IBDSim version 1.4

User manual

November 7, 2011

IBDSim is a computer package for the simulation of genotypic data at multiple unliked loci under general isolation by distance models. It is based on a backward "generation by generation" coalescent algorithm allowing the consideration of various isolation by distance models on a lattice with deterministically varying deme size, migration rates and mutation rates. IBDSim can hence consider a large panel of subdivided population models representing discrete subpopulations as well as a large continuous population. Many dispersal distributions can be considered as well as heterogeneities in space and time of the demographic parameters. Typical applications of our program include the study of the effect of various sampling, mutational and demographic factors on the pattern of genetic variation at different spatial scales and the production of test data sets to assess the influence of these factors on any inferential method available to analyze genotypic data for independent loci. The program runs on MacOs X and PC under Windows or Linux systems, but we also provide the source code that can be compiled under any system using C++ ISO compiler. It is freely available on the website at http://kimura.univ-montp2.fr/~rousset/IBDSim.html.

1 Requirements

1.1 Executables and source compilation for various OS

The program IBDSim is available for download on the web at the address http://kimura.univ-montp2.fr/~rousset/IBDSim.html and is provided as Windows executable as well as original source code.
Windows users can run the provided executables (IBDSim.exe, build with Code::Blocks). Note that for easier distribution, we renamed the IBDSim.exe file to IBDSim.zut, so that windows users have to do the opposite operation, i.e. rename the IBDSim.zut file to IBDSim.exe.
Linux and Mac users should easily recompile the sources using gcc and the following command line

```
g++ -DNO_MODULES -o IBD_Sim lattice_sampling.cpp -O3
```

Various versions of the code have been compiled and run on PCs under Windows and Linux. Some preprocessor instructions were added to compile the code on these different systems. So this should essentially work on most Unix-based systems, including Mac OS X and previews versions (I have only tested it on MacOs X). When compiling with specific compilers (i.e. others than GCC) or specific IDE like wxDevC++ , one should sometimes manually edit/add/delete some "#include <library>" instructions in the first lines of the file "lattice_sampling.cpp".

1.2 Hardware

IBDSim should run on any reasonably recent computer and has limited memory needs for most reasonable settings. There is virtually no limitation or maxima for the parameter values concerning the lattice and subpopulation sizes, the sample size (i.e. number of individuals and loci), 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 hours for 1000 data sets of 20 loci for 1000 individuals evolving on a 100x100 lattice with subpopulation sizes of 40 individuals). Note that considering heterogeneities in space will strongly increase computation times as well as memory usage, especially for large lattice sizes.

2 Principle of the simulation algorithm

The `IBDSim` program is based on the backward in time coalescent approach, which is well known to allow the development of efficient simulation tools (Hudson, 1993). Such an approach allows the generation of large genotypic data sets considering complex migration schemes including those with heterogeneities in space and time of the demographic parameters. Moreover, because our program allows various deme size and migration rates, it can simulate genotypic data under both a model of subdivided population with discrete subpopulations and a model corresponding to a large continuous population with any intermediate situation.

For neutral genes, the coalescent process depends solely on the demographic history of the population and is independent from the mutational processes. So we first generate the genealogy of the sampled genes going backward in time and then we 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.

Because of the complexity of the IBD models considered, the coalescent algorithm used to build the genealogical tree is not based on the large-N approximation of the n-coalescent theory (Kingman, 1982; Nordborg, 2007). It is rather an exact algorithm for which coalescence and migration events are considered generation by generation up to the common ancestor of the 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. 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 we summarized the main ideas underlying the global algorithm in Leblois *et al.* (2009). The algorithm and the program used in this study were checked at every step during its elaboration 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) with adaptation to different mutation models following Rousset (1996)).

3 Using IBD-Sim

3.1 Input file format

IBDSim reads one generic text file (ASCII) named by default "IbdSettings.txt" that must be in the same folder as the application. The file is read at the beginning of each execution and allows one to control all options of IBDSim. It contains lines of the form keyword=value(s), where value(s) can take various formats as described below. All setting options are explained in details in the next subsections. The default name of the settings file is IbdSettings.txt but you can change this through the command line: running IBDSim using 'IBD_Sim(.exe) SettingsFile=mysettings.txt' will make the program read mysettings.txt rather than ibdsettings.txt. Here is an example of a complete setting file :

```
%%%%%SIMULATION PARAMETERS%%%%%%%%%%%%
Data_File_Name=TestPapier
.txt_extension=true
Run_Number=10
Random_Seeds=87144630

%%%%%MARKERS PARAMETERS%%%%%%%%%%%%%%%
Locus_Number=5
Min_Allele_Number=2
Max_Allele_Number=200
Mutation_Rate=0.05
Variable_Mutation_Rate=true
Mutation_Model=GSM
Allelic_Lower_Bound=1
Allelic_Upper_Bound=100
Allelic_State_MRCA=0
SMM_Probability_In_TPM=0.8
Geometric_Variance_In_TPM=10
Geometric_Variance_In_GSM=0.36
Ploidy=Diploid

%%%%%%OUTPUT FILE FORMAT OPTIONS%%%%%%%
Genepop=true
Migraine=false
Migraine_AllStates=false
Migrate=false
```

```
Migrate_Lettre=false

%%%%%%VARIOUS COMPUTATION OPTIONS%%%%%%%
Generic_Computations=true
Hexp_Nei=true
DeltaH=true
Allelic_Variance=true
Iterative_Identity_Probability=true
Iterative_Statistics=true
Prob_Id_Matrix=true
Effective_Dispersal=true
Constant_Dispersal=true
Total_Range_Dispersal=true

%%%%%%%%DEMOGRAPHIC OPTIONS%%%%%%%%%%%%%

  %%NOT TIME DEPENDANT PARAMETERS%%
  Lattice_Boundaries=absorbing
  Max_Lattice_SizeX=300
  Max_Lattice_SizeY=300
  Sample_SizeX=10
  Sample_SizeY=10
  Min_Sample_CoordinateX=5
  Min_Sample_CoordinateY=5

  Specific_Sample_Design=false
  SpecificSampleDesign_SampleSize=10
  Sample_Coordinates_X=3 4 8 9 10 11 12 13 17 18
  Sample_Coordinates_Y=1 6 9 12 15 16 17 18 19 20

  Ind_Per_Pop_Sampled=1
  Void_Sample_Node=1

  Min_Zone_CoordinateX=1
  Max_Zone_CoordinateX=1
  Min_Zone_CoordinateY=1
  Max_Zone_CoordinateY=1

  %%TIME DEPENDANT PARAMETERS%%
  %%From G=0 to G=GN1%%
  Ind_Per_Pop0=10
```

```
Lattice_SizeX0=20
Lattice_SizeY0=20

Void_Nodes0=1
Zone0=false
Void_Nodes_Zone0=1
Ind_Per_Pop_Zone0=1

Specific_Density_Design=false

Dispersal_Distribution0=9
Total_Emigration_Rate0=0.1
Disp_max0=48
Pareto_Shape0=2.16574
Geometric_Shape0=0.75
Sichel_Gamma0=-2.15
Sichel_Xi0=20.72
Sichel_Omega0=-1

ContinuousDemeSizeVariation0=Exponential

%%From G=Gn1 to G=Gn2%%
%%(for constant model in time set Gn1=Gn2=Gn3=2147483647)%%
Gn1=2147483647
Ind_Per_Pop1=1
Lattice_SizeX1=10
Lattice_SizeY1=10

Random_Translation=true
Void_Nodes1=1
Zone1=false
Void_Nodes_Zone1=1
Ind_Per_Pop_Zone1=1

Dispersal_Distribution1=9
Total_Emigration_Rate1=0.1
Disp_max1=48
Pareto_Shape1=2.16574
Geometric_Shape1=0.75
Sichel_Gamma1=-2.15
Sichel_Xi1=20.72
```

```
Sichel_Omega1=-1

ContinuousDemeSizeVariation1=Exponential

 %% From G=Gn2 to G=Gn3
Gn2=2147483647
Ind_Per_Pop2=1
Lattice_SizeX2=10
Lattice_SizeY2=10

Random_Translation=true
Void_Nodes2=1
Zone2=false
Void_Nodes_Zone2=1
Ind_Per_Pop_Zone2=1
Dispersal_Distribution2=9
Total_Emigration_Rate2=0.1
Disp_max2=48
Pareto_Shape2=2.16574
Geometric_Shape2=0.75
Sichel_Gamma2=-2.15
Sichel_Xi2=20.72
Sichel_Omega2=-1

ContinuousDemeSizeVariation2=Exponential

%%From G=Gn3 to G=infinity
Gn3=2147483647
Ind_Per_Pop3=1
Lattice_SizeX3=10
Lattice_SizeY3=10

Random_Translation=true
Void_Nodes3=1
Zone3=false
Void_Nodes_Zone3=1
Ind_Per_Pop_Zone3=1
Dispersal_Distribution3=9
Total_Emigration_Rate3=0.1
Disp_max3=48
Pareto_Shape3=2.16574
```

```
  Geometric_Shape3=0.75
  Sichel_Gamma3=-2.15
  Sichel_Xi3=20.72
  Sichel_Omega3=-1
%%%%%EndOfSettings%%%%%%%
```

3.1.1 Simulation parameters

All options in this category are quite straightforward to understand :

Data_File_Name=Test tells IBDSim the generic file name for the simulated data sets. This generic file name will be incremented with the number of the run. Example : simulated data file number 56 will be named here 'Test56'.

.txt_extension=true OR false tells IBDSim to add or not to add a '.txt' extension to each simulated file Example : if set to true, simulated data file number 56 will be named here 'Test56.txt'.

Run_Number=1000 tells IBDSim to run a given number of iterations, i.e. a given number of simulated data sets, here 1000.

Random_seeds=568974526 are the seed for the random number generator. Different runs with precisely the same parameter values and same seeds will give exactly the same results.

3.1.2 Genetic marker parameters

All options in this category concern the genetic markers parametrization and are also, for most of them, straightforward to understand.

Locus_Number=10 is the number of loci to simulate per data set.

Min_Allele_Number=2 sets the minimum number of alleles that a locus should have to be incorporated into the simulated data sets. That means that specifying a value of 2 here will tell IBDSim to consider only polymorphic loci for the simulated data set. If Min_Allele_Number is larger than one, IBDSim will keep simulating new loci until he found Locus_Number loci with a minimum of Min_Allele_Number alleles for each data set.

Max_Allele_Number=200 sets the maximum number of alleles that a locus should have to be incorporated into the simulated data sets. It works exactly as the last option but will be usually less useful as it has been implemented to limit the number of alleles at each locus when computing the ΔH statistic of Cornuet & Luikart (1996). The reason is that IBDSim has only in memory the expected heterozygosity values for a number of alleles limited to 200 (see option `DeltaH` p.14).

`Mutation_Rate=0.0005` is the mutation rate of all simulated loci, specified by locus and by generation.

`Variable_Mutation_Rate=true OR false` tells `IBDSim` to simulate a constant or a variable mutation rate among loci. If a variable mutation rate is chosen, `IBDSim` will automatically draw random mutation rates for each locus in a Gamma distribution with parameters (shape and scale) being (2, `Mutation_Rate`/2) so that the mean mutation rate across loci will be equal to the specified value for `Mutation_Rate`.

`Mutation_Model=IAM or KAM or SMM or TPM or GSM` sets the mutation model for all loci. Five theoretical mutation models are implemented in `IBDSim` :
(i) the infinite allele model (IAM, Kimura & Crow, 1964) in which each mutation give rise to a new allele;
(ii) 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. The number of possible allelic states K is then given by the options
`Allelic_Lower_Bound` and `Allelic_Upper_Bound` with
K =`Allelic_Upper_Bound - Allelic_Lower_Bound` +1;
(iii) the strict stepwise mutation model (SMM, Ohta & Kimura, 1973), especially designed for microsatellite markers, where each mutation adds or removes a repeated unit to the mutated allele;
(iv) the two phase model (TPM, Di Rienzo *et al.*, 1994), where each mutation adds or removes X repeated units to the mutated allele. With a probability `SMM_Probability_In_TPM`, X is equal to 1 and with a probability of (1-`SMM_Probability_In_TPM`) X is randomly chosen from a geometric distribution with a variance of `Geometric_Variance_In_TPM`, implying a gain or a loss of more than one repeated unit.
and (v) the generalized stepwise model (GSM, e.g. Pritchard *et al.*, 1999), similar to the TPM but where there is only one phase of geometric loss or gain of X repeated units (geometric distribution with variance equals to `Geometric_Variance_In_GSM`).

`Allelic_Lower_Bound=1` sets the lowest possible allelic state for the mutation model considered.

`Allelic_Upper_Bound=36` sets the largest possible allelic state for the mutation model considered. Note that using those bounds can be used with all mutation models except the `IAM`. see also the `KAM Mutation_Model` for its use with the `KAM` model.

`SMM_Probability_In_TPM=0.8` see the `Mutation_Model=TPM` option.

`Geometric_Variance_In_TPM=10` see the `Mutation_Model=TPM` option.

`Geometric_Variance_In_GSM=0.36` see the `Mutation_Model=GSM` option.

`Ploidy=Diploid or Haploid` set the ploidy level of the marker used. Note that while our model assumes that individuals are haploids and that dispersal occurs through gametes only, diploid data are simulated by considering Hardy-Weinberg equilibrium within each lattice node at sampling time.

3.1.3 Data set output options

These options set the different data file format to be generated for each data set simulated by `IBDSim`. Those data file can be then analyzed by other programs such as `Genepop`, `Migraine` or any others than can read one of the three following formats.

`Genepop=true OR false` tells `IBDSim` to write or not each data file in the classical and widely used Genepop format (actually the extended input file format of `Genepop` v.4; Rousset, 2008). 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") are separated by `pop` statements. For each "population" the values before the coma of the last individual indicates geographic coordinates of the "populations" localization. This is a widely used format, and both convenient information (names of samples) and information relevant to the analyzes (spatial coordinates of samples) can be included. See the `Genepop` documentation for details and examples.

`Migraine=true OR false` tells `IBDSim` to write or not each data file in the ad hoc MIGRAINE file format; Rousset & Leblois (2007). One is advised to use the Genepop input format, as the ad hoc format may become obsolete in some later version of `Migraine`. In its simplest form, the ad hoc format consists of one file per locus, containing a population (row) by allele (column) table of allelic counts, where each row is terminated by a semicolon. For example, data at the first locus may be

```
0 0 0;
5 4 3;
0 0 0;
3 6 77;
10 10 20;
0 0 0;
0 0 0;
0 0 0;
0 0 0;
```

This input means that the data will be analyzed according to a 9-populations model (the number of rows), only three of which have been sampled (row 2, 4 and 5, which are also the relative positions of the samples in the array of populations). The columns are allelic counts for each allelic type. There is one such file for each locus. See the `Migraine` documentation for details and examples.

`Migraine_AllStates=true OR false` tells `IBDSim` to write empty allelic classes or not in the simulated data files. This option is specific to the ad hoc MIGRAINE file format. In MIGRAINE, this option, implemented for ad hoc input file format only, controls the number of alleles assumed in the KAM. If this option is set to `TRUE` `Migraine` will consider a K-AM model with a K being the total number of columns in the input file (i.e. allelic states present or not in the sample). For example, in order to use a $K = 6$-AM model even if only 3 alleles have been simulated, this setting can be used to generate the following 6-column output file:

```
0 0 0 0 0 0;
5 0 0 4 0 3;
0 0 0 0 0 0;
3 0 0 6 0 77;
10 0 0 10 0 20;
0 0 0 0 0 0;
0 0 0 0 0 0;
0 0 0 0 0 0;
0 0 0 0 0 0;
```

Its default mode is false and gives the following 3 column output file :

```
0 0 0;
5 4 3;
0 0 0;
3 6 77;
```

```
 10 10 20;
 0 0 0;
 0 0 0;
 0 0 0;
 0 0 0;
```

`Migrate=true OR false` tells `IBDSim` to write or not each data file in the MIGRATE file format Beerli & Felsenstein (2001). The generic input file format for MIGRATE is the following, where < token > in "angle-brackets" is obligatory, [token] in square brackets are optional, {token} in parenthesis is obligatory for some:

```
<Nb of pops> <nb of loci> {delimiter between alleles}
<Nber of individuals> <title for pop 0-79>
<Individual 1 10-10> <data>
<Individual 2 10-10> <data>
....
<Nber of individuals> <title for pop 0-79>
<Individual 1 10-10> <data>
<Individual 2 10-10> <data>
....
```

Here is an example of Migrate Input data file with microsatellite loci :

```
2 3 . Rana lessonae: Seeruecken versus Tal
2 Riedtli near G\"undelhart-H\"orhausen
0 42.45 37.31 18.18
0 42.45 37.33 18.16
4 Tal near Steckborn
1 43.46 33.37 18.18
1 44.46 33.35 19.18
1 44.46 35.? 18.18
1 43.42 35.31 20.18
```

`Migrate_Letter=true OR false` tells `IBDSim` to write or not each data file in the special MIGRATE file format with letters representing alleles Beerli & Felsenstein (2001). Note that this option is yet limited to a maximum number of alleles of 36. Here is an example of such format with allelic states represented as letters (e.g for allozymes) :

```
2 11 Migration rates between two Turkish frog populations
3 Akcapinar (between Marmaris and Adana)
PB1058 ee bb ab bb bb aa aa bb ?? cc aa
```

```
PB1059 ee bb ab bb bb aa aa bb bb cc aa
PB1060 ee bb b? bb ab aa aa bb bb cc aa
2 Ezine (between Selcuk and Dardanelles)
PB16843 ee bb ab bb aa aa aa cc bb cc aa
PB16844 ee bb bb bb ab aa aa cc bb cc aa
```

3.1.4 Various computational options

For more details on all possible output files generated by IBDSim, go to section 3.4.

Generic_Computations=true OR false tells IBDSim to compute or not probabilities of identity and coalescence times between pairs of genes at different levels (intra-individuals, intra-populations, inter-populations). It increases computation times by 10% but is needed for some of the following calculations. Be carrefull it is not compatible with a specific sample design (see Specific_Sample_Design option).

Hexp_Nei=true OR false tells IBDSim to compute or not the expected Nei's heterozygosity and to report it in the output files 'various_statistics.txt' and 'Iterative_Statistics.txt'. Be carrefull it is not compatible with a specific sample design (see Specific_Sample_Design option).

DeltaH=true OR false tells IBDSim to compute or not the ΔH statistic (described in Cornuet & Luikart, 1996) and to report it in the output files 'various_statistics.txt' and 'Iterative_Statistics.txt'. It is yet implemented only for the GSM mutation model and for sample sizes of 60 and 200 genes. The ΔH statistic is the deficit or excess of the expected heterozygosity given the number of alleles in the sample compared to its value in an equilibrium Wright-Fisher population and is especially designed to test for past bottlenecks or expansions. Be carrefull it is not compatible with a specific sample design (see Specific_Sample_Design option).

Allelic_Variance=true OR false tells IBDSim to compute or not the variance in allelic size as well as the $M-$statistic of Garza & Williamson (2001) and to report them in the output files 'various_statistics.txt' and 'Iterative_Statistics.txt'. Those two statistics are designed for microsatellite markers only.

Iterative_Identity_Probability=true OR false tells IBDSim to compute, for each simulated data file, identity probabilities for the pairs of sampled genes and to report them in the output file 'Iterative_IdProb.txt'. This option is essentially implemented to plot the identity probabilities trough the run to check the program against analytical expectations. Be carrefull it is not compatible with a specific sample design (see `Specific_Sample_Design` option).

`Iterative_Statistics=true OR false` tells `IBDSim` to compute, for each simulated data file and for each locus, various statistics such as F_{IS}, heterozygosity and all other statistics described in the previous options and to report them in the output file `'Iterative_Statistics.txt'` (see section 3.4 for details about the `'Iterative_Statistics.txt'` file format).
`Generic_Computations=true` is necessary for this option to work.

`Allelic_Number_Folder=true OR false` tells `IBDSim` to group simulated data sets, but only with the ad hoc `Migraine` file format, in specific folders according to the number of alleles in the data set.

`Prob_Id_Matrix=true OR false` tells `IBDSim` to write an output file named `'Matrix_IdProb.txt'` with the matrix of identity probabilities as a function of the place of the sampled genes on the lattice. Be carrefull it is not compatible with a specific sample design (see `Specific_Sample_Design` option).

`Effective_Dispersal=true OR false` tells `IBDSim` to compute or not the empirical effective dispersal distribution computed from all dispersal events that occurred during the multilocus coalescent trees used for each simulation of one data set. Empirical dispersal distances are computed considering the habitat as a plane, even if the simulation settings actually considers a torus. Descrepancies between theoretical and empirical dispersal distributions are thus expected when working on a torus, especially for small size lattices and/or large maximal dispersal distances. This empirical distribution is then written in a file named `EmpDisp_CurrentDataFileName`. At the end of the file various statistics (mean, σ^2, kurtosis and skewness) are computed on the whole distribution and on the semi distribution (axial values).

`Constant_Dispersal=true OR false` tells `IBDSim` if there is any change in dispersal through time in the simulation settings. `IBDSim` will run faster with this option turned on.

`Total_Range_Dispersal=true OR false` tells `IBDSim` to keep maximum dispersal distances from the specific settings of the chosen distribution (or by the option `Disp_Max` p.19) rather than constrained by lattice size (i.e. individuals can disperse up to *dx_max* steps, where *dx_max* is set according to each dispersal distribution settings rather than limited by lattice size, see section 3.2 for more details on dispersal distributions).

3.1.5 Time independent demographic parameters

In this section all settings for the demographic part of the model that are independent of time are specified. Those time independent parameters correspond to demographic settings kept at fixed values during the whole simulation. They have to be compatible with the time dependent demographic options detailed in the next section.

`Lattice_Boundaries=Circular OR Absorbing OR Reflecting` set the habitat (i.e. lattice) boundaries type (also called edge effects) to be considered for the entire simulation. Habitat boundaries can not be changed through time. See section 3.3 for details on the different possible habitat boundaries implemented in `IBDSim`.

`Max_Lattice_SizeX=300` is the maximum lattice size in the first dimension X. All other lattice size settings have to be less or equal to this value. So be careful to set correct lattice size in the `Time dependent options` below.

`Max_Lattice_SizeY=100` is the maximum lattice size in the second dimension Y. All other lattice size settings have to be less or equal to this value. So be careful to set correct lattice size in the `Time dependent options` below.

`Sample_SizeX=10` is the axial number of sampled nodes in dimension X.

`Sample_SizeY=15` is the axial number of sampled nodes in dimension Y.

`Min_Sample_CoordinateX=145` is the coordinate of the most left sampled node in dimension X.

`Min_Sample_CoordinateY=145` is the coordinate of the most left sampled node in dimension Y.

`Specific_Sample_Design=true OR false` tells `IBDSim` to consider (1) a general square sample of size `Sample_SizeX x Sample_SizeY` located on the lattice at coordinates (`Min_Sample_CoordinateX`,`Min_Sample_CoordinateY`) if set to false; Or (2) a user specific sample configuration where each of the `SpecificSampleDesign_SampleSize` sampled node have coordinates given by the next options `Sample_Coordinates_X` and `Sample_Coordinates_Y`, if set to true. In such case, the number of individuals sampled at each sampled node is still `Ind_Per_Pop_Sampled`.

`SpecificSampleDesign_SampleSize=XX` is the number of sampled nodes/demes using the `Specific_Sample_Design=true` option.

`Sample_Coordinates_X=20 56 78 98 101 102 121 134 156 199` is a list of dimension `SpecificSampleDesign_SampleSize` with specific X coordinates for the
`Specific_Sample_Design=true`.

`Sample_Coordinates_Y=4 8 15 17 20 26 34 50 56 65 72 78 82 88 98` is a list of dimension `SpecificSampleDesign_SampleSize` with specific Y coordinates for the option `Specific_Sample_Design=true`.

`Ind_Per_Pop_Sampled=4` is the number of individuals sampled on each lattice node (i.e. "subpopulation" or individual for the continuous population model) on the sampled area.

`Void_Sample_Node=1` is a tricky setting to not sample every node on the previously designed sampling area. With a value of 1 `IBDSim` will sample all node on the sampling area, with a value of 2 `IBDSim` will only sample one node over two, etc...

`Min_Zone_CoordinateX=20` is the lowest coordinate (left border) in dimension X of the "zone" (i.e. a portion of the lattice where demographic parameters are different from the rest of the lattice, see option `Zone0` p. 18).

`Max_Zone_CoordinateX=40` is the highest coordinate (right border) in dimension X of the "zone".

`Min_Zone_CoordinateY=20` is the lowest coordinate (bottom border) in dimension Y of the "zone".

`Max_Zone_CoordinateY=40` is the highest coordinate (top border) in dimension Y of the "zone".

3.1.6 Time dependent demographic parameters

This section concerns demographic parametrization that can change through time. All settings in this section are repeated three times with index 0,1,2,3 for the present time setting and for each potential past demographic change respectively. `IBDSim` will consider the demographic settings with index 0 from Gn=0 to Gn1, with index 1 from Gn1 to Gn2, with index 2 from Gn2 to Gn3, and with index 3 from Gn3 to infinity. I will give only one description of those settings for the present time configuration (i.e; with the index 0) as they are exactly the same for all demographic change (except the generation at which the change occurs that must be set by the options `GnX=200` where X is the number of the demographic change, i.e. $1, 2$ or 3). By setting Gn1, Gn2 and Gn3 to the highest integer value (i.e. 2 147 483 647), the model will be constant through time.

`Ind_Per_Pop0=10` is number of individual per lattice node that `IBDSim` will consider. It also correspond to the density in number of individuals per lattice node.

`Lattice_SizeX0=100` is the lattice size in the first dimension X. Must be less or equal to the previous time independent option `Max_Lattice_SizeX`.

`Lattice_SizeY0=100` is the lattice size in the second dimension Y. Must be less or equal to the previous time independent option `Max_Lattice_SizeY`.

`Random_Translation=true OR false` tells `IBDSim` where, after a change in time of the lattice size, to place the smaller lattice on the larger one. If true it will be randomly placed on the larger surface, if false it is placed on the most left bottom corner of the larger lattice.

`Void_Nodes0=10` is a tricky setting to consider that a given proportion of lattice nodes are empty (i.e. they do not carry any individuals of the population). It has been implemented to decrease density without changing dispersal functions in Leblois *et al.* (2004). It can generaly be used to consider low densities (e.g less than one individual per lattice node) without changes in total lattice surface and dispersal distributions. With a value of 1, `IBDSim` will consider that all lattice node have individuals on them. With a value of 2, `IBDSim` will consider that one node over two is empty and can not receive any individual during simulation.

`Zone0=true OR false` tells `IBDSim` if there is heterogeneities in space in the density/subpopulation sizes by considering a special demographic “zone” (i.e. a portion of the lattice where demographic parameters are different from the rest of the lattice).

`Void_Nodes_Zone0=2` is the equivalent of the option `Void_Nodes0=10` but for the specific demographic “zone”.

`Ind_Per_Pop_Zone0=5` is number of individual per lattice node on the special demographic “zone” if there is one. In other words, it is the density in number of individuals per lattice node on the special demographic “zone”.

`Specific_Density_Design=true OR false` tells `IBDSim` 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 have a number of individuals (i.e. deme size) specified in a file named DensityMatrix.txt, if set to true. The format of DensityMatrix.txt is a matrix with X coordinates in columns and Y coordinates in rows. The file begin with coordinate X=0 and Y=0 in the upper left corner, X=LatticeSizeX and Y=0 in the lower left corner, X=0 and Y= LatticeSizeY in the upper right corner, and X= LatticeSizeX and Y= LatticeSizeY in the lower right corner, so that the density matrix specified in DensityMatrix.txt is a “transposed” image of the lattice. With `Specific_Density_Design=true` , it is better to use

`Specific_Sample_Design=true` with sampled nodes corresponding to lattice nodes where density is greater than 0, to avoid bad behavior of the program.

`Dispersal_Distribution0=9` Its argument is a character, either a letter or a number, referring to one of the implemented dispersal distributions. This option tells `IBDSim` to consider one of the preset dispersal distribution on the time interval considered. Detailed description of all implemented dispersal distribution is given in the next section (3.2.2).

`Total_Emigration_Rate0=0.1` is the total emigration rate (i.. probability to disperse) for the stepping stone model (case "b"), the general truncated Pareto distribution (case "P") and the geometric distribution (case : "g"). It corresponds to the terms mig or M in the next option descriptions.

`Disp_Max0=48` is the maximum distance moved at each generation, or the bound of the dispersal distribution, in lattice steps, for the custom Pareto (case "P") and the geometric distribution (case : "g").

`Pareto_Shape0=0.1` is the shape parameter value of the custom truncated Pareto distribution (case : "P"). For more details on this distribution, see the description of truncated Pareto distribution on p.21. It corresponds to the term n in the formula $Prob(dist = k) = M/k^n$.

`Geometric_Shape0=0.1` is shape parameter value of the geometric distribution (case : "g"). It corresponds to the term g in the formula $Prob(dist = k) = mig/2 * g^{k-1} * (g - 1)$ (see p.21).

`Sichel_Gamma0=-2.15` is the first parameter of the Sichel distribution (case : "S"), it must be negative. See the complete Sichel distribution description p.21 for more details.

`Sichel_Xi0=20.72` is the second parameter of the sichel distribution (case : "S").

`Sichel_Omega0=-1` is the third parameter of the sichel distribution (case : "S").

`ContinuousDemeSizeVariationX=Linear,Exponential OR false` tells `IBDSim` to consider (1) time constant density on the lattice if set to false; Or (2) a linear or exponential continuous change in density bewteen GnX and GnX+1. By a continuous change in density we mean a continuous change in deme size, i.e. the number of individual in each lattice node.

3.2 Details on dispersal distributions

3.2.1 Different types of distributions

We used the "backward" dispersal distribution in the coalescent algorithm because the position of the parental gene is determined knowing the position of its descendant gene (remember that dispersal is gametic and thus involves haploide entities). This "backward" function is computed using $f_{dx,dy}$, the forward dispersal density function describing where descendants go. To do so, we assume first that dispersal is independent in each direction, so that $f_{dx,dy} = f_{dx} \cdot f_{dy}$. In the simplest case, considering that density is homogeneous in space, backward dispersal functions are equal to forward dispersal functions, so that $b_{dx,dy} = f_{dx,dy} = f_{dx} \cdot f_{dy}$. 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. Let $N_{x,y}$ be the number of individuals at node (x, y) and K_{max} the maximum distance of dispersal. Then for any node (x, y) the probability $b_{dx,dy}$ for a gene to move backward dx steps in one direction and dy in the other is equal to :

$$b_{dx,dy} = \frac{N_{(x+dx),(y+dy)} \cdot f_{dx,dy}}{\sum_{dx,dy \leq K_{max}} N_{(x+dx),(y+dy)} \cdot f_{dx,dy}} \tag{1}$$

With regards to forward dispersal distributions, it is worth pointing that biologically realistic dispersal functions often have a high kurtosis (Endler, 1977; Kot *et al.*, 1996). As previously explained (Rousset, 2000), the commonly used discrete probability distributions for dispersal are not the most appropriate ones for isolation by distance because high kurtosis can be achieved only by assuming a low dispersal probability, i.e. that most offspring reproduce exactly where their parents reproduced. Therefore we used two different families of forward dispersal distributions for which suitable choice of their parameter values allows high kurtosis and high migration rates. The first 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

$0 < k \leq K_{max}$) in one direction being of the form:

$$f_k = f_{-k} = \frac{M}{k^n} \tag{2}$$

with parameters M and n, controlling the total dispersal rate and the kurtosis, respectively. 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 $\omega \to 0$, $\xi \to \inf$ with $\omega\xi \to \kappa$ 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).

For convenience, we also considered geometric dispersal distributions for which the probability of moving k steps (for $0 < k < K_{max}$) in one direction is :

$$f_k = f_{-k} = \frac{m}{2}(1-g)g^{(k-1)}, \tag{3}$$

with m controlling the total emigration rate and g the shape of the distribution. Note that (i) geometric distributions cannot be used to achieve high kurtosis with large migration rates; (ii) the stepping stone dispersal is the limit of the geometric distribution with $g \to 0$.

3.2.2 Implemented dispersal distributions

Here is a list with detailed descriptions of all dispersal distributions that are yet implemented in `IBDSim`. Note that this list will be regularly updated to take into account all new dispersal distribution implementations. For all the descriptions below, dx_{max} sets the maximum distance, in lattice steps, than can be moved in one generation, and all parameter values refers to the parameters described above.

`case '0'` truncated Pareto distribution (see p.21) with $\sigma^2 = 4$ and $dx_{max} = 15$, and parameters $M = 0.3$ and $n = 2.51829$.

`case '1'` stepping stone distribution with total emigration rate $M = 2/3$.

`case '2'` truncated Pareto distribution (see p.21) with $\sigma^2 = 1$ and $dx_{max} = 49$, and parameters $M = 0.599985$ and $n = 3.79809435$.

`case '3'` truncated Pareto distribution (see p.21) with $\sigma^2 = 100$ and $dx_{max} = 48$, and parameters $M = 0.6$ and $n = 1.246085$.

`case '4'` truncated Pareto distribution (see p.21) especially designed for lattice with one empty node over two (see p.18) with $\sigma^2 = 1$ and $dx_{max} = 48$, and parameters $M = 0.824095$ and $n = 4.1078739681$.

`case '5'` truncated Pareto distribution (see p.21) especially designed for lattice with one empty node over three (see p.18) with $\sigma^2 = 1$ and $dx_{max} = 48$, and parameters $M = 0.913378$ and $n = 4.43153111547$.

`case '6'` truncated Pareto distribution (see p.21) with $\sigma^2 = 20$ and $dx_{max} = 48$, and parameters $M = 0.719326$ and $n = 2.0334337244$.

`case '7'` truncated Pareto distribution (see p.21) with $\sigma^2 = 10$ and $dx_{max} = 49$, and parameters $M = 0.702504$ and $n = 2.313010658$.

`case '8'` truncated Pareto distribution (see p.21) especially designed for lattice with one empty node over three (see p.18) with $\sigma^2 = 4$ and $dx_{max} = 48$, and parameters $M = 0.678842$ and $n = 4.1598694692$.

`case '9'` truncated Pareto distribution (see p.21) with $\sigma^2 = 4$ and $dx_{max} = 48$, and parameters $M = 0.700013$ and $n = 2.74376568$.

`case 'a'` stepping stone distribution with total emigration rate $M = 1/3$.

`case 'b'` custom stepping stone distribution with total emigration rate set in the input file by the option `Total_Emigration_Rate` p.19.

`case 'g'` custom geometric distribution with total emigration rate and shape set in the input file by the options `Total_Emigration_Rate` p.19 and `Geometric_Shape` p.19 respectively. Note that high kurtosis can not be achieved with a geometric distribution without small emigration rates.

`case 'P'` custom truncated Pareto distribution (see 3.2.1) with parameters M and n set in the input file by the options `Total_Emigration_Rate` and `Pareto_Shape`, respectively.

`case 'S'` custom Sichel mixture distribution with parameters ξ, ω and γ set in the input file by the options `Sichel_Gamma`, `Sichel_Xi`, `Sichel_Omega`. Some parameter values which gives biologically realistic dispersal distribution can be found in Watts *et al.* (2007).

3.3 Habitat shape and boundaries : from a torus to a plane

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 facilitate analytical developments. However, as such torus or circle models are not generally realistic, we implemented various edge effects in IBDSim :

- no edges: the lattice is represented on a circle or a torus for a one or a two-dimensional model respectively;
- reflective boundaries: 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;
- absorbing boundaries: the lattice is represented on a line or plane and trajectories of dispersal events are constrained by the fact that each movement has to happen inside the lattice (i.e. the probability mass of going outside the lattice is equally shared on all other movements inside the lattice).

3.4 Output files

IBDSim can generate different types of output files depending on the options chosen :

(i) all simulated data sets in 3 different formats: the extended input file format of Genepop v.4 (Rousset, 2008) with spatial coordinates of sampled individuals and two others specific file formats that can be read as input for MIGRATE (Beerli & Felsenstein, 2001) and MIGRAINE (Rousset & Leblois, 2007). See data file output options p.11 for a detailed description of each of those three formats.

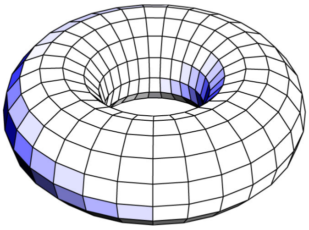

Figure 1: Graphical representation of a torus

(ii) a summary file named "Simul_Params.txt" where most parameter values used for the simulation are summarized and some statistics on the chosen dispersal distribution are computed (mean dispersal, second moment σ and kurtosis).

(iii) a summary statistic file named "Various_Statistics.txt" where the mean over all multilocus runs of various genetic statistics, such as TMRCA, probability of identity between pairs of genes, observed and expected heterozygosity (Nei, 1987), Cornuet's DH statistic (Cornuet & Luikart, 1996, Leblois *et al.*, 2006), variance in allelic size, Garza and Williamson's M statistic (Garza & Williamson, 2001), FIS and mean coalescence times are computed on the simulated data sets as well as theoretical expectations, based on mutation rates, populations sizes and number of possible allelic states for some of those statistic in models where relatively simple analytical results are available.

(iv) a file named "Iterative_Statistics.txt" with all records, for each simulated data file with details for each locus and mean values among loci of various genetic statistics (observed and expected heterozygosity, Cornuet's DH statistic, variance in allelic size, Garza and Williamson's M statistic, FIS and number of alleles). 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 loci as well as means for all loci for each data set (i.e. for each line) so that each statistic is represented by Locus_Number + 1 columns.

(v) a file named "Iterative_IdProb.txt" with frequencies of pairs of genes identical in state at all distances represented in the sample. Each line of this file represent one simulated data set and Identity Probability values are given in two columns as specific values for the simulated data set considered as well as mean values considering all previous simulated data sets.

(vi) A file named "Matrix_IdProb.txt" with the mean over all runs of probabilities of identities between pairs of genes computed on the generated data sets as a function of the location (i.e. spatial coordinates) of the genes on the lattice.

(vii) For each data set, a file named "EmpDisp_DataFileName" with the empirical effective dispersal distribution computed from all dispersal events that occurred during the multilocus coalescent trees used for the simulations. Dispersal distribution is represented as a table that can be used to plot an histogram. At the end of the file various statistics (mean, second moment σ, kurtosis and skewness) are computed on the whole distribution and on the semi distribution (axial values). Empirical dispersal distances are always computed considering the habitat as a plane, even if the simulation settings actually considers a torus. Descrepancies between theoretical and empirical dispersal distributions are thus expected when working on a torus, especially for small size lattices and/or large maximal dispersal distances.

(viii) A file named "MeanEmpDisp.txt" with the mean empirical effective dispersal distribution over all simulated data sets and, at the end of the file, various statistics (mean, second moment σ, kurtosis and skewness) are computed on the whole distribution and on the semi distribution (axial values). Format is the same than for the previous output file "EmpDisp_DataFileName".

3.5 Interaction with Genepop

Interaction of IBDSim 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). Genepopś behavior can now be controlled using an option 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 IBDSim. For example, users can easily evaluate the performance of two different estimators of the so-called neighborhood size under simulation conditions of their choice, by simulating samples using IBDSim and analyzing them using the Performance setting of Genepop V.4.

4 Credits (code, grants, etc.)

IBDSim uses R.J. Wagner's implementation of the Mersenne Twister random number generator, http://www-personal.umich.edu/~wagnerr/ and extracts of the “Mathlib : a C Library of Special Functions” code from the R foundation. IBDSim also uses small bits of code from Numerical Recipes in C by Press et al. This work was financially supported by the AIP no. 02002 “biodiversité” from the Institut Français de Biodiversité.

5 Copyright

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.

Cornuet, J. M. & Luikart, G., 1996. Description and power analysis of two tests for detecting recent population bottlenecks from allele frequency data. *Genetics* **144**: 2001–2014.

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

Di Rienzo, A., Peterson, A. C., Garza, J. C., Valdes, A. M., Slatkin, M. & Freimer, N. B., 1994. Mutational processes of simple-sequence repeat loci in human populations. *Proc. Natl. Acad. Sci. U. S. A.* **91**: 3166–3170.

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

Garza, J. C. & Williamson, E. G., 2001. Detection of reduction in population size using data from microsatellite loci. *Molecular Ecology* **10**: 305–318.

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. Sunderland, MA.

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. IBD_Sim: A computer program to simulate genotypic data under Isolation by Distance. *Molecular Ecology Ressources* **9**: 107–109.

Leblois, R., Estoup, A. & Streiff, R., 2006. Habitat contraction and reduction in population size: Does isolation by distance matter? *Molecular Ecology* **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.

Nei, M., 1987. *Molecular Evolutionary Genetics.* Columbia University Press, New York.

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.

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 FStatistics Under Isolation by Distance. *Genetics* **145**: 1219–1228.

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

Rousset, F., 2008. GENEPOPÆ007: a complete re-implementation of the GENEPOP software for Windows and Linux. *Molecular Ecology Resources* **8**: 103–106.

Rousset, F. & Leblois, R., 2007. Likelihood and approximate likelihood analyses of genetic structure in a linear habitat: performance and robustness to model mis-specification. *Mol. Biol. Evol.* **24**: 2730–2745.

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