

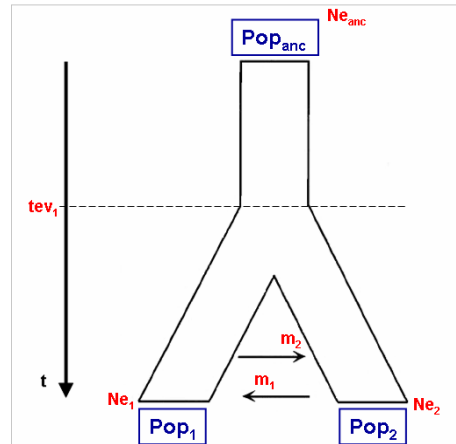
Joao Lopes and Mark Beaumont
 University of Reading, Whiteknights, PO Box 228, Reading RG6 6AJ

Contacts information:

joao.lopes@rdg.ac.uk
 www.rdg.ac.uk/~sar05sal

Introduction

With the development of new molecular biology techniques it is possible to scan the entire genomes of multiple individuals within a population, for instance, in a dense map of SNP (Single Nucleotide Polymorphism) markers, microsatellite markers or DNA sequence data. These data can then be used to infer aspects of the history of a population, such as population growth and migration rates, recombination rates, and selection coefficients. The Bayesian statistical paradigm offers an efficient framework for such inferences, because it allows maximal extraction of information from data under the specified model, and background information can be incorporated via prior distributions.



The wide use of MCMC (Markov Chain Monte Carlo) methods in the early 1990s made possible accurate statistical inferences from molecular genetic data (Wilson and Balding, 1998). Further increases in the volume of data as well as the ambitions of researchers have quickly outstripped the capabilities of these computer-intensive methods. In the past few years, some developments in computation statistics have arisen that pushed back the boundaries of the models that can be analysed at the cost of some approximation (Excoffier and Heckel 2006; Marjoram and Taveré, 2006).

These developments consist of: characterization of the data by summary statistics; and a simulation method that removes the necessity to deal analytically with common Bayesian functions such as the prior, posterior, marginal and likelihood distributions. These Bayesian computational methods have come to be known as ABC (approximate Bayesian computation). Early studies suggest that this method is highly competitive when compared to full-data analysis approaches (Estoup, *et al.*, 2001; Beaumont, *et al.*, 2002; Excoffier, *et al.*, 2005; Hickerson, *et al.*, 2006).

popABC is a package that enables the user to work within an ABC framework. It can be used either for investigation and research or education and teaching purposes. It is a GNU licensed program and its source code is available for population geneticists that aim to expand these Bayesian methods.

Index

1	Applications.....	- 3 -
	Parameters to be estimated	- 3 -
	Software features.....	- 4 -
2	Type of files used	- 6 -
	Input files.....	- 6 -
	Output files.....	- 7 -
	Other files	- 8 -
	Converting different file types to be used in popABC	- 8 -
3	The command line executables	- 11 -
	rejection v1.0	- 11 -
	shuffle v1.0	- 12 -
	simulate v1.0.....	- 12 -
	summdata v1.0.....	- 13 -
4	The ASCII-based menu	- 14 -
5	Quick start guides	- 16 -
	Compile the executable files	- 16 -
	Quick start (using the menu interface).....	- 16 -
	Quick start (using the individual executables).....	- 18 -
6	Approximate Bayesian computation	- 19 -
	Assumptions	- 20 -
	Mutation Models.....	- 20 -
	Summary Statistics used	- 21 -
7	Other software	- 26 -

1 Applications

The **popABC** program has been developed to infer the demographic history of related populations or species. It can assume tree models based on the Isolation-with-Migration model (Nielsen and Wakeley, 2001; Hey and Nielsen, 2004), where two populations descend directly from a common ancestor one. After a splitting event the populations are kept isolated from each other with or without the occurrence of migration events (Nielsen and Wakeley, 2001). The complexity of the model varies according to the considered number of modern populations and the migration matrix assumed.

A simple case is when considering just two modern populations, this involves six demographic parameters. A scenario considering only one more population, however, has as many as eleven parameters (Figure 1).

In situations with more than two populations the topology of the population tree is itself a parameter. For instance, in a three-population case there are three possible branching histories: Pop1 and Pop2 splitting from the most recent ancestor population; Pop1 and Pop3 splitting from the most recent ancestor population; or Pop2 and Pop3 splitting from that same ancestor population.

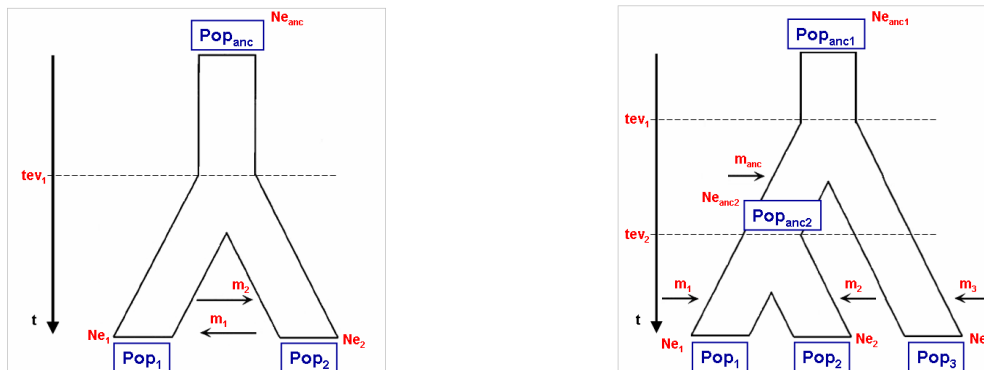


Figure 1 – Two-population (left) and three-population scenarios (right) in an Isolation with Migration model. The demographic historic parameters are discriminated in red.

In the current version of the program the user can specify an unlimited number of modern populations. The migration matrix can be specified within some limitations. The package can perform model-choice inferences automatically on tree topologies with at the most **five modern populations**. In scenarios with six or more populations the number of possible topologies is immense. In such situations the user should fix the demographic history in a few most likely tree branching histories.

Parameters to be estimated

Tree branching history (M) – This parameter corresponds to the topology of a population tree. It is a categorical variable where each identifier corresponds to a particular tree branching history.

Modern population sizes (N_e) – These parameters correspond to the effective population size of modern populations and are measured as number of individuals.

Ancestral population sizes ($N_{e,anc}$) – These correspond to effective population sizes of ancestor populations. They are also measured as number of individuals.

Time of splitting events (t_{ev}) – These parameters corresponds to the time, in years, when one population splits originating two different populations.

Immigration rates (m) – These parameters correspond to the fraction of immigrants of a population, i.e. the fraction of migrants entering a population, irrespective of their population of origin. These parameters are measured in fraction of immigrates per generation.

Mutation rates (μ) – The mutation rate is measured as the number of mutations per generation per locus.

Recombination rates (r) – The recombination rate is measured as the number of recombination events per generation per locus.

Software features

popABC as it is allows for a considerable variety of studies using DNA data within an ABC approach. Some other features however are planned to be added in a later stage. The already implemented features are summarized below.

1. Population Model:

Build population trees using "Isolation with Migration" events:

- population splitting events;
- migration events.

Migration rates are set as proportion of immigrants in a population. Probabilities for the population of origin of migrates can be set freely.

Use an unlimited number of modern populations.

2. Genealogical Model:

Use two different DNA data types separately or at the same time:

- Microsatellite DNA;
- DNA Sequence.

Use two different mutation models with varying mutation rate:

- Microsatellite DNA: stepwise mutation model
- DNA sequence: infinite-sites model

Use an unlimited number of loci from different DNA origins:

- nuclear DNA;

- X-linked DNA;
- Y-linked DNA;
- mitochondrial DNA.

3. Statistical Model:

Use different prior distributions:

- generalized gamma distribution;
- uniform distribution;
- normal and lognormal distributions.

Use different combinations of summary statistics:

- Microsatellite DNA: 6 summary statistics;
- DNA Sequence: 9 summary statistics.

4. Possible Inferences:

Estimate demographic parameters:

- size of the populations;
- splitting time of related populations;
- migration rates;

Estimate genetic population parameters:

- mutation rate (in one locus or among several loci);
- recombination rate (in one locus or among several loci).

Compare different models for the same analysis:

- between different migration patterns
- with migration vs. without migration;
- with recombination vs. without recombination;
- between population topologies (for more than 2-population models);

5. Usability:

Can be used in two different formats:

- user-friendly ASCII menu;
- strictly command line-based executables.

2 Type of files used

The *popABC* uses a file system to allow the users to easily identify each particular file type. This system is constituted by 9 different files:

- Data (*.dat)
- Table (*.len)
- Priors (*.prs)
- Prior sample (*.pri)
- Report (*.txt)
- Rejection (*.rej)
- Sample size (*.ssz)
- Summary statistics set (*.sst)
- Target (*.trg)

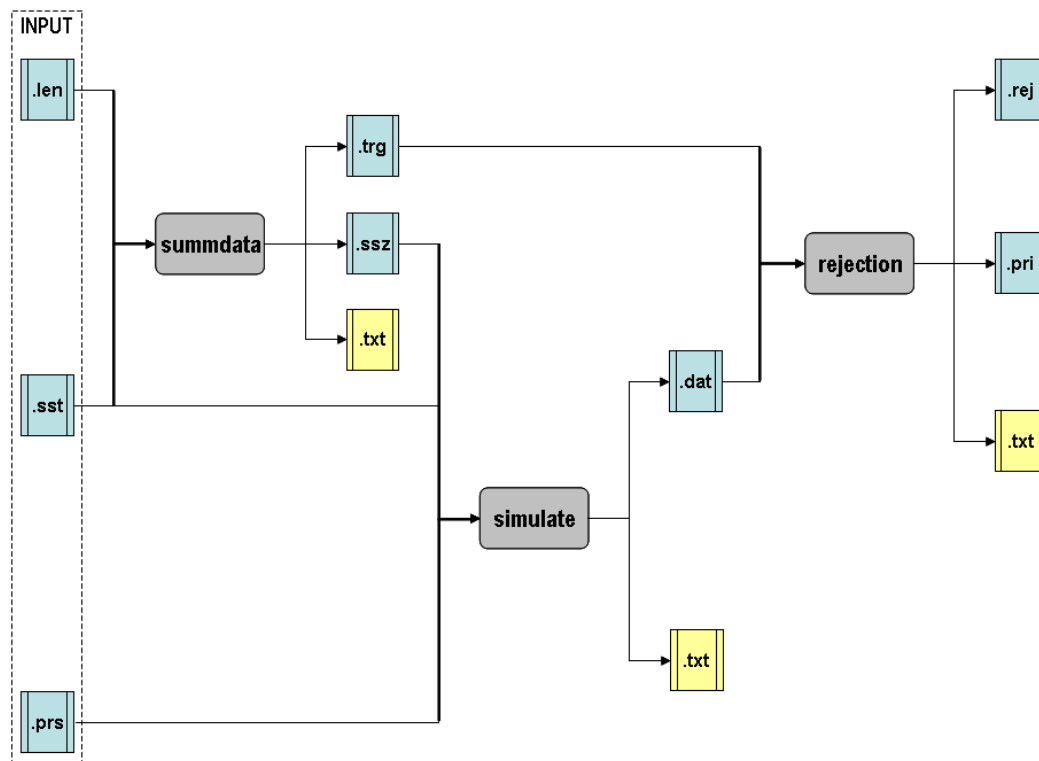


Figure 2 – Schematic representation of the File and Executables System: Files (blue), Executables (grey) and Report Files (yellow).

Input files

Table files (*.len)

The table files specify all the different haplotypes/alleles present and have information on their frequencies in each population separately.

The first lines can be used to state any comments as long as they start with the '#' character. The following line should state the number of populations present, the next one should inform about the number of loci and a following one

should declare the type of DNA data of each locus ('s' for sequence data and 'm' for microsatellite data).

Then, for each locus the number of different alleles/haplotypes present should be stated followed by their frequency. In the presence of sequence data the line after the frequencies should present haplotypes identifiers followed by a legend stating the different haplotypes that each identifier represents.

Summary statistics set (*.sst)

The summary statistics set file specifies which summary statistics to use in a particular run (see section 6 for a detail description of the implemented summary statistics).

Priors files (*.prs)

The prior files are where the prior distributions of each parameter are specified. This file also has other information. The first line states the number of iterations to run the ABC algorithm, the generation time of the individuals (in years), the number of populations considered and the number of loci present. The second line should have scaled parameters for each locus, this scaled parameters concern a locus inheritance factor (for nuclear DNA the scalar should be 1.00, for X-linked DNA it should be 0.75 and for mitochondrial or Y-linked DNA the scalar should be 0.25). In a different line, the type of DNA of each considered locus should be stated ('s' for sequence data and 'm' for microsatellite data).

All the other lines should have information on the prior distributions of the parameters.

These files have a detailed legend and also a representation of the considered tree topology.

Output files

Rejection files (*.rej)

The rejection files consist of the points chosen from a data file according to their distance to the 'real' data, thus, representing the results of the rejection method analysis to a data file (*.dat). As in a data file the first numbers of a line will be the parameters and the last the correspondent summary statistic values. The order of the parameters and of the summary statistics in a rejection file (*.rej) are stated in the report file (*.txt) correspondent to the data file (*.dat) that generated it.

Prior sample files (*.pri)

These files can be use to control the analysis made. They consist of random points picked from the previously chosen parameter values. The distribution of the points in the prior sample files should, then, be similar to the prior distributions of the parameters themselves.

Report files (*.txt)

The report files are used to inform about different analyses: the 'real' data summarization; the iterative simulation step and the rejection-step of the ABC procedure. These files are mainly used to keep track of the options chosen for the mentioned steps. They state several Input, Output and Runtime information.

Other files

Target files (*.trg)

The target files are constituted only by the values of the considered summary statistics obtained by summarizing the 'real' data set. The order of the summary statistics are stated in its report file (*.txt).

Sample size files (*.ssz)

The sample size files have information about the number of samples per population per loci. Each line of the file reflects a particular locus, and has one value of sample size for each population analysed.

Data files (*.dat)

The data files contain the points simulated by the ABC algorithm. The first numbers of a line are the parameter values used to simulate the genetic data. The last ones are the values of the summary statistics that summarize those same genetic data.

The order of the parameters and of the summary statistics in a data file (*.dat) are stated in the correspondent report file (*.txt).

Converting different file types to be used in popABC

The *popABC* package allows for conversion of input files of two popular population genetics programs to its own input files (*.len). These programs are IM (Hey and Nielsen, 2004) and GenePop (Rosseau, 2008). Nexus files (Maddison et al., 1997) can also be converted to *popABC* usable files. It is also possible to create table files (.len) from a simple text file describing population samples (*.pop) developed for *popABC* itself.

IM input files

Typical IM sample files can be automatically converted to table files (*.len). Note, however, that strictly only sequence data and microsatellite markers can be used by *popABC*. In addition the mutation rates disclosed in the IM input files will not be used. An example of an IM sample file is presented below:

```
# im test data
population1 population2
3
locus1 1 1 13 I 1
pop1_1 ACTACTGTCATGA
pop2_1 AGTACTATCACGA
strexample 2 2 1 S1 1
strpop11a 23
strpop11b 26
strpop21a 25
strpop21b 31
```

GenePop input files

Input files for GenePop can be automatically converted to table files (*.len). The following is an example of a GenePop sample file:

```
Title line: "Grape populations in southern France"
ADH Locus 1
ADH #2
```



```

ADH three
ADH-4
ADH-5
mtDNA
Pop
Grange des Peres , 0201 003003 0102 0302 1011 01
Grange des Peres , 0202 003001 0102 0303 1111 01
Grange des Peres , 0102 004001 0202 0102 1010 01
Grange des Peres , 0103 002002 0101 0202 1011 01
Grange des Peres , 0203 002004 0101 0102 1010 01
POP
Tertre Roteboeuf , 0102 002002 0201 0405 0807 01
Tertre Roteboeuf , 0102 002001 0201 0405 0307 01
Tertre Roteboeuf , 0201 002003 0101 0505 0402 01
Tertre Roteboeuf , 0201 003003 0301 0303 0603 01
Tertre Roteboeuf , 0101 002001 0301 0505 0807 01
pop
Bonneau 01 , 0101 002002 0304 0805 0304 01
Bonneau 02 , 0201 002002 0404 0505 0304 01
Bonneau 03 , 0101 002100 0304 0505 0101 01
Bonneau 04 , 0101 100100 0204 0805 0304 01
Bonneau 05 , 0101 100002 0104 0808 0304 01
Pop
, 0000 002001 0202 0402 0007 01
, 0200 002001 0202 0205 0707 01
, 0010 002001 0101
0105 0807 01
last pop, 0101 002001 0101 0401 0807 02

```

Nexus files (*.nex)

Nexus files can be used in a wide variety of software. For a good description of this file types one can refer to Maddison et al. (1997). There are some restrictions to these files configuration when using them in **popABC**, namely, the only blocks that can be used are DATA and SETS. As usual, comments can be written between brackets, i.e. “[]”. An example of a Nexus file is described:

```

#NEXUS
[Example of a Nexus file used in PopABC1.0]

[Individuals defined here]
BEGIN DATA;
DIMENSIONS NTAX=40 NCHAR=120;
FORMAT DATATYPE=DNA;
MATRIX
sample_b TTTAATTTTCATAGTGGGGTTAGGGTAGATTGACTATTTTTAGACTCCATTTGGCTGGA
ATTTCTTCTATTCTTGGAGCGATTAATTTTATTACTACAATTATTAATATACGACCTAAA
sample_b1 TTTAATTTTCATAGTGGGGTTAGGGTAGATTGACTATTTTTAGACTCCATTTGGCTGGA
ATTTCTTCTATTCTTGGAGCGATTAATTTTATTACTACAATTATTAATATACGACCTAAA
sample_b1c TTTAATTTTCATAGTGGGGTTAGGGTAGATTGACTATTTTTAGACTCCATTTGGCTGG
AATTTCTTCTATTCTTGGAGCGATTAATTTTATTACTACAATTATTAATATACGACCTAAA

[not used]
sample_B1M2i TTTAATTTTCATAGTAGGATTAGAGTAGATTGACTATTTTTAGACTCCATTAGCT
GGAATCTCTTCTATTCTTGGAGCAATTAATTTTATCACTACAATTATTAATATACGACCTAAA

sample_B1M2l TTTAATTTTCATAGTAGGATTAGAGTAGATTGACTATTTTTAGACTCCATTAGCT
GGAATCTCTTCTATTCTTGGAGCAATTAATTTTATCACTACAATTATTAATATACGACCTAAA
sample_B16a TTTAATTTTCATAGTAGGATTAGAGTAGATTGACTATTTTTAGACTCCATTAGCTG
GAATCTCTTCTATTCTTGGAGCAATTAATTTTATCACTACAATTATTAATATACGACCTAAA
sample_B16b TTTAATTTTCATAGTAGGATTAGAGTAGATTGACTATTTTTAGACTCCATTAGCTG
GAATCTCTTCTATTCTTGGAGCAATTAATTTTATCACTACAATTATTAATATACGACCTAAA
sample_B16c TTTAATTTTCATAGTAGGATTAGAGTAGATTGACTATTTTTAGACTCCATTAGCTG
GAATCTCTTCTATTCTTGGAGCAATTAATTTTATCACTACAATTATTAATATACGACCTAAA
sample_B16d TTTAATTTTCATAGTAGGATTAGAGTAGATTGACTATTTTTAGACTCCATTAGCTG
GAATCTCTTCTATTCTTGGAGCAATTAATTTTATCACTACAATTATTAATATACGACCTAAA
sample_B16e TTTAATTTTCATAGTAGGATTAGAGTAGATTGACTATTTTTAGACTCCATTAGCTG
GAATCTCTTCTATTCTTGGAGCAATTAATTTTATCACTACAATTATTAATATACGACCTAAA
;
END;

```

```
[Populations defined here]
BEGIN SETS;
  TAXSET 'population1' = 1-3 10;
  TAXSET 'population2' = 5-9;
END;
```

popABC sample files (*.pop)

popABC sample files begin with optional first lines for comments starting with the '#' symbol. The next line should contain the number of loci present in the data. The line following should be composed by a set of letters that identify the type of DNA data of each locus ('m' for microsatellites data and 's' for sequence data). In the next lines the genetic information of all the individuals should be stated, one line per individual. In these lines there should be a name to identify the population to which the individual belongs to. Then it should be stated the genetic information of all the loci, separated by spaces or tabs. For microsatellites one should have an integer value corresponding to the number of repetitions of the microsatellite, for sequence data a series of '0' or '1' should be used representing ancestor and derived states respectively. In case of absence of data a '0' should be used for microsatellites and a '9' for each segregating site of a sequence. Below there is an example of a **popABC** sample file (*.pop):

```
# Example of a .pop file used in PopABC1.0
# date: Mon May 11 23:05:46 2009

6
s s m m s s

popA 001010010000001111 000001100110000000 6 8 00000 00000
popA 001010010000001111 000000010001001011 6 8 00000 00100
popA 001010010000001111 000000010001001011 7 8 00000 99999
popA 001010010000001111 000000010001001011 7 8 00000 99999
popA 001010010000001111 000000010001001011 7 0 00000 99999
popA 001010010000001111 000000010001001011 0 7 00000 99999
popA 001010010000001111 000000010001001011 7 9 00000 99999
popA 000001100110000000 000000010001001011 7 9 00000 99999
popA 000001100110000000 000000010001001011 9 7 00000 99999
popA 000001100110000000 000000010001001011 8 0 00000 99999
populationB 000001001110000000 100001000110000000 10 10 00000 00000
populationB 000101000000010000 999999999999999999 10 0 00000 10000
populationB 000101000000010000 000001001110000000 11 0 00000 00000
populationB 0100000100000001011 999999999999999999 11 0 00000 10001
populationB 0100000100000001011 999999999999999999 11 0 00000 00000
populationB 0001010000000110000 999999999999999999 11 0 00000 10000
populationB 999999999999999999 000001001110000000 11 0 00000 00000
populationB 999999999999999999 999999999999999999 10 0 10001 00000
populationB 999999999999999999 000001001110000000 0 0 00010 10000
populationB 999999999999999999 999999999999999999 0 9 10000 10000
populationB 000001001110000000 999999999999999999 7 0 00000 10000
populationB 999999999999999999 000001001110000000 10 0 00000 99999
populationB 000001001110000000 999999999999999999 10 10 00000 10001
populationB 999999999999999999 999999999999999999 0 0 11000 10001
```

3 The command line executables

The *popABC* package can be used either within a user-friendly menu-based program or strictly with command line executables. The advantage of using the last is the gain on usability. If it is intended to run multiple jobs in parallel or consecutively, it is simpler and more efficient to use a batch file that runs the executables.

These command line executables are independent of each other and run a particular step of the ABC algorithm. They comprise 4 executable files that will be described in this section:

- rejection v1.0
- shuffle v1.0
- simulate v1.0
- summdata v1.0

rejection v1.0

This program prints out a proportion, previously specified, of the simulated points that are closer in Euclidian distance to the 'real' data point, i.e. rejection step of the ABC algorithm (Pritchard, et al., 1999). It will also create a file which will contain at the most 10000 sampled points from the prior distributions.

authors: Joao Lopes and Mark Beaumont
workplace: University of Reading
date: 1st May 2009

Arguments:

- 1) input filename (*.dat)
- 2) input filename (*.trg)
- 3) output filename (*.rej)
- 4) number of parameters
- 5) number of summary statistics
- 6) tolerance of the rejection step (from 0 to 1)

Input files:

Data file (.dat)
 Target file (.trg)

Output files:

Posterior sample file (.rej)
 Prior sample file (.pri)
 Report file (_rej.txt)

Example:

rejection.exe input/output.dat input/target.trg output/results 9 17 0.1

shuffle v1.0

This function changes the random numbers table used by *popABC* and stored in INTFILE.

author: Joao Lopes and Mark Beaumont
workplace: University of Reading
date: 1st May 2009

Arguments:

1) number of iterations to perform to the INTFILE

Input files:

INTFILE

Output files:

INTFILE

Example:

shuffle.exe 1000000

simulate v1.0

This function simulates the coordinated points (summary statistics, parameters) to be used in the Approximate Bayesian Computation framework.

authors: Joao Lopes and Mark Beaumont
workplace: University of Reading
date: 1st May 2009

Arguments:

- 1) input filename (*.prs)
- 2) input filename (*.ssz)
- 3) input filename (*.sst)
- 4) output filename (*.dat AND *.txt)
- 5) print or not the .len file of every genetic tree (0-don't print; 1-print)
- 6) print or not the .mut file (0-don't print; 1-print)
- 7) print or not the .rec file (0-don't print; 1-print)

Input files:

Prior file (.prs)
Sample Size file (.ssz)
Summary Statistics file (.sst)

Output files:

Data file (.dat)
Report file (.txt)
Mutation rates file (.mut)*
Recombination rates file (.rec)*
Simulated Genetic data file (.len)*

Example:

```
simulate.exe input/priors1.prs input/samp.ssz input/sstats.sst output/output 0 1 0
```

*- optionally created files.

summdata v1.0

This function uses a .len file and creates two output files, one containing the summary statistics of the given file and the other containing the size of the sampled populations.

authors: Joao Lopes and Mark Beaumont
workplace: University of Reading
date: 1st May 2009

Arguments:

- 1) input filename (*.len)
- 2) input filename (*.sst)
- 3) output filename (*.trg AND *.szz)

Input files:

frequency table file (.len)
Summary Statistics file (.sst)

Output files:

Target file (.trg)
Sample Size file (.szz)

Example:

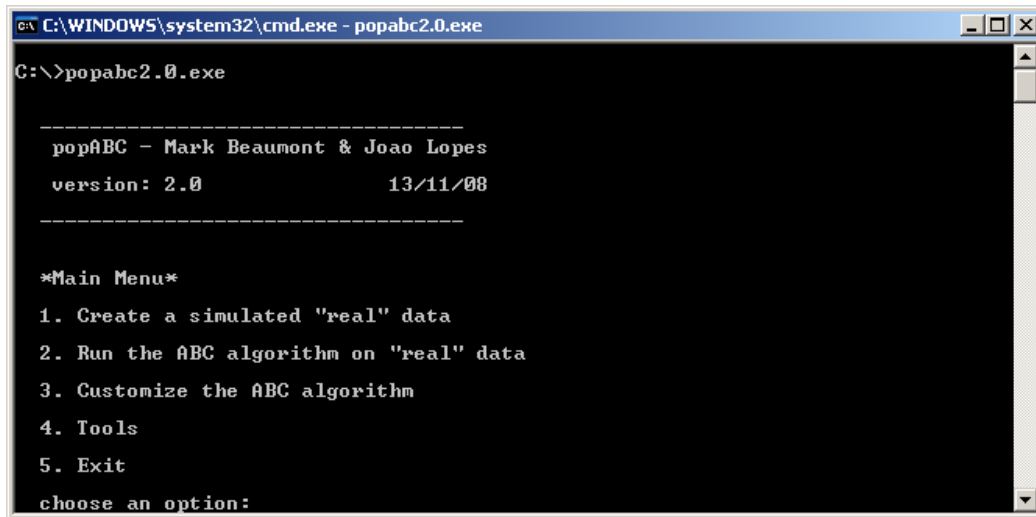
```
summdata.exe input/input.len input/sstats.sst output/target
```

4 The ASCII-based menu

popABC can also be used within a user-friendly stand-alone application. This application is especially suitable to situations when the user only wants to run the ABC algorithm on a few particular data sets or when the user intentions are to use one of the tools that come with the **popABC** menu-based application.

This program mode was build to be user-friendly and can account for errors when choosing some of the function options. Also, by using this menu-based program one can easily understand what potentialities this application has and to be aware of the existing functionalities.

It is advised, therefore, to start using **popABC** within the menu mode. After having a good ‘feel’ of the potentialities of the program, the user should switch to the individual command executables. These last allow the user to run Approximate Bayesian Computation methods with greater flexibility: running these applications sequentially or in parallel becomes easier and their submission to cluster machines will be simpler.



```

C:\WINDOWS\system32\cmd.exe - popabc2.0.exe
C:\>popabc2.0.exe

-----
popABC - Mark Beaumont & Joao Lopes
version: 2.0          13/11/08
-----

*Main Menu*
1. Create a simulated "real" data
2. Run the ABC algorithm on "real" data
3. Customize the ABC algorithm
4. Tools
5. Exit
choose an option:

```

Figure 3 – Screenshot of the “Main Menu” of the popABC menu mode.

The stand-alone application is composed by three menus, each one with their own functionalities.

The “Main Menu” presents four options plus the exit one (Fig. 4). The first one assists the user to simulate genetic data to be used as ‘real’ data. The second one runs the ABC algorithm continuously from the beginning to the end. It starts by using a raw sample data and finishes after obtaining the posterior distribution using the rejection-step (Pritchard, *et al.*, 1999). The third option directs the user to another menu where it is possible to customize the ABC algorithm. As for the fourth option, it displays a list of tools available with the program.

```

C:\WINDOWS\system32\cmd.exe - popabc2.0.exe
1. Create a simulated "real" data
2. Run the ABC algorithm on "real" data
3. Customize the ABC algorithm
4. Tools
5. Exit
choose an option: 3

*Customize ABC Menu*
1. Summarize 'real' data with summstats set
2. Run simulations from prior file (<.prs>)
3. Perform rejection step
4. Exit to Main Menu
choose an option:

```

Figure 4 – Screenshot of the “Customize ABC Menu” of the *popABC* menu mode.

The “Customize ABC Menu” presents three options, as well as, the exit one (Fig. 5). Each option allows the user to start the ABC algorithm in a different step. One can either summarize data to be study with the ABC approach; run the simulation machine to obtain the simulated points; or perform the rejection-step in order to obtain samples from the posterior distributions.

```

D:\WINDOWS\system32\cmd.exe - popabc.exe
choose an option: 4

*Tools Menu*
1. Build prior file (<.prs>)
2. Build summary statistics set file (<.sst>)
3. Convert sample file (<.pop>) to table file (<.len>)
4. Convert IMA input file to table file (<.len>)
5. Convert GenePop input file to table file (<.len>)
6. Convert Nexus file (<.nex>) to table file (<.len>)
7. Create sample file (<.pop>) from table file (<.len>)
8. Join data files together (<.dat>)
9. Exit to Main Menu
choose an option:

```

Figure 5 – Screenshot of the “Tools Menu” of the *popABC* menu mode.

Finally, the third menu is the “Tools Menu” (Fig. 6). This menu presents functions especially useful in the context of *popABC*.

Among this functions there is one that enables the user to join different data files (*.dat) which enables one to easily perform model-choice analysis.

Other functions that could be quite useful are the ones that allows the user to convert different sample files to a table file (*.len) that can be used as input for the *popABC* program. This menu also provide easy frameworks to build such files as summary statistics set files (*.sst) and prior files (*.prs).

5 Quick start guides

The **popABC** package is available for WinXP, Unix and MacOX operating systems. Each of these versions comes with executables previously compiled in the mentioned systems.

The ASCII-based menu can be run by double-clicking its respective icon. As for the command line executables they should be run in a command line window. In a WinXP environment to access this application you should select Run from the Start menu and type `cmd`. In a MacOX system you need to select Terminal from the Applications folder. All the executables can be found in the bin folder of the **popABC** package.

If necessary, to compile the files, follow the instructions below.

Compile the executable files

First you need to compile all the executables by running the Makefile in the base directory. Just type the following instruction in the command line:

```
make
```

The executables `popabc.exe`, `rejector.exe`, `shuffled.exe`, `simulate.exe` and `summdata.exe` should have been created in the base directory.

If you want to delete the files created by running the Makefile just type in the base directory:

```
make clean
```

Quick start (using the menu interface)

A good starting point to get use to **popABC** is using a toy example with example files provided in the program's package.

First you should run the program by clicking the icon `popabc.exe` (in case you are running Windows Operating Systems) or by typing the following:

```
./popabc.exe (or .\popabc.exe depending on the Operating System)
```

Now, you should choose the option 2 ("Run the ABC algorithm on 'real' data") of the "Main Menu" by typing '2'.

Now you are going to create a `.trg` and a `.ssz` file from a `.len` file to be used as 'real' data. Type the following path for the table file:

```
examples/toytable.len
```

Then you have to pick the set of summary statistic to use in the analysis. Just choose the following file:

```
examples/toysim.sst
```


Finally, just choose the name (no extension) you want to give to the files that are going to be created:

```
examples/toytarget
```

Both `.trg` and `.ssz` files should be created. As well as a report file.

Now you need to specify which prior file to use to simulate the points. Choose this file:

```
examples/toyprior.prs
```

Afterwards you need to choose a name (no extension) to give to the files to be created:

```
examples/toydata
```

You then have to choose if you want to print information about the mutation or the recombination rates. Just type '0' in both, to not to have this information disclosed. Now just wait a couple of minutes until all the points have been simulated.

After the `.dat` file has been created you are going to run the ABC's rejection step. You will need to input the name (no extension) of the rejection file to be created:

```
examples/toyreject
```

Then you have to specify the number of parameters and summary statistics obtained during the simulation process. You will find this information on the report file created ("files/toydata.txt"). Just type '16' and '18'. Finally you will have to choose the tolerance for this step, just choose '0.01' in order to obtain a rejection file with 100 points. This will create a report file in the same folder as the analysed file. The ABC algorithm is finally complete, now just type '5' to exit the program.

You can now plot histograms for each studied parameter so that you obtain a good approximation of their posterior distributions. We suggest using R (R Development Core Team, 2007) but any other less sophisticated spreadsheet will do. We provide a couple of R scripts. To use them just open R as usually, change the working directory to the **scripts** folder and use the source command to run the script `plot_hist.r` [`source("plot_hist.r")`]. Now just run the `plot_hist` function as follow:

```
plot_hist(rej="../examples/toyreject.rej",pri="../examples/toyreject.pri")
```

The `plot_line.r` script plots lines instead of histograms. It uses the `locfit` library (Loader, 1996) so to run it you will have to install that library. To run the function use the source command to have the function available [`source("plot_line.r")`] and then just type:

```
plot_line(rej="../examples/toyreject.rej",pri="../examples/toyreject.pri")
```

As you can see from the results you should probably use much more simulated points to begin with, the uniform priors for instance do not even resemble a uniform distribution.

After the rejection step is performed, the user is advised to run the regression step (Beaumont, et al., 2002) on the .rej file. For that you can use the R files available in Mark Beaumont's website (<http://www.rubic.rdg.ac.uk/~mab/>). We provide a simple R script to perform it. Just use the source command one again [`source("reg_step.r")`] and then just run the function as:

```
reg_step(data="../examples/toytarget.trg",rej="../examples/toyreject.rej",pri="../examples/toyreject.pri")
```

Quick start (using the individual executables)

Running the individual executables is similar to running them within the ASCII-based menu.

First you should run the **summdata.exe** to summarize your data:

```
./summdata.exe      examples/toytable.len      examples/toysim.sst
examples/toytarget
```

The target files should have been created. Now it is time to run the simulations using **simulate.exe**:

```
./simulate.exe     examples/toyprior.prs     examples/toytarget.ssz
examples/toysim.sst examples/toydata 0 0 0
```

After the simulations ended it is time to perform the rejection-step. For that run the **rejection.exe** with the following command:

```
./rejection.exe   examples/toydata.dat     examples/toytarget.trg
examples/toyresults 16 18 0.01
```

You will then obtain a rejection file which stores the simulated points that are closer to your real data. By plotting histograms for each studied parameter you can obtain a good approximation of their posterior distributions.

Again the user is advised to run the regression-step on the file resulting from performing the rejection-step (.rej).

6 Approximate Bayesian computation

In this section we give a brief description of the ABC algorithm. For a more detailed explanation the reader is invited to read a chapter on the subject from the book “Simulations, Genetics and Human Prehistory” (Beaumont, 2008). This document is provided in the *popABC* package

- First the prior has to be defined. It is assumed that the joint prior on all the parameters is the product of their individual priors, and so one prior distribution (ϕ_i) needs to be specified for each of the n parameter to be estimated. These prior distributions should be used to input previous knowledge on the analysis. For instance, if ϕ_1 is the size of effective population (N_e) and it was known that its value was around μ with a standard deviation of σ , it is possible, and even advised, to estimate this parameter by selecting values from a normal distribution defined as $\text{normal}(\mu, \sigma)$;

- One value per parameter is sampled from the n prior distributions. This set of n values is then used to obtain genetic data under a simulation process [in *popABC* the genetic data is simulated using a coalescent method (Hudson, 1990; Fu and Li, 1999; Nordborg, 2001) modified from the method described in Beaumont and Nichols, 1996)];

- After obtaining the genetic data, these are going to be summarized by a previously chosen set of summary statistics. A coordinated point between the parameter set (Φ) and the set of summary statistics (S) is going to be obtained;

- After repeating the previous process enough times a good characterization of the problem space (S, Φ) is expected to be obtained (Figure 6).

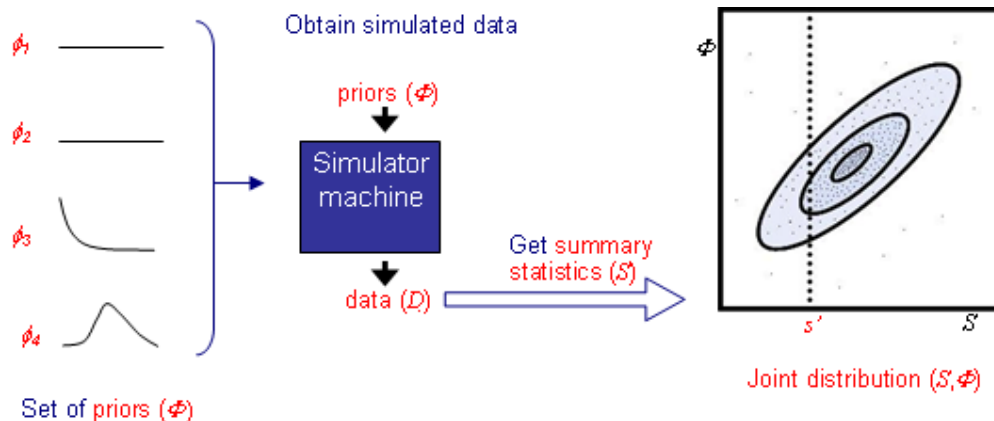


Figure 6 – Graphic representation of the simulation process used in *popABC*.

- The next step is to choose the closest simulated points to our DNA data sample. In order to perform this action the values of the summary statistic of our simulated points are going to be compared with the ones that summarize the sample data, s' ;

- Using the chosen points, a probability distribution of the parameter values is going to be obtained, *i.e.* the posterior distribution (Figure 7).

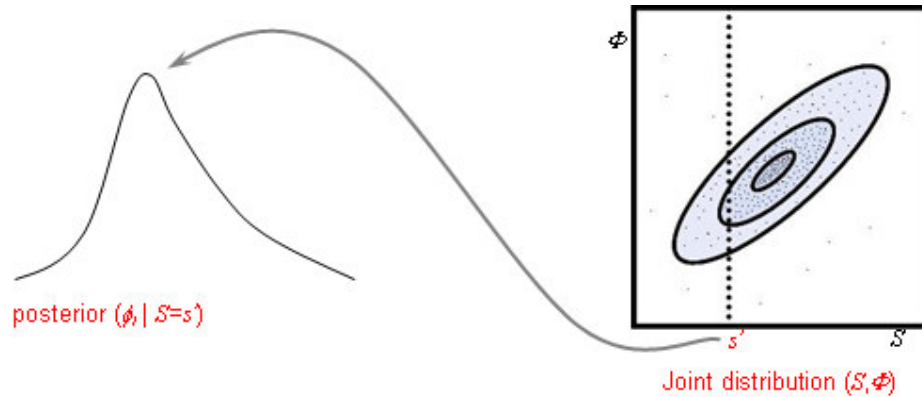


Figure 7 – Sampling from the posterior distribution.

Assumptions

The presented program, as the IM software (Hey and Nielsen, 2004), assumes an “Isolation with Migration” model. The assumptions of the latter are, then, followed by **popABC**. As described in IM manual this assumptions are the following:

- there should not be other populations that are more closely related to the sampled populations than they are to each other;
- and there should not be unsampled populations exchanging genes with the studied populations or their own ancestors (Nielsen and Wakeley, 2001).

There are also population genetics related assumptions that arise from the chosen simulation method, which is based on the coalescence:

- Model under neutrality: The simulation method assumes that the variation within the DNA data is neutral. It is not affected by selective pressures, either directional or balancing.
- Free recombination between loci (sequence DNA): The simulation method models each locus as having segregated independently over time.
- Mutations follow the Mutation Model considered: Accordingly to the DNA data used the **popABC** program assumes different mutation models (e.g. DNA sequence data assumes an Infinite Sites Mutation model; microsatellites data assumes a Stepwise Mutation model). Each considered model held assumptions about the occurred mutation process.

Mutation Models

In the present version of the program two different mutation models can be assumed according to the DNA data type considered:

The **Infinite Sites** (IS) model (Kimura, 1969). This model is applied to DNA sequence data. Under this model it is assumed that every mutation will occur in a different place from all the past ones in the history of the DNA sequence.

This assumption is in fact very much applicable for most DNA sequence data sets. Genes from the nucleus, for instances, generally have mutation rates on the order of 10^{-9} per base pair per generation. Unless a branch is very many generations long, it is not expect to find multiple mutations per base pair along the same lineage.

The Isolation with Migration model is mostly applied for relatively recent cases of population splitting (last 10 Mya). In this situation, where polymorphic sites tend to have low density along a sequence, the IS mutation model is perfectly suitable to be used.

The **Stepwise Mutation Model** (SMM) (Kimura and Ohta, 1978). This model is mostly applied to allelic variation. Each mutation causes an allele to increase or decrease by one step of the used alleles measured scale. Microsatellite or Short Tandem Repeat (STR) loci have high mutation rates, these mutations are translated as different numbers of repeats in an approximately stepwise manner (Estoup, *et al.*, 2002).

Summary Statistics used

Genetic data can be summarized by a large number of available summary statistics. Some of which are strictly related to a particular DNA data type. The available summary statistics were chosen according to their inference value in the literature. Some summary statistics are better to estimate particular demographic parameters. Others work better when used together (e.g. the mean of mutation frequency spectrum (MFS) summarizes the data more efficiently if used together with the standard deviation of MFS).

The choice of the summary statistics has, however, to consider the balance between a good summarization of the data and the increase of the dimensionality of the problem. This so-called curse of dimensionality is strictly related to the number of summary statistics considered.

The available summary statistics for microsatellite data are the following:

Hetetozygosity (H) – This index represents the total expected heterozygosity within samples. Heterozygosity is also known as gene diversity (Nei, 1987). It can be described by the following:

$$He = 1 - \sum_{i=1}^n (f_i)^2$$

where n is the number of alleles at the target locus, and f_i is the frequency of the i^{th} allele at the target locus.

Variance of alleles' length (varL) – This index is commonly used for population genetics.

$$Var(length) = \sum_{i=1}^n (x_i - \mu)^2 f_i,$$

where n is the number of alleles at the target locus, x_i is the alleles length of the i^{th} allele, μ is the average alleles length; and f_i is the frequency of the i^{th} allele at the target locus.

Number of different alleles (k) – This index just counts the number of different allele forms there are in a sample. Its usage to characterize genetic data set has been fairly common.

Kurtosis of alleles' length (kurL) – This index represents the fourth moment of the alleles' length. At the present time there is no study on its improvement on DNA data summarization. It is expected, however, to increase the amount of information acquired by the summarization using the variance of alleles' length.

$$Kur(length) = \frac{n \sum_{i=1}^n (x_i - \mu)^4 f_i}{\left(\sum_{i=1}^n (x_i - \mu)^2 f_i \right)^2} - 3,$$

where n is the number of alleles at the target locus, x_i is the alleles length of the i^{th} allele, μ is the average alleles length; and f_i is the frequency of the i^{th} allele at the target locus.

Shannon's index (sH) – This is a diversity index derived from information theory by (Shannon, 1948). It can be interpreted as the natural logarithm of the number of equally common haplotypes required to produce the measured value in a sample. It has been used occasionally in population genetics, e.g. (Sherwin, et al., 2006). This index is very similar to the heterozygosity index described above.

$$sH = - \sum_{i=1}^n f_i \ln f_i,$$

where n is the number of alleles at the target locus and f_i is the frequency of the i^{th} allele at the target locus.

Nm estimator using heterozygosity (Nm_H) – The estimation of FST levels from average divergence of pairs of sequences within population and over the whole population has long been suggested (Slatkin, 1991). Since early on, the relation between FST and Nm in an island model as been established (Wright, 1950). By combining these two approaches we can get the following index:

$$Nm = \frac{1}{2} \frac{H_w}{H_b - H_w}.$$

This estimator of Nm was simplified to:

$$Nm = \frac{H_w}{1 + H_a - H_w},$$

where H_a is the heterozygosity of all the populations pooled together and H_w is the heterozygosity measured within a population. In its calculation, when $H_a - H_w$ was less or equal to zero, the estimator was assumed to be equal H_w .

The summary statistics that can be used for inferences using DNA sequence data are the following:

Mean of pairwise difference (π) – It is based on the average number of nucleotide differences between two sequences randomly chosen from a sample. This index is commonly used in population genetics since its suggestion in 1979 by Nei and Li.

$$\pi = \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^{n-1} f_i f_j \pi_{ij},$$

where f_i and f_j are the frequencies of the i^{th} and the j^{th} sequence, respectively, in the population, and π_{ij} is the number of nucleotide differences per nucleotide site between the i^{th} and j^{th} sequences.

Number of segregating sites (S) – It is calculated just by counting the number of segregating sites in DNA sequence data. As the previous index, it is very common in population genetics studies, either in its raw format or weighted by the sample size, *i.e.* Watterson's index (1975).

Number of different haplotypes (k) – A common statement in population genetics is that this index in a random sample of sequences is not informative since, for long DNA sequences, all the sequences can be different from each other (Nei, 1987). A study by Depaulis and Michel Veuille (1998) however demonstrate that between the extremes of having the same sequence over and over and having all the sequences different from each other lays realistic conditions that can be highly informative.

Shannon's index (sH) – (As described in the microsatellites' summary statistics).

Mean of MFS (mMFS) – The Mutation Frequency Spectrum (MFS) is the distribution of segregating-sites frequencies. It has received a considerable attention in the last years from population geneticists. It is in fact the basis of a popular estimate among this field, the Tajima's D (1989). According to the study of Wakeley and Aliacar (2001) this spectrum can record information about migration regimes and extinction/recolonization events. Depending on having or not information to differentiate ancient from derivative forms of the data we can use unfolded or folded distributions (Bustamante, *et al.*, 2001), respectively. In the first situation the following formula is used to calculate this index:

$$\frac{1}{S_{total}} \sum_{i=1}^{S_{total}} x_i ,$$

where x_i is the number of segregating sites in i^{th} segregating site. And S_{total} is the total number of sites segregating.

In the case when the folded distribution is in use the formula is as followed:

$$\frac{1}{S_{pop}} \sum_{i=1}^{S_{pop}} x_i ,$$

where x_i is the folded number of segregating sites in i^{th} segregating site. And S_{pop} is the folded total number of sites segregating in a particular population.

Standard deviation of MFS (sdMFS) – (As described above). When the unfolded distribution is in use the formula for this index is the following:

$$\frac{1}{S_{total}} \sqrt{\sum_{i=1}^{S_{total}} (x_i - \bar{x})^2} ,$$

where the previous definitions are again in use and \bar{x} is the average number of segregating sites a long all the sites segregating.

When the folded distribution is in use the calculation of this index follows:

$$\frac{1}{S_{pop}} \sqrt{\sum_{i=1}^{S_{pop}} (x_i - \bar{x})^2} ,$$

where once again the previous definitions are in use and \bar{x} is once more the average number of segregating sites a long all the sites segregating.

Nm-related statistic using the number of segregating sites (Nm_S) – This index is very similar to the Nm estimator using the heterozygosity, but this one is intended to be used for haplotypes. The use of segregating sites to calculate FST values was first introduced in (Wakeley, 1998), who provided an estimator that was a complicated function of S. Inspired by this, we have noticed that the following statistic has a high correlation with Nm:

$$\frac{1}{2} \frac{S_w}{S_b - S_w} .$$

This can be simplified to:

$$Nm_S = \frac{S_w}{S_a - S_w} ,$$

where S_a is the number of segregating sites of all the populations pooled together and S_w is the segregating sites of a particular population. In its calculation, when $S_a - S_w$ was equal to zero, the estimator was assumed to be equal S_w . Note that this is not a direct estimator of Nm itself.

Number of private segregating sites (privS) – Wakeley and Hey (1997) noted the importance of classifying the segregating sites according to their exclusivity or not to a population. To the segregating sites exclusive of a population I call private segregating sites. This index is specially indicated to distinguish between isolated populations from populations changing migrants.

Frequency of private segregating sites [S(1)] – This index is related to the previous one. It calculates the average number of sites segregating exclusively in a population.

$$\frac{1}{privS} \sum_{i=1}^{privS} f_i ,$$

where f_i is the number of segregating sites in the i^{th} site segregating only in the considered population and $privS$ is the total number of sites segregating only in the considered population.

Number of values per summary statistics chosen:

It should be noted that most of the summary statistic actually comprises one value per number of modern population plus one value per each two populations pooled together. The exceptions are: the **mMFS**; **sdMFS**; **Nm_H**; **Nm_S**; **privS**; and **S(1)**. These are calculated only for each modern population.

7 Other software

IM (Hey & Nielsen)

The IM program is a MCMC-based program that is used to estimate demographic historic parameters. As a MCMC method, when analysing a great amount of data it can take up to several months to get a good mixing of the results' space and to reach convergence. However, being a full-likelihood Bayesian approach it should have, in theory, a better accuracy than an ABC method. **popABC** can then be quite useful by performing preliminary studies before starting a considerable time consuming analysis with MCMC methods.

In general, ABC methods have also the possibility to use more complex models. The IM software is available in the homepage of the authors.

<http://lifesci.rutgers.edu/~heylab/HeylabSoftware.htm>

DIY ABC (Cornuet et al)

DIY ABC is a program that also runs ABC analysis. It has a graphical user interface and a fully click-able environment. It can assume complex demographic models without migration. Currently it can only deal with microsatellite data. This program can also be used to compare different scenarios

<http://www1.montpellier.inra.fr/CBGP/diyabc>

msBayes (Hickerson et al)

This program uses hierarchical approximate Bayesian computation to estimate hyper-parameters for two major phylogeographic models: the first one assuming simultaneous divergence and the other assuming colonization across multiple co-distributed pairs of populations or species. It can be used with DNA sequence data. msBayes can be used to distinguish between the two mentioned phylogeographic models.

<http://msbayes.sourceforge.net>

Rejector (Jobin and Mountain)

Rejector is a software package that combines four different executables to perform the ABC-rejection algorithm. It can assume complex demographic models and can use several different markers, such as, DNA sequence data, AFLP's markers and microsatellites. This software relies on the SIMCOAL2 program (Laval and Excoffier, 2004) for the coalescent simulations.

<http://www.stanford.edu/~mjobin/rejector>

Serial SimCoal (Anderson et al)

This software also performs the full ABC-rejection algorithm. Serial SimCoal uses the SIMCOAL simulator to perform the coalescence. This allows it to use complex demographic models and several different markers. This package, however, presents a few alterations to that coalescent simulator which widens its usage. For example, Serial CimCoal allows the user to use ancient genetic data at the same time as modern data.

<http://www.stanford.edu/group/hadlylab/ssc/BayeSSC.htm>

R files (Beaumont)

After running the rejection method the user is advised to run a regression-step (Beaumont, et al., 2002). To perform this step there are some r scripts available

in the homepage of Mark Beaumont. In his web address there are also some R files that will easily calculate the mode of the posterior distributions, as well as credible intervals.

<http://www.rubic.rdg.ac.uk/~mab/>

R files (Blum)

A standard ABC-Regression algorithm uses a multi-linear regression to correct the parameter values according to the distance to the ‘real’ data set. Michael Blum and Olivier Francois developed an R script to perform a more sophisticated non-linear regression (Blum and Francois, 2008). This script is available in the authors’ web address.

http://membres-timc.imag.fr/Michael.Blum/my_publications.html

References

- Anderson, C.N.K., Ramakrishnan, U., Chan, Y.L. and Hadly, E.A. (2005) Serial SimCoal: A population genetics model for data from multiple populations and points in time, Oxford Univ Press, 1733-1734.
- Beaumont, M. and Nichols, R.A. (1996) Evaluating loci for use in the genetic analysis of population structure, Proceedings of the Royal Society of London, Series B 263, 1619-1626.
- Beaumont, M.A., Zhang, W. and Balding, D.J. (2002) Approximate Bayesian computation in population genetics, *Genetics*, 162, 2025-2035.
- Beaumont, M.A., (2008) Joint determination of topology, divergence time, and immigration in population trees. In Matsura, S., Forster, P. and Renfrew, C. (eds), *Simulations, Genetics and Human Prehistory*. McDonald Institute for Archaeological Research, Cambridge, 135-154.
- Blum, M.G.B. and Francois, O. (2008) Highly tolerant likelihood-free Bayesian inference: An adaptive non-linear heteroscedastic model, *Stat. Comput.*
- Bustamante, C.D., Wakeley, J., Sawyer, S. and Hartl, D.L. (2001) Directional Selection and the Site-Frequency Spectrum, *Genetics*, 159, 1779-1788.
- Cornuet, J.M. et al. (2008) Inferring population history with DIYABC: a user-friendly approach to Approximate Bayesian Computation, *Bioinformatics* 24, 2713.
- Depaulis, F. and Veuille, M. (1998) Neutrality tests based on the distribution of haplotypes under an infinite-site model, *Mol Biol Evol*, 15, 1788-1790.
- Estoup, A., Jarne, P. and Cornuet, J.-M. (2002) Homoplasy and mutation model at microsatellite loci and their consequences for population genetics analysis, *Molecular Ecology*, 11, 1591-1604.
- Estoup, A., Wilson, I.J., Sullivan, C., Cornuet, J.M. and Moritz, C. (2001) Inferring population history from microsatellite and enzyme data in serially introduced cane toads, *Bufo marinus*, *Genetics*, 159, 1671-1687.
- Excoffier, L. and Heckel, G. (2006) Computer programs for population genetics data analysis: a survival guide, *Nature Reviews. Genetics (Print)*, 7, 745-758.
- Excoffier, L., Estoup, A. and Cornuet, J.M. (2005) Bayesian analysis of an admixture model with mutations and arbitrarily linked markers, *Genetics*, 169, 1727-1738.
- Fu, Y.X. and Li, W.H. (1999) Coalescing into the 21st century: An overview and prospects of coalescent theory, *Theor Popul Biol*, 56, 1-10.
- Hey, J. and Nielsen, R. (2004) Multilocus methods for estimating population sizes, migration rates and divergence time, with applications to the divergence of *Drosophila pseudoobscura* and *D. persimilis*, *Genetics*, 167, 747-760.
- Hickerson, M.J., Dolman, G. and Moritz, C. (2006) Comparative phylogeographic summary statistics for testing simultaneous vicariance, *Molecular Ecology*, 15, 209-223.
- Hickerson, M.J. et al (2007) msBayes: Pipeline for testing comparative phylogeographic histories using hierarchical approximate Bayesian computation, *BMC Bioinformatics*, 8, 268.
- Hudson, R.R. (1990) Gene genealogies and the coalescent process, *Oxford Survey of Evolutionary Biology*, 7, 1-44.

- Jobin, M. J., Mountain, J. L. (2008) REJECTOR: software for population history inference from genetic data via a rejection algorithm. *Bioinformatics* 24, 2936.
- Kimura, M. (1969) The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations, *Genetics*, 61, 893-903.
- Kimura, M. and Ohta, T. (1978) Stepwise mutation model and distribution of allelic frequencies in a finite population, *Proc Natl Acad Sci U S A*, 75, 2868-2872.
- Laval, G. and Excoffier, L. (2004) SIMCOAL 2.0: a program to simulate genomic diversity over large recombining regions in a subdivided population with a complex history. Oxford Univ. Press, 2485–2487.
- Loader, C.R. (1996) Local Likelihood Density Estimation, *The Annals of Statistics*, 24, 1602-1618.
- Marjoram, P. and Tavaré, S. (2006) Modern computational approaches for analysing molecular genetic variation data, *Nat Rev Genet*, 7, 759-770.
- Maddison, D. R., Swofford, D. L., Maddison, W. P., (1997) NEXUS: an extensible file format for systematic information. *Syst. Biol.* 46, 590.
- Nei, M. (1987) *Molecular Evolutionary Genetics*. Columbia University Press.
- Nei, M. and Li, W.-H. (1979) Mathematical Model for Studying Genetic Variation in Terms of Restriction Endonucleases, *PNAS*, 76, 5269-5273.
- Nielsen, R. and Wakeley, J. (2001) Distinguishing migration from isolation: a Markov chain Monte Carlo approach, *Genetics*, 158, 885-896.
- Nordborg, M. (2001) Coalescent Theory. In Balding, D.J., Bishop, M. and Cannings, C. (eds), *Handbook of statistical genetics*. Wiley, Chichester, 602-635.
- Pritchard, J.K., Seielstad, M.T., Perez-Lezaun, A. and Feldman, M.W. (1999) Population growth of human Y chromosomes: a study of Y chromosome microsatellites, *Mol. Biol. Evol.*, 16, 1791-1798.
- R_Development_Core_Team, R. (2007) A language and environment for statistical computing, RfS Vienna, Austria: Computing.
- Rousset, F. C. O., (2008) GenePop'007: a complete re-implementation of the GenePop software for Windows and Linux. *Mol. Ecol. Resources* 8, 103-106.
- Shannon, C.E. (1948) A mathematical theory of communication, *The Bell System Technical Journal*, 27, 379-423, 623-656.
- Sherwin, W.B., Jabot, F., Rush, R. and Rossetto, M. (2006) Measurement of biological information with applications from genes to landscapes, *Molecular Ecology*, 15, 2857-2869.
- Slatkin, M. (1991) Inbreeding coefficients and coalescent times, *Genet Res*, 58, 167-175.
- Tajima, F. (1989) Statistical method for testing the neutral mutation hypothesis by DNA polymorphism, *Genetics*, 123, 585-595.
- Wakeley, J. (1998) Segregating sites in Wright's island model, *Theor Popul Biol*, 53, 166-174.
- Wakeley, J. and Aliacar, N. (2001) Gene genealogies in a metapopulation, *Genetics*, 159, 893-905.
- Wakeley, J. and Hey, J. (1997) Estimating ancestral population parameters, *Genetics*, 145, 847-855.

- Watterson, G.A. (1975) On the number of segregating sites in genetical models without recombination, *Theoretical Population Biology*, 7, 256-276.
- Wilson, I.J. and Balding, D.J. (1998) Genealogical inference from microsatellite data, *Genetics*, 150, 499-510.
- Wright, S. (1950) Genetical Structure of Populations, *Nature*, 166, 247-249.

Annexes

It is possible to specify a population tree branching history in the prior files (*.prs) just by choosing a number. The following list presents all the possible topologies to choose in models with 3, 4 and 5 populations. The topologies are described as population joining events when looking from the present back. The colour code represents the order of joining: 1st – black; 2nd – blue; 3rd – green; 4th – red.

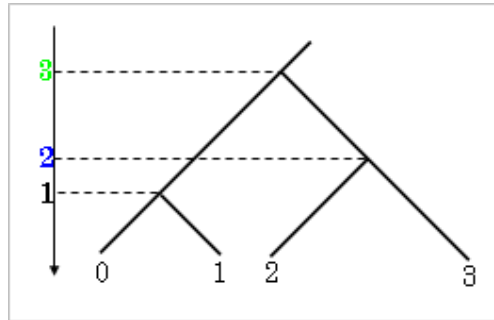


Figure 9 – Tree branching scheme number 13 for 4 populations

As an example consider topology 13 for 4 populations $[(0, 1), (2, 3)]$. The correspondent tree is represented in Figure 9.

<u>3 populations:</u>	<u>5 populations:</u>	25 - (((0, 1), 2), (3, 4))
1 - ((0, 1), 2)	1 - (((0, 1), 2), 3), 4)	26 - (((0, 1), 2), 4), 3)
2 - ((0, 2), 1)	2 - (((0, 2), 1), 3), 4)	27 - ((0, 1), 2), (3, 4))
3 - ((1, 2), 0)	3 - (((1, 2), 0), 3), 4)	28 - (((0, 4), 2), 1), 3)
	4 - (((0, 1), 3), 2), 4)	29 - ((1, 4), (0, 2)), 3)
<u>4 populations:</u>	5 - (((0, 2), 3), 1), 4)	30 - (((2, 4), 0), 1), 3)
1 - (((0, 1), 2), 3)	6 - (((0, 3), 2), 1), 4)	31 - ((3, 4), ((0, 2), 1))
2 - (((0, 2), 1), 3)	7 - (((0, 3), 1), 2), 4)	32 - (((0, 2), 4), 1), 3)
3 - (((1, 2), 0), 3)	8 - (((1, 2), 3), 0), 4)	33 - ((0, 2), (1, 4)), 3)
4 - (((0, 1), 3), 2)	9 - (((1, 3), 2), 0), 4)	34 - ((0, 2), 1), (3, 4))
5 - (((0, 2), 3), 1)	10 - (((1, 3), 0), 2), 4)	35 - (((0, 2), 1), 4), 3)
6 - (((0, 3), 2), 1)	11 - (((2, 3), 0), 1), 4)	36 - ((0, 2), 1), (3, 4))
7 - (((0, 3), 1), 2)	12 - (((2, 3), 1), 0), 4)	37 - ((0, 4), (1, 2)), 3)
8 - (((1, 2), 3), 0)	13 - (((0, 1), (2, 3)), 4)	38 - (((1, 4), 2), 0), 3)
9 - (((1, 3), 2), 0)	14 - (((0, 2), (1, 3)), 4)	39 - (((2, 4), 1), 0), 3)
10 - (((1, 3), 0), 2)	15 - (((0, 3), (1, 2)), 4)	40 - ((3, 4), ((1, 2), 0))
11 - (((2, 3), 0), 1)	16 - (((2, 3), (0, 1)), 4)	41 - ((1, 2), (0, 4)), 3)
12 - (((2, 3), 1), 0)	17 - (((1, 3), (0, 2)), 4)	42 - (((1, 2), 4), 0), 3)
13 - ((0, 1), (2, 3))	18 - (((1, 2), (0, 3)), 4)	43 - ((1, 2), 0), (3, 4))
14 - ((0, 2), (1, 3))	19 - (((0, 4), 1), 2), 3)	44 - (((1, 2), 0), 4), 3)
15 - ((0, 3), (1, 2))	20 - (((1, 4), 0), 2), 3)	45 - ((1, 2), 0), (3, 4))
16 - ((2, 3), (0, 1))	21 - (((2, 4), (0, 1)), 3)	46 - (((0, 4), 1), 3), 2)
17 - ((1, 3), (0, 2))	22 - ((3, 4), ((0, 1), 2))	47 - (((1, 4), 0), 3), 2)
18 - ((1, 2), (0, 3))	23 - (((0, 1), 4), 2), 3)	48 - ((2, 4), ((0, 1), 3))
	24 - (((0, 1), (2, 4)), 3)	49 - ((3, 4), (0, 1)), 2)
		50 - (((0, 1), 4), 3), 2)

51 - ((0, 1), 3), (2, 4)
52 - ((0, 1), (3, 4)), 2
53 - (((0, 1), 3), 4), 2
54 - ((0, 1), 3), (2, 4)
55 - (((0, 4), 2), 3), 1
56 - ((1, 4), ((0, 2), 3))
57 - (((2, 4), 0), 3), 1
58 - ((3, 4), (0, 2)), 1
59 - (((0, 2), 4), 3), 1
60 - ((0, 2), 3), (1, 4)
61 - ((0, 2), (3, 4)), 1
62 - (((0, 2), 3), 4), 1
63 - ((0, 2), 3), (1, 4)
64 - (((0, 4), 3), 2), 1
65 - ((1, 4), ((0, 3), 2))
66 - ((2, 4), (0, 3)), 1
67 - (((3, 4), 0), 2), 1
68 - (((0, 3), 4), 2), 1
69 - ((0, 3), 2), (1, 4)
70 - ((0, 3), (2, 4)), 1
71 - (((0, 3), 2), 4), 1
72 - ((0, 3), 2), (1, 4)
73 - (((0, 4), 3), 1), 2
74 - ((1, 4), (0, 3)), 2
75 - ((2, 4), ((0, 3), 1))
76 - (((3, 4), 0), 1), 2
77 - (((0, 3), 4), 1), 2
78 - ((0, 3), (1, 4)), 2
79 - ((0, 3), 1), (2, 4)
80 - (((0, 3), 1), 4), 2
81 - ((0, 3), 1), (2, 4)
82 - ((0, 4), ((1, 2), 3))
83 - (((1, 4), 2), 3), 0
84 - (((2, 4), 1), 3), 0
85 - ((3, 4), (1, 2)), 0
86 - ((1, 2), 3), (0, 4)
87 - (((1, 2), 4), 3), 0
88 - ((1, 2), (3, 4)), 0
89 - ((1, 2), 3), (0, 4)
90 - (((1, 2), 3), 4), 0
91 - ((0, 4), ((1, 3), 2))
92 - (((1, 4), 3), 2), 0
93 - ((2, 4), (1, 3)), 0
94 - (((3, 4), 1), 2), 0
95 - ((1, 3), 2), (0, 4)
96 - (((1, 3), 4), 2), 0
97 - ((1, 3), (2, 4)), 0
98 - ((1, 3), 2), (0, 4)
99 - (((1, 3), 2), 4), 0
100 - ((0, 4), (1, 3)), 2
101 - (((1, 4), 3), 0), 2
102 - ((2, 4), ((1, 3), 0))
103 - (((3, 4), 1), 0), 2
104 - ((1, 3), (0, 4)), 2
105 - (((1, 3), 4), 0), 2
106 - ((1, 3), 0), (2, 4)
107 - (((1, 3), 0), 4), 2
108 - ((1, 3), 0), (2, 4)
109 - ((0, 4), (2, 3)), 1
110 - ((1, 4), ((2, 3), 0))
111 - (((2, 4), 3), 0), 1
112 - (((3, 4), 2), 0), 1
113 - ((2, 3), (0, 4)), 1
114 - ((2, 3), 0), (1, 4)
115 - (((2, 3), 4), 0), 1
116 - ((2, 3), 0), (1, 4)
117 - (((2, 3), 0), 4), 1
118 - ((0, 4), ((2, 3), 1))
119 - ((1, 4), (2, 3)), 0
120 - (((2, 4), 3), 1), 0
121 - (((3, 4), 2), 1), 0
122 - ((2, 3), 1), (0, 4)
123 - ((2, 3), (1, 4)), 0
124 - (((2, 3), 4), 1), 0
125 - ((2, 3), 1), (0, 4)
126 - (((2, 3), 1), 4), 0
127 - ((0, 4), 1), (2, 3)
128 - ((1, 4), 0), (2, 3)
129 - ((2, 4), 3), (0, 1)
130 - ((3, 4), 2), (0, 1)
131 - (((0, 1), 4), (2, 3))
132 - ((0, 1), ((2, 4), 3))
133 - ((0, 1), ((3, 4), 2))
134 - (((0, 1), 4), (2, 3))
135 - ((0, 1), ((2, 3), 4))
136 - (((0, 4), 2), (1, 3))
137 - ((1, 4), 3), (0, 2)
138 - ((2, 4), 0), (1, 3)
139 - ((3, 4), 1), (0, 2)
140 - (((0, 2), 4), (1, 3))
141 - ((0, 2), ((1, 4), 3))
142 - ((0, 2), ((3, 4), 1))
143 - (((0, 2), 4), (1, 3))
144 - ((0, 2), ((1, 3), 4))
145 - (((0, 4), 3), (1, 2))
146 - ((1, 4), 2), (0, 3)
147 - ((2, 4), 1), (0, 3)
148 - ((3, 4), 0), (1, 2)
149 - (((0, 3), 4), (1, 2))
150 - ((0, 3), ((1, 4), 2))
151 - ((0, 3), ((2, 4), 1))
152 - (((0, 3), 4), (1, 2))
153 - ((0, 3), ((1, 2), 4))
154 - (((0, 4), 1), (2, 3))
155 - ((1, 4), 0), (2, 3)
156 - ((2, 4), 3), (0, 1)
157 - ((3, 4), 2), (0, 1)
158 - ((2, 3), ((0, 4), 1))
159 - ((2, 3), ((1, 4), 0))
160 - ((2, 3), 4), (0, 1)
161 - ((2, 3), ((0, 1), 4))
162 - ((2, 3), 4), (0, 1)
163 - (((0, 4), 2), (1, 3))
164 - ((1, 4), 3), (0, 2)
165 - ((2, 4), 0), (1, 3)
166 - ((3, 4), 1), (0, 2)
167 - ((1, 3), ((0, 4), 2))
168 - ((1, 3), ((2, 4), 0))
169 - ((1, 3), 4), (0, 2)
170 - ((1, 3), ((0, 2), 4))
171 - ((1, 3), 4), (0, 2)
172 - (((0, 4), 3), (1, 2))
173 - ((1, 4), 2), (0, 3)
174 - ((2, 4), 1), (0, 3)
175 - ((3, 4), 0), (1, 2)
176 - ((1, 2), ((0, 4), 3))
177 - ((1, 2), 4), (0, 3)
178 - ((1, 2), ((3, 4), 0))
179 - ((1, 2), 4), (0, 3)
180 - ((1, 2), ((0, 3), 4))