = false` [or `true`] tells `GSpace` to consider (1, if set to `false`) homogeneous dispersal parameters over the whole lattice, based on the settings for dispersal distributions described above; or (2, if set to `true`) a user specific forward migration rate matrix where each cell $m_{i,j}$ of the matrix is a real number describing the forward migration rate from node i with coordinates (x_f, y_f) ($i = 1 + (x_f - 1) *$ (`Lattice_Size_X`+1) $+ (y_f - 1)$) to node j with coordinates (x_t, y_t) ($j = 1 + (x_t - 1) *$ (`Lattice_Size_X`+1) $+ (y_t - 1)$). The matrix begins with the indices (1,1) (corresponding to forward migration rate from node 1 (1,1) to node 1 (1,1)) in the upper left corner and finishes with the indices (`Lattice_Size_X`*`Lattice_Size_Y`, `Lattice_Size_X`*`Lattice_Size_Y`) (corresponding to forward migration rate from node `Lattice_Size_X`*`Lattice_Size_Y` (`Lattice_Size_X`, `Lattice_Size_Y`) to node `Lattice_Size_X`*`Lattice_Size_Y` (`Lattice_Size_X`, `Lattice_Size_Y`)) in the lower right corner. By default, this migration matrix is read from a file named `Migration_Matrix.txt`, and this default name of the specific forward migration matrix file can be changed this through :
  - `Migration_Matrix_Filename = Migration_Matrix.txt` [or any textfile name].
    `Migration_Matrix_Filename = My_Matrix_Filename.txt` will make the program read the `My_Matrix_Filename.txt` file rather than the default `Migration_Matrix.txt` file (Note that complete path can be included in the filename).

4.2.5 Sample parameters

In this section all settings for the sample configuration are specified. These sample parameters have to be compatible with some of the demographic settings detailed in the previous sections. `GSpace` simulates postdispersal samples (i.e. genes are samples after the last dispersal event in the life cycle).

All settings for simulating samples are listed below.

- Sample_Coordinates_X = 2,5,7,9,10,12,21,34,56 [No Default values] is a comma separated list of specific X coordinates for a user defined specific sample design. Need to have the same size as Sample_Coordinates_Y.

- Sample_Coordinates_Y = 4,8,15,17,20,26,34,50,56 [No Default values] is a list of specific comma separated Y coordinates for a user defined specific sample design. Need to have the same size as Sample_Coordinates_X. If Sample_Coordinates_X and Sample_Coordinates_Y are set, all other sampling parameters will not be used.

- Sample_Size_X = 1 is the axial number of sampled nodes in dimension X when a restangular sample is simulated.

- Sample_Size_Y = 1 is the axial number of sampled nodes in dimension Y when a restangular sample is simulated.

- Min_Sample_Coord_X [Min_Sample_Coordinate_X] = 1 is the coordinate of the most left sampled node in dimension X when a restangular sample is simulated.

- Min_Sample_Coordinate_Y [Min_Sample_Coordinate_Y] = 1 is the coordinate of the most left sampled node in dimension Y when a restangular sample is simulated.

- Ind_Per_Node_Sampled [Ind_Per_Pop_Sampled] = 1 [or a single or a vector of any integer values, e.g. 2,1,3,10,22] is the number of individuals sampled on each lattice node within the sampled area or for each sampled coordinate. It can be a single value, in which case, the sample size is equal to this this value for all sampled nodes. It can also be a vector (comma or semi-colon separated values) of size the number of sample nodes (i.e. Sample_Size_X * Sample_Size_Y or length of Sample_Coordinates_X/Y), in which case, each sample node has a specific sample size. If the sample coordinates are specified using a rectangle sample (i.e. using Sample_Size_X, Sample_Size_Y, Min_Sample_Coord_X, Min_Sample_Coord_Y and Void_Sample_Nodes), this vector of sample sizes is ordered beginning from the lower left corner of the sampled nodes, going up on the Y coordinates for each X coordinate, and then left on the next X coordinate, up to the upper right corner of the sampled nodes. If the sample coordinates are specified using Sample_Coordinates_X, Sample_Coordinates_Y and Void_Sample_Nodes, then the vector of sample sizes is ordered similarly to the coordinate vectors.

- Void_Sample_Node_X [Void_Sample_Pop_X] = 1 controls whether, when a rectangular sample is simulated, every node in X dimension within the previously designed sampling area is sampled or not. With a value of 1 GSpace will sample all node (in X dimension) within the sampling area, with a value of 2 GSpace will only sample one node every second, etc...

- Void_Sample_Node_Y [Void_Sample_Pop_Y] = = 1 controls whether every node in Y dimension within the previously designed sampling area is sampled or not. With a value of 1 GSpace will sample all node (in Y dimension) within the sampling area, with a value of 2 GSpace will only sample one node every second, etc...

- Void_Sample_Node [Void_Sample_Pop] = 1 controls whether every node within the previously designed sampling area is sampled or not. With a value of 1 GSpace will sample all nodes within the sampling area, with a value of 2 GSpace will only sample one node every second, etc...

4.2.6 Various computational settings

GSpace uses a single keyword Post_Sim_Computations followed by one specific keyword for each computation or mode of computation setting (the order is not important).

- Post_Sim_Computations [DiagnosticTables] = [Prob_Id_2Loc, Prob_Id_1Loc, Effective_Dispersal, Iterative, Per_Loc, Per_Chrom] tells GSpace whether to compute various statistics on each simulated data with their mean and variances over loci/sites, and/or over chromosomes and/or all simulated data sets and to report them in the output files Data_File_Name_Global_Stats.txt for values among all runs, Data_File_Name_Y_Stats_per_chr.txt for values among chromosomes for the simulated data set Y, and Data_File_Name_Y_chr_X_Stats_per_loc.txt for values among loci for chromosome X and data set Y. These settings are not compatible with all sampling designs and/or demographic mode and some computations require a value for the parameter Dist_Class_Nbr described below. See the details of each setting below for more details:

  - Prob_Id_1Loc [Identity_Probability_1_Locus] (Qr, Qwi, Qwd, Qbd) tells GSpace to compute, for each simulated data file, identity probabilities (Id Prob) for the pairs of sampled genes. This setting is essentially implemented to plot the evolution of the mean identity

probabilities through the run to check the program against analytical expectations.
Q_r : Id Prob for pairs of genes within a each distance class, which number is set by the parameter Dist_Class_Nbr(see below). It is not compatible with a maximum sample size shorter than the number of distance class.
Q_{wi} : Id Prob for pairs for genes within individuals. It is not compatible with haploid simulations.
Q_{wd} : Id Prob for pairs of genes within nodes/demes (but not within individuals).
Q_{bd} : Id Prob for pairs of genes between nodes/demes. It is not compatible with single population simulations.

- Prob_Id_2Loc [Identity_Probability_2_Locus] tells GSpace to compute, for each simulated data file, the frequency of jointly identical pairs of gene copies taken at different loci, averaged over all pairs of gene copies at the two locus in the sample. The expected value of this statistic is a decreasing function of the rate of recombination. This setting has been implemented mainly to check the implementation of the recombination process by comparing the statistic to its analytical expectation (see section 3.1).

- Effective_Dispersal tells GSpace to compute the empirical effective backward dispersal distribution computed from all backward dispersal events that occurred during the multichromosome/locus simulation for each data set. Empirical dispersal distances are computed considering the habitat as a plane, even if the simulation settings actually considers a torus. Discrepancies between theoretical and empirical dispersal distributions are thus expected for all edge effects, especially for small size lattices and/or large maximal dispersal distances. Because of these discrepancies, the interpretation of the realized backward dispersal distribution given the settings specified for the forward distribution is sometimes difficult and may be troubling. Such mean empirical dispersal distribution has notably not much sense if the lattice is spatially heterogeneous (e.g. with heterogeneous density and/or heterogeneous dispersal). This empirical distribution for each simulated data set is then written in a file named Data_File_Name_Y_emp_disp.txt for data set Y. At the end of the file various statistics (mean, σ^2, kurtosis and skewness) are computed on the semi distribution (axial values) on each dimension. Finally, mean empirical dispersal distributions along the whole run (i.e. for all data sets) can be found at the end of the Data_File_Name_Global_Stats.txt file.

- Iterative [Iterative_Statistics] tells GSpace to compute, for Data_File_Name_Global_Stats.txt, the iterative mean and variance for the various statistics described in Post_Sim_Computations and report them. When this setting is not set, GSpace only reports average and variance values for a run.
- Per_Loc [Per_Loc_Statistics] tells GSpace to calculate the value of each locus for the various statistics describe in Post_Sim_Computations and to report them by chromosome in Data_File_Name_Y_Stats_per_loc_chr_X. Can be time consuming when the sequence is long and the chromosomes are numerous.
- Per_Chrom [Per_Chromosome_Statistics] tells GSpace to calculate the mean of each chromosome for the various statistics describe in Post_Sim_Computations and to report them by chromosome in Statistics_per_chr_simu_Y. Can be time consuming when the sequence is long and the chromosomes are numerous.

- Dist_Class_Nbr = 1 [or any integer value] tells GSpace to compute identity probabilities between individuals separated by a distance within the distance class r (Q_r), corresponding to all distances falling in $](r-1)*dist\_sample\_max/Dist\_Class\_Nbr; r{*}dist\_sample\_max/Dist\_Class\_Nbr]$ for $r > 0$ and *dist_sample_max* being the maximum distance between sampled individuals; or being at a null distance for $r = 0$.

4.3 Output files

GSpace can generate different types of output files depending on the settings chosen:

1. all simulated data sets in three different formats:

   (1) the extended input file format of Genepop v.4 (Rousset, 2008) with spatial coordinates of sampled individuals,

   (2) the format VCF (v 4.3) that consist in one files,

   (3) one specific file format that can be read as input for MIGRATE (Beerli & Felsenstein, 2001).

   (4)

2. a summary file named "Data_File_Name_GSpace_param_summary.txt" where most parameter values used for the simulation are summarized

and some statistics of the chosen dispersal distribution are computed (mean dispersal, second moment σ and kurtosis, etc...).

3. a summary statistic file named "Data_File_Name_Global_Stats" where the average per simulation of various genetic statistics, such as probability of identity between pairs of genes, are computed on the simulated data sets. The average and variance over simulation of these statistics are computed to. This file is presented as a table with the first line containing the names of each column (i.e. each statistic, usually straightforward to understand) followed by one line per simulated data set with the corresponding values. Note that it has specific values for each simulation.

4. a summary statistic file per simulation named "Statistics_per_chr_simu_Y" where the average per chromosome for simulation Y of various genetic statistics are computed on the simulated data sets. This file is presented as a table with the first line containing the names of each column (i.e. each statistic, usually straightforward to understand) followed by one line per chromosome data set with the corresponding values. Note that it has specific values for each chromosome.

5. a Chromosome_Number summary statistic file per simulation named "Statistics_per_loc_simu_Y_chr_X" where the average per locus for chromosome X simulation Y of various genetic statistics are computed on the simulated data sets. This file is presented as a table with the first line containing the names of each column (i.e. each statistic, usually straightforward to understand) followed by one line per locus data set with the corresponding values. Note that it has specific values for each locus.

4.4 Interaction with Genepop

Interaction of GSpace with Genepop to evaluate the performance of inferences under isolation by distance has been greatly enhanced in the latest version of Genepop (V. 4 and later). Genepop's behavior can now be controlled using a setting file and by inline arguments in a console command line.

This allows batch calls to Genepop and repetitive use of Genepop on simulated data. Such automatic batch mode of Genepop makes it easy for anyone to test the performance of the regression estimators of $D\sigma^2$ by the regression methods (Rousset, 1997; Rousset, 2000; see the Genepop documentation section 5 for details),including the performance of the bootstrap confidence intervals, using simulated data sets produced by GSpace.

5 Credits and Copyright (code, grants, etc.)

This work was financially supported by the recurrent funding from the INRAe to RL, and from the CNRS to FR, and by the Agence Nationale de la Recherche (ANR projects GENOSPACE ANR-16-CE02-0008, INTROSPEC ANR-19-CE02-0011 and DISLAND ANR-20-CE32-XXX). This work also benefited from access to the national INRAe MIGALE http://migale.jouy.inrae.fr and GENOTOUL (Toulouse Midi-Pyrénées) bioinformatics HPC platforms, as well as the CBGP and the local Montpellier Bioinformatics Biodiversity (MBB, supported by the LabEx CeMEB ANR-10-LABX-04-01) HPC platform services for providing storage and computing resources.

`GSpace` uses a re-implementation of the Hudson's modified algorithm implemented in `msprime` and described in Kelleher *et al.* (2016).

Bibliography

Beerli, P. & Felsenstein, J., 2001. Maximum likelihood estimation of a migration matrix and effective population sizes in n subpopulations by using a coalescent approach. *Proc. Natl. Acad. Sci. U. S. A.* **98**: 4563–4568.

Chesson, P. & Lee, C. T., 2005. Families of discrete kernels for modeling dispersal. *Theor. Popul. Biol.* **67**: 241–256.

Crow, J. F. & Kimura, M., 1970. *An introduction to population genetics theory.* Harper & Row, New York.

Endler, J. A., 1977. *Geographical variation, speciation, and clines.* Princeton University Press, Princeton.

Felsenstein, J., 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* **17**: 368–376.

Hasegawa, Kishino & Yano, 1985. Dating of human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.* **22**: 160–174.

Hudson, R. R., 1983. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology* **23**: 183–201.

Hudson, R. R., 1993. The how and why of generating gene genealogies. In: *Mechanisms of molecular evolution* (N. Takahata & A. G. Clark, eds.), pp. 23–36. Sinauer, Sunderland, MA.

Jukes, T. & Cantor, C., 1969. Evolution of Protein Molecules. In: *Mammalian Protein Metabolism* (M. Munro, ed.), pp. 21–132. New York. Academic Press.

Kelleher, J., Etheridge, A. M. & McVean, G., 2016. Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes. *PLOS Computational Biology* **12**: e1004842.

Kimura, M., 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotides sequences. *J. Mol. Evol.* **16**: 111–120.

Kimura, M. & Crow, J. F., 1964. The number of alleles that can be maintained in a finite population. *Genetics* **49**: 725–738.

Kingman, J. F. C., 1982. The coalescent. *Stoch. Processes Applic.* **13**: 235–248.

Kot, M., Lewis, M. A. & van den Driessche, P., 1996. Dispersal data and the spread of invading organisms. *Ecology* **77**: 2027–2042.

Leblois, R., Estoup, A. & Rousset, F., 2003. Influence of mutational and sampling factors on the estimation of demographic parameters in a “continuous” population under isolation by distance. *Mol. Biol. Evol.* **20**: 491–502.

Leblois, R., Estoup, A. & Rousset, F., 2009. IBDSim: A computer program to simulate genotypic data under isolation by distance. *MolecolRes* **9**: 107–109.

Leblois, R., Estoup, A. & Streiff, R., 2006. Habitat contraction and reduction in population size: Does isolation by distance matter? *Mol. Ecol.* **15**: 3601–3615.

Leblois, R., Rousset, F. & Estoup, A., 2004. Influence of spatial and temporal heterogeneities on the estimation of demographic parameters in a continuous population using individual microsatellite data. *Genetics* **166**: 1081–1092.

Malécot, G., 1975. Heterozygosity and relationship in regularly subdivided populations. *Theor. Popul. Biol.* **8**: 212–241.

Nordborg, M., 2007. Coalescent theory. In: *Handbook of statistical genetics* (D. J. Balding, M. Bishop & C. Cannings, eds.), pp. 843–877. Wiley, Chichester, U.K., 3rd edn.

Ohta, T. & Kimura, M., 1973. A model of mutation appropriate to estimate the number of electrophoretically detectable alleles in a finite population. *Genet. Res.* **22**: 201–204.

Patil, G. P. & Joshi, S. W., 1968. *A dictionary and bibliography of discrete distributions*. Oliver & Boyd, Edinburgh.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A. & Feldman, M. W., 1999. Population growth of human Y chromosome microsatellites. *Mol. Biol. Evol.* **16**: 1791–1798.

Robledo-Arnuncio, J. J. & Rousset, F., 2010. Isolation by distance in a continuous population under stochastic demographic fluctuations. *J. Evol. Biol.* **23**: 53–71.

Rousset, F., 1996. Equilibrium values of measures of population subdivision for stepwise mutation processes. *Genetics* **142**: 1357–1362.

Rousset, F., 1997. Genetic Differentiation and Estimation of Gene Flow from FStatisticsUnder Isolation by Distance. *Genetics* **145**: 1219–1228.

Rousset, F., 2000. Genetic differentiation between individuals. *J. Evol. Biol.* **13**: 58–62.

Rousset, F., 2008. GENEPOP007: a complete re-implementation of the GENEPOP software for Windows and Linux. *Molecular Ecology Resources* **8**: 103–106.

Tamura, K. & Nei, M., 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol. Biol. Evol.* **10**: 512–526.

Vitalis, R. & Couvet, D., 2001. Estimation of Effective Population Size and Migration Rate From One- and Two-Locus Identity Measures. *Genetics* **157**: 911.

Watts, P. C., Rousset, F., Saccheri, I. J., Leblois, R., Kemp, S. J. & Thompson, D. J., 2007. Compatible genetic and ecological estimates of dispersal rates in insect (*Coenagrion mercuriale*: Odonata: Zygoptera) populations: analysis of 'neighbourhood size' using a more precise estimator. *Mol. Ecol.* **16**: 737–751.

Index