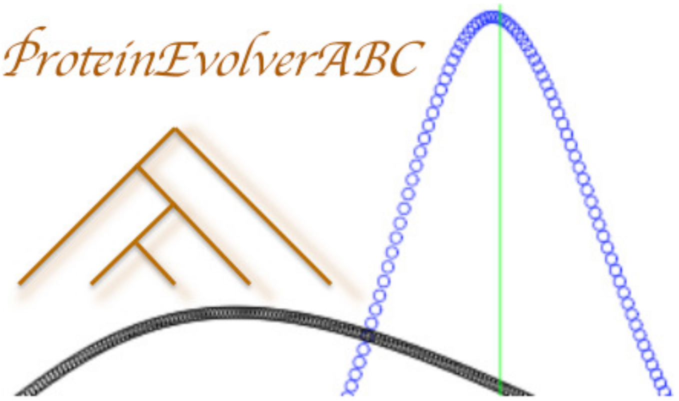
Documentation for **ProteinEvolverABC**

Estimation of Recombination and Substitution rates in alignments of protein sequences by approximate Bayesian computation

Current version is 1.0

© 2020. Miguel Arenas
miguelmmmab@gmail.com - marenas@uvigo.es

September 1, 2020



Contents

Disclaimer

Credits

This program was developed at,

- *Department of Biochemistry, Genetics and Immunology, University of Vigo, 36310 Vigo, Spain.*
- *Biomedical Research Center (CINBIO), University of Vigo, 36310 Vigo, Spain.*
- *Galicia Sur Health Research Institute (IIS Galicia Sur), 36310 Vigo, Spain.*

This work was supported by the Spanish Ministry of Economy and Competitiveness and Ministry of Science and Innovation through the Grants [RYC- 2015-18241] and [PID2019-107931GA-I00/AEI/10.13039/501100011033] to MA.

1. Purpose

ProteinEvolverABC is an evolutionary framework to estimate recombination and substitution rates by approximate Bayesian computation (ABC) from protein sequence alignments. *ProteinEvolverABC* is designed to be run either on Linux OS or Mac OSX and it is freely available from https://github.com/MiguelArenas/ProteinEvolverABC.

The user can specify prior distributions for the parameters to estimate (i.e. recombination and substitution rates), but also for other evolutionary parameters that will be treated as nuisance parameters, like amino acid frequencies, proportion of invariable sites (+I) or heterogeneity change across sites according to a gamma distribution (+G). A graphical user interface (GUI) *ProteinEvolverABC_GUI* is also available, which allows for a user-friendly procedure to specify the input settings of *ProteinEvolverABC*.

The computer simulations are performed via the coalescent program *ProteinEvolverProtABC*, an adapted version of the simulator *ProteinEvolver* (https://github.com/MiguelArenas/proteinevolver) [1] to ABC.

ProteinEvolverABC implements a total of 16 summary statistics, some of them have been shown able to extract information from DNA data to accurately estimate the parameters of interest [2, 3]. Conveniently, *ProteinEvolverABC* can run the simulations on parallel, according to user specifications, to save computer time.

Different ABC estimation methods are implemented in *ProteinEvolverABC*, in particular a rejection algorithm [4] and a weighted multiple linear regression algorithm as suggested in [5]. For both algorithms the user can specify the number of simulations to use, the number of simulations to retain (tolerance of the rejection algorithm) and which summary statistics to use (although I recommend applying all the implemented summary statistics). For the regression algorithm, the user can also specify whether to transform the parameters and which transformation to use (log-transformation or logit-transformation), and whether to correct for the heteroscedastic variance of the parameters. *ProteinEvolverABC* provides also several diagnostic plots to assess qualitatively how well the model fitted the data, namely: histograms of the summary statistics of the retained simulations; scatterplots of the summary statistics and each parameter; and a scatterplot of the two principal components calculated from the summary statistics of the simulations. In all the plots the values for the target protein alignment data are superimposed.

1.1. Parameters to estimate

The parameters that can be estimated by *ProteinEvolverABC* are recombination and substitution rates. Bellow follow their technical details.

The population recombination rate per site (amino acid site) per generation (ρ) is described by the following equation
$\rho = 2zNrl$,
where z = 1 or 2 if haploid or diploid, N is the effective population size, r is the recombination rate per site and l is the sequence length in amino acids.
In nature, ρ commonly ranges between 0 (no recombination) and 100 [e.g., 2, 6, 7]. Given ρ and the number of sequences one can obtain the expected number of recombination events by $R[R(n)] = Rho\sum_{j=1}^{n-1}\frac{1}{j}$ where n is the number of sequences.
For example, for 10 sequences, 0, 2.72, 10.87, 43.49 and 86.97 are the number of recombination events expected for ρ values of 0, 1, 4, 16 and 32, respectively. Importantly, ρ is frequently estimated from DNA sequences [e.g., 2, 3, 8, 9-11] and hence a goal of this framework is the estimation of ρ in protein sequences.

The population amino acid substitution rate per site per generation (θ) is described by the following equation
$\theta = 2zN\mu L$,
where z = 1 or 2 if haploid or diploid, N is the effective population size, μ is the substitution rate per amino acid and L is the sequence length in amino acids. Common values for θ are between 50 and 300 [e.g., 2, 12, 13]. A θ = 0 means that there are not substitutions (fixed mutations) in the alignment. As an example, for 25 sequences, evolved without recombination, a θ = 50, 100 and 200 can provide alignments with sequence identity of 0.92, 0.84 and 0.73, respectively.

2. Graphical User Interface

ProteinEvolverABC includes a user-friendly GUI, written in java, that helps to create the main input file (Settings.txt) to run the analysis. The GUI consists of a series of tabs based on user specifications.

- Settings specified within each tab must be saved by clicking on the “Save and next” button, which is placed at the right-bottom of the tab. This button will also move the process towards a new tab. The user can also move among opened tabs, but notice that

new settings will only be saved by clicking in the "Save and next" button of the modified tab.
- Settings must be specified carefully, so that to have biological meaning. The GUI will check for incorrect methodological settings (e.g., it does not allow negative substitution rates), but it will not evaluate the settings in terms of biological meaning. Importantly, errors in *ProteinEvolverABC* may be caused by incorrect user-specified settings or user-specified settings without biological meaning.
- At the end of the user specifications (last tab), a file "Settings.txt" can be written by clicking on the "Generate Settings.txt file" button. This file is the input of the *ProteinEvolverABC* pipeline (Section 4.1). In addition, a shell script to run *ProteinEvolverABC* into a single step can also be generated.

Tab "General Settings": The user must upload a file with a multiple alignment of protein sequences in phylip format (see Section 4). When selecting the alignment file for uploading, the file may appear in grey but that is not a problem, just select the file. The number of simulations, how to consider indels (gaps), number of processors, option to save simulations and option to show information about the simulations on the screen (section 4) must be specified too. When everything is specified just press the button "Save and Next".

Tab "Evolutionary History": This tab includes settings for coalescent simulations (e.g., population size). Settings on the left must be specified while settings on the right are optional. Note that the sample size and sequence length are given by the alignment uploaded in the previous tab. The effective population size by default is 1,000, it can be modified (a window would appear clicking in "fix"). The recombination rate per site is a parameter to be estimated and its prior distribution can be selected from a variety of available distributions, also the values of the parameters of the selected distribution must be specified. The equivalent values of ρ (Rho) are printed by the GUI after their specification. See details about the settings in Section 4.

Tab "Substitution process": The substitution rate per site (parameter to be estimated) must be introduced selecting a prior distribution and indicating the he values of the parameters of the selected distribution. The equivalent values of θ (Theta) are printed by the GUI after their specification. An empirical substitution model of protein evolution (i.e., previously identified with *Prottest* [14]) must be selected. Other capabilities that are optional include: customizing amino acid frequencies, considering a proportion of invariable sites (+I), and considering heterogeneity rate across sites according to a gamma distribution (+G); these parameters can be fixed but also can vary among simulations according to user-specified prior distributions (nuisance parameters). Further details in Section 4.

Tab "ABC Settings": In this tab the settings for the ABC estimation must be specified, namely: number of simulations to consider; number of simulation to retain; choosing the summary statistics to consider (I recommend applying all of them); performing or not the multiple linear regression algorithm; transforming or not the parameters before the regression-step; correcting or not the estimation for the heteroscedescity of the parameters. Further details in Section 4.

Tab "Make input file": Finally, the "Settings.txt" file (main input file of *ProteinEvolverABC*) can be generated and placed in the working directory. Optionally, a shell script can also be generated by which the entire *ProteinEvolverABC* can be executed.

The specification of each prior distribution will open a new window where the parameter values for such a distribution can be introduced.

Notice that the GUI is not required to execute *ProteinEvolverABC* and the user can create his/her own Settings file manually. For example, advanced users might generate a large number of different Settings files without using the GUI (e.g., through a script). On the other hand, the GUI is recommended for beginners.

3. Software required for ProteinEvolverABC

3.1. Software installations

The *ProteinEvolverABC* framework runs on the command line and requires that Perl and R are installed. Perl can be downloaded from http://www.perl.org/ and R from http://www.r-project.org/. *ProteinEvolverABC* also requires the following R libraries: *lattice*, *MCMCpack*, *ape*, *graphics* and *abc*. These libraries can be installed by the command: install.packages("*library_name*"), see further details in http://cran.r-project.org/doc/manuals/R-admin.html#Installing-packages. For example,

```
install.packages("lattice")
```

The folder "source" contains the source codes of the simulator *ProteinEvolverProtABC*, the software *PhiTest* [15] and the GUI. The first two programs are compiled with the compiler *gcc* and the GUI requires Java (javac, jar) for its compilation.
The folder "scripts" contains several scripts required to execute *ProteinEvolverABC*.
These two folders must not be modified by the user as they are required to run the program.

3.2. Execute ProteinEvolverABC

First, *ProteinEvolverABC* must be compiled for the OS through a Makefile that is placed on the main directory,

```
make all
```

With this specification both *ProteinEvolverABC* and *PhiTest* will be compiled and the executable files will be placed in the folder "*bin*". If the GUI cannot be compiled (i.e., the used OS does not have Java compilers installed), it is not a problem for running *ProteinEvolverABC*, just the user has to make the Settings file by hand (adapting the Settings file of an example provided with the package is recommended).

The execution of *ProteinEvolverABC* consists of the following 4 separated steps.
Importantly, the input files (Settings.txt and the multiple alignment of protein sequences) must be in the same directory where the main *ProteinEvolverABC* scripts are present.

1. Read input files, **compute summary statistics for the input alignment** and **sample from the prior distributions to later perform the simulations**. Type,

```
perl ProteinEvolverABC_Phase1.pl Settings.txt
```

2. Launch simulations (can run in parallel). Type,

```
perl ProteinEvolverABC_Phase2.pl Settings.txt
```

3. Compute summary statistics for the simulated data. Type,

```
perl ProteinEvolverABC_Phase3.pl Settings.txt
```

4. Compute estimations of substitution and recombination rates (posterior distributions), provides information about the summary statistics and posterior distributions. This execution can be repeated with different ABC settings (see section 4.5). Type,

```
perl ProteinEvolverABC_Phase4.pl Settings.txt
```

Where "Settings.txt" is the main input file with the user-specified settings (see section 4.1). For each step, running information is written on the screen (see section 4.3 for details).

A full execution may take time. In particular, step 2 may take a considerable amount of time especially when performing a large number of simulations, when the protein dataset includes a large number and long sequences, or when considering high recombination rates. However, the step 2 can be run in parallel on a Linux environment if the user specifies this in the settings file and works with a machine presenting many processors. If simulation settings are unexpected (e.g., too high recombination rates) the analysis may fail (see section 4.7 about “Message Errors”). Indeed, this step 2 generates a number of simulated sequence alignments in the working directory (in the step 3 they can be removed or saved as a zip file depending on user specifications).

4. *ProteinEvolverABC* Usage

The main input file of the program is the Settings file, where the user must specify settings such as prior distributions for evolutionary parameters or the estimation method (see section 4.1).

A standard run of *ProteinEvolverABC* framework requires the following files from the user:

- **Settings file**. This input file must contain all the specifications for the estimation. It should be made by the user (i.e., modifying a Settings.txt file provided in the examples or by creating a Settings.txt file with the GUI) with the desired specifications (details below). Note that the name of this file must be considered to run *ProteinEvolverABC* (e.g., *perl ProteinEvolverABC_Phase1.pl Settings.txt*).
- **Multiple alignment of protein sequences**. The protein sequence alignment that will be analyzed must be provided by the user in sequential phylip format (.phy). **This file should be placed in the Settings file directory and its name should be specified in the Settings file.** See details in section 4.2.

Examples of a Settings and multiple sequence alignment files for a standard *ProteinEvolverABC* run are provided in the package (folder “examples”).

The user is advised to create a folder in his/her desired location (e.g. his own home folder) and copy there a Settings file, multiple alignment files and all *ProteinEvolverABC* material (including executable files and folders).

After concluding a full run of *ProteinEvolverABC* two folders are created in the working directory, the “MainOutputs” and the “OtherOuputs” folders (see details in section 4.4). Since the estimation phase is quite fast in comparison with the simulation phase, the user is advised to explore different settings for the ABC method (see section 4.5 in "Re-analyzing data") if desired, without having to run again the simulation phases.

4.1. The Settings input file

The Settings input file must contain all the information required to perform the analysis and should be carefully checked by the user since mistakes may alter results (error messages will be displayed on the screen and may suggest stop the execution by typing CTRL+C).

The file consists of two main blocks, the simulation phase and the estimation phase.

Important notes when dealing with the Settings file

- Parameter values must be introduced in the line after the parameter description, otherwise such parameter will be considered as "not specified" (see examples below).
- Do not modify the parameter description (it is used by the program to identify the parameter).
- Some parameters are mandatory and must be specified, these parameters contain an "*" (see below).
- Some parameters require a prior distribution (see section 4.1.1), this is a distribution within which the estimation should fall (the range of the prior distribution should include the estimation; for example, if the user believes that the recombination rate can be 0 then the prior distribution of the recombination rate should include 0).
- Optional parameters (or text that is not read by the program) can be "turned-off" by having an "#" at the beginning of the line.

4.1.1. Available prior distributions

Several distributions are included in *ProteinEvolverABC* to simulate protein data under different evolutionary scenarios (Table 1). In addition, most of them can be truncated at lowest and highest values. However, note that some parameters must be specified by an integer number (e.g., generation time) while others by a non-integer number (e.g., amino acid frequencies). Therefore, not all distributions can be applied to any parameter. Details for each particular parameter are described in the following subsections. Indeed, several distributions can be truncated at both sides of the distribution with user-specified values (Table 1).

Table 1. Available distributions in *ProteinEvolverABC*.

Distribution	Description	Truncated	Examples
fix	Fixed value (integer or non-integer)	n.a.	fix 4; fix 0.7
Uniform (uniform)	Random between two values (integer or non-integer): lowest highest	n.a.	uniform 1.0e-8 1.0e-5; uniform 2e-9 5e-6; uniform 0 3
Normal (norm)	Normal distribution (mean, sd)	t # #	norm 1.0e-8 1.0e-5; norm 1.0e-8 1.0e-5 t 1.0e-9 1.0e-6
Exponential (exp)	Exponential distribution (rate)	t # #	exp 1.0e-7; exp 1.0e-7 t 1.0e-8 1.0e-6

gamma	Gamma distribution (shape, rate “1/scale”)	t # #	gamma 1.0e-7 5.0e-7; gamma 1.0e-7 5.0e-7 t 2.0e-7 1.0e-6
beta	Beta distribution (shape1, shape 2)	t # #	beta 1.0e-7 5.0e-7; beta 1.0e-7 5.0e-7 t 2.5e-7 1.0e-6
dirichlet	Dirichlet distribution (alpha “vector”)	n.a.	dirichlet 1 1 1 1; dirichlet 1 1 1 1 1 1

4.1.2. Simulation phase

General settings for the simulation

- **Name of the file with the target protein sequences alignment.** This specification is mandatory. Only specify the filename (i.e. no pathway) since the target alignment file needs to be placed in the same pathway as the Settings file. See section 4.2 concerning the format of the multiple sequences alignment. For example,

```
 ### Target alignment file ###         # phylip format
*NameOfPhylipFile= ProtSeq1.phy
```

- **Number of simulations.** This parameter is mandatory. I recommend at least 20,000 computer simulations but with 50,000 simulations results should be accurate. With 100,000 simulations results could be even better and so on. However, the number of simulations required to obtain accurate estimates depends on many factors, especially the target multiple sequence alignment. More complex sequences (large molecular diversity) could require more simulations because the parametric landscape could be more irregular and should be more sampled. For example,

```
### Total number of simulations ###
*NumberOfSimulations=250
```

- **Consideration of indels.** This parameter is mandatory. The user specifies if indels (gaps) should be ignored (by default and recommended) or considered as a new state. This decision can affect the summary statistics if there are indels in the multiple sequence alignment. For example,

```
# Consideration of indels. 0 (indels are ignored), 1 (indels are considered as a new state)
*Indels=0
```

- **Number of processors to run the simulations in parallel on a machine.** This parameter is optional (by default simulations run on only 1 processor). This parallelization works on machines using Linux OS with shared memory. Ideally, one should specify at the most the number of available processors of the machine. In a Linux OS the number of available processors can be provided by,

```
grep -c ^processor /proc/cpuinfo
```

The number of processors must be specified in the settings file. For example,

```
# Number of available processors to run the simulations in parallel. 1 by default
NumberOfProcessors=8
```

- **Save simulated data.** This parameter is mandatory. “0” the simulated data is not saved (recommended, simulated data may require a lot of space in the hard disk of user’s computer), “1” the simulated data is saved. Note that even if choosing not to save simulated data, simulated data is created and saved temporally since this data is required for calculating the Summary Statistics in the following step. For example,

```
# Save simulated data. 0 (No), 1 (Yes, but it requires space in the disk)
* SaveSimulations=0
```

- **Show running information.** This parameter is mandatory. "0" reduces the amount of information printed on the screen during the simulation and computation of summary statistics phases. "1" show all the information printed on the screen during the simulation and computation of summary statistics phases. Applying "0" or "1" does not change the final results, it only affects to the amount of information shown on the screen during the execution. It is recommended specify "0" to save computer time (printing information on the screen requires more computer time). For example,

```
# Show running information (simulations and summary statistics) on the screen.
0 (No), 1 (Yes, but it slows down the running time)
*ShowInformationScreen=0
```

Demographic settings

- **Haploid/diploid simulated data.** Haploid is defined with a value of 1 and diploid with a value of 2. This parameter is mandatory. For example,

```
# Haploid or Diploid data (haploid=1, diploid=2)
*Haploid/Diploid=2
```

- **Effective population size (*N*).** The parameter value must be an integer. This parameter is mandatory. For example,

```
# Population size (i.e., 1000).
*PopulationSize=1000
```

Longitudinal sampling

- **Sampling at different times.** This parameter is optional. The user can specify the time at which the tip nodes of the tree (i.e. samples) were sampled in years. In the example, 4 sampling times are specified: sampled in 1995 - sequences 1 to 10; sampled in 2003 - sequences 11 to 16; sampled in 1997 - sequences 17 to 26; sampled in 2001: sequences 7 and 8. Note that this option does not work if a deme converges, backwards in time, before the last sampling time. For example,

```
# Logitudinal sampling. Requires GenerationTime.
DatedTips=4 1995 1 10 2003 11 16 1997 17 26 2001 27 29
```

- **Generation time.** This parameter is optional. The user can specify the time for each generation. The parameter value can be fixed (fix) or sampled from a uniform distribution (uniform). For example,

```
# Generation time. fix, uniform; i.e., uniform 500 1000
GenerationTime=fix 1200
```

Recombination

- **Recombination rate per site.** This parameter is mandatory and will be estimated. Distributions allowed: fix, uniform, norm(t), exp(t), gamma(t), beta(t).

Important: Note that ρ (see section 1.1) should not be higher than 150, or there might be out-of-memory errors during simulation resulting from considering "infinite" nodes in the ancestral recombination graphs (ARGs [16, 17]). For example,

```
#Recombination rate per site. fix, uniform, gamma, beta, normal, exponential;
i.e., gamma 0.02 0.5 t 1.5e-07 9.3e-07. -PARAMETER TO BE ESTIMATED-
*RecombinationRate=uniform 0 5.56e-6
```

Protein evolution and substitution model

- **Amino acid substitution rate per site.** This parameter is mandatory and will be estimated. Distributions allowed: fix, uniform, norm(t), exp(t), gamma(t), beta(t). For example,

```
# Amino acid substitution rate. i.e., fix 7.0e-6. -PARAMETER TO BE ESTIMATED-
*SubstitutionRate=uniform 0 1.67e-4
```

- **Model of amino acid substitution.** The user must specify one of the following empirical substitution model of protein evolution [18, 19]: Blosum62, CpRev, Dayhoff, DayhoffDCMUT, HIVb, HIVw, JTT, JonesDCMUT, LG, Mtart, Mtmam, Mtrev24, RtRev, VT, WAG, UserEAAM (the latter is a user-defined model that must be provided as an additional input file with name UserEAAM, the format of this file is explained with details in the documentation of the simulator *ProteinEvolver* [1]). This parameter is mandatory. For example,

```
# Model of amino acid substitution (i.e., Blosum62, CpRev, Dayhoff,
DayhoffDCMUT, HIVb, HIVw, JTT, JonesDCMUT, LG, Mtart, Mtmam, Mtrev24, RtRev,
VT, WAG, UserEAAM)
*SubstitutionModel=JTT
```

- **Amino acid frequencies.** Frequencies for each amino acid site along sequences. Distributions allowed: fix or dirichlet. This parameter is optional, and if not specified the program will assume equally distributed frequencies (all fix with value 0.05). For example,

```
# Amino acid frequencies. fix or dirichlet. By default equally distributed
frequencies. i.e., dirichlet 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*AminoacidFrequencies=fix 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
```

- **Rate of heterogeneity across sites (+G).** Distributions allowed: fix, uniform, norm(t), exp(t), gamma(t), beta(t). This parameter is optional. For example,

```
# Rate of heteregeneity across sites, +G. fix, uniform, gamma, beta, normal,
exponential; i.e., fix 0.6
RateHetSites=uniform 0.76 0.90
```

- **Proportion of invariable sites (+I).** Distributions allowed: fix, uniform, norm(t), exp(t), gamma(t), beta(t). This parameter is optional. For example,

```
# Proportion of invariable sites, +I. fix, uniform, gamma, beta, normal,
exponential; i.e., exponential 0.002 t 0 1.0
#PropInvSites=uniform 0.3 0.5
```

4.1.3. Estimation phase

All these parameters are mandatory. The estimation phase (through *ProteinEvolverABC_Phase4.pl*) can be repeated any number of times allowing to perform estimations under different simulated data and analytical methods (see details in section 4.5) without having to repeat the phases 1-3.

ABC estimation

- **ABC iterations.** Number of simulations to consider in the ABC analysis. Unless testing the methodology, the user is advised to use all the simulations performed. Note that the number of simulations to use has to be obviously equal or less to the number of simulations performed (specified in *NumberOfSimulations*). For example,

```
#ABC iterations. Number of simulations to consider (Iterations <=
NumberOfSimulations)
*ABCIterations=250
```

- **ABC tolerance.** Number of simulations closest to real data to retain in the ABC procedure. Note that the tolerance considered has to be obviously less than the total number of simulations (specified in *NumberOfSimulations*). A tolerance of 100 can be enough but the best tolerance varies among multiple sequence alignments. It is recommended to explore different values of this parameter. For example,

```
 #ABC tolerance. Number of simulations closest to real data to retain in the
ABC procedure (Tolerance < NumberOfSimulations)
*ABCTolerance=50
```

- **ABC method.** The ABC algorithm has to be specified. The user can choose either to perform only the rejection algorithm, by choosing the option "rejection", or to perform both the rejection and the multiple linear regression algorithms, by choosing the option "loclinear". For example,

```
 #ABC method (rejection, loclinear).
*ABCMethod=loclinear
```

- **ABC transformation.** This parameter allows the user to choose which, if any, transformation will be performed to the data before adjusting the values with the regression algorithm. The user can choose not to perform any transformation (option "none", to perform a logarithm transformation (option "log"), or to perform a logit transformation (option "logit"). Note that even if choosing to perform only the rejection algorithm the user has to specify this parameter (although it will not be used). For example,

```
 #ABC transformation (none, log, logit).
*ABCTransf=logit
```

- **ABC correction.** This parameter allows the user to choose to correct for the heteroscedasticity of the data when performing the regression adjustment. The user chooses to perform or not this corrections by specifying option "1" or "0", respectively. For example,

```
#ABC correction. Correct for heteroscedastic variance (no=0, yes=1)
*ABCHCorr=1
```

- **Summary statistics to use.** The user needs to choose which summary statistics to use for the ABC estimation by specifying their numeric identifiers (see section 5 "Models and Methods" for further details). It is recommended to specify the 16 summary statistics implemented in the program, since the evaluation of the program was done using these 16 summary statistics. For example,

```
#Summary statistics to use. See documentation for details
*SummaryStatistics= 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

Graphical options

- **Multiple pages.** This setting specifies if the plots produced by *ProteinEvolverABC* estimation phase will be presented in single-page PDF documents (option "0") or in multiple-pages PDF documents (option "1"). The last option is advised since it reduces the number of files produced (see section 4.4 "Output Files" for details). For example,

```
#Multiple pages. PDF documents with multiple pages (no=0, yes=1)
*MultiPage=1
```

4.2. The target alignment of protein sequences

The name of the file with the multiple alignment of protein sequences to be analyzed must be indicated in the Settings File (specified in *NameOfPhylipFile*). Importantly, this file must be placed in the directory of the settings file and only its filename should be specified (i.e. do not include its pathway).

The alignment must be presented in *standard phylip sequential* format. In particular, the first line must indicate the number of sequences and the sequence length (in amino acids). The following lines must indicate a taxa name (10 characters) followed by the sequence (20 amino acids in one letter code or indels as "-"). A variety of examples are provided in the *ProteinEvolverABC* package (in the folder "examples").

Example,

```
5 15
seq000001 AYPFQLGFQDATSPI
seq000002 AYPLQLGFQDATSPI
seq000003 AYPFQLGFQDAASPI
seq000004 AYPLQLGFQDATSPI
seq000005 AYPLQLGFQDAVSPI
```

In order to avoid errors generating the correct format, I recommend use the format alignment converter *ALTER* (http://sing.ei.uvigo.es/ALTER/) [20]. Therein after uploading your input alignment, go to "*Select program*" and in the section "*Population genetics*" please select "*CodABC*". Then select "*PHYLIP*" and finally, covert.

4.3. Screen information

The information written on the screen during an execution example is shown below.

-> Let us be in the **parent directory** of the *ProteinEvolverABC* package, I copied there the settings and multiple sequence alignment files of the example "Example1rapid" provided with the package,

```
ProteinEvolverABC_proof$ ls
Makefile     ProteinEvolverABC_GUI.sh  ProteinEvolverABC_Phase4.pl
       documentation      ProtSeq1.phy       ProteinEvolverABC_Phase1.pl
       README.html        license       ProteinEvolverABC_1.0_examples
       ProteinEvolverABC_Phase2.pl      Settings.txt             scripts
       ProteinEvolverABC_AllPhases.sh   ProteinEvolverABC_Phase3.pl      bin
       source
```

-> **Compile *ProteinEvolverABC*** (it will compile *Phi*, *CoalEvolProteinEvolverABC* and the GUI),

```

ProteinEvolverABC_proof$ make clean
Removing executables ..
/Applications/Xcode.app/Contents/Developer/usr/bin/make  -C  source/PhiPack/src
clean
rm -f *.o
rm -f Phi
rm -f Profile
/Applications/Xcode.app/Contents/Developer/usr/bin/make                      -C
source/ProteinEvolverProtABC clean
Removing object and executable files to save space
Finished cleanup.

/Applications/Xcode.app/Contents/Developer/usr/bin/make                      -C
source/ProteinEvolverABC_GUI clean
rm -f *.class
rm -f *.jar
Done!
```

```
ProteinEvolverABC_proof$ make all
Compiling Phi ..
/Applications/Xcode.app/Contents/Developer/usr/bin/make  -C  source/PhiPack/src
clean
rm -f *.o
rm -f Phi
rm -f Profile
/Applications/Xcode.app/Contents/Developer/usr/bin/make  -C  source/PhiPack/src
Phi
gcc -c  -O3  -Wall normal.c -o normal.o
gcc -c  -O3  -Wall stats.c -o stats.o
gcc -c  -O3  -Wall maxChi.c -o maxChi.o
gcc -c  -O3  -Wall main.c -o main.o
In file included from main.c:17:
./normal.h:1:9: warning: 'NORM' is used as a header guard here, followed by
#define of a different macro [-Wheader-guard]
#ifndef NORM
        ^~~~
./normal.h:2:9: note: 'NORMAL' is defined here; did you mean 'NORM'?
#define NORMAL
        ^~~~~~
        NORM
1 warning generated.
gcc -c  -O3  -Wall queue.c -o queue.o
gcc -c  -O3  -Wall graphCode.c -o graphCode.o
gcc -c  -O3  -Wall fasta.c -o fasta.o
gcc -c  -O3  -Wall phylip.c -o phylip.o
gcc -c  -O3  -Wall mem.c -o mem.o
gcc -c  -O3  -Wall misc.c -o misc.o
gcc -c  -O3  -Wall pairScore.c -o pairScore.o
gcc -c  -O3  -Wall seqManip.c -o seqManip.o
gcc -c  -O3  -Wall global.c -o global.o
gcc  -O3   -Wall   -o  Phi  normal.o  stats.o  maxChi.o  main.o  queue.o  graphCode.o
fasta.o phylip.o mem.o misc.o pairScore.o seqManip.o global.o   -lm
cp Phi ../
cd ppma_2_bmp && /Applications/Xcode.app/Contents/Developer/usr/bin/make
make[2]: Nothing to be done for `default'.
cp ppma_2_bmp/ppma_2_bmp ../
Done!


Compiling ProteinEvolverProtABC ..
/Applications/Xcode.app/Contents/Developer/usr/bin/make                       -C
source/ProteinEvolverProtABC clean
Removing object and executable files to save space
Finished cleanup.
SEVERAL MINOR WARNINGS CAN BE PRINTED HERE, THEY CAN BE IGNORED
10 warnings generated.
gcc  -lm -O3 -Wall -o ProteinEvolverProtABC1.2.0 ProteinEvolverProtABC1.2.0.o
Finished compiling.

Done!


Compiling ProteinEvolverABC_GUI ..
/Applications/Xcode.app/Contents/Developer/usr/bin/make                       -C
source/ProteinEvolverABC_GUI clean
rm -f *.class
rm -f *.jar
/Applications/Xcode.app/Contents/Developer/usr/bin/make                       -C
source/ProteinEvolverABC_GUI all
javac ProteinEvolverABC_GUI.java
jar cmf ProteinEvolverABC_GUI.txt ProteinEvolverABC_GUI.jar *.class
rm -f *.class
Done!
```

```
Note that R will require the following libraries: lattice, MCMCpack, ape,
graphics, abc
These libraries can be installed from R by typing:
install.packages(lattice)
install.packages(MCMCpack)
install.packages(ape)
install.packages(graphics)
install.packages(abc)
See the documentation for additional details about ProteinEvolverABC
Compilation completed!
```

Notice, at the end, the reminder about the required R libraries (lattice, MCMCpack, ape, graphics, abc). These R libraries must be installed in the OS.
Executable files will be placed in the "*bin*" folder from where *ProteinEvolverABC* will use them,

```
ProteinEvolverABC_proof$ cd bin
ProteinEvolverABC_proof$ cd ls
Phi    ProteinEvolverABC_GUI.jar ProteinEvolverProtABC1.2.0       bin_Linux
       bin_MacOS_10.10.5
```

Note that executable files for some OS (Linux and MacOS 10.10.5) are included in folders present in the folder "*bin*", to apply them just copy the corresponding executable files to the first level of the folder "*bin*".

-> Now one can run *ProteinEvolverABC* with **the Settings and multiple sequence alignment files in the working directory**,

```
ProteinEvolverABC_proof$ cd ..
ProteinEvolverABC_proof$ ls
Makefile     ProteinEvolverABC_GUI.sh  ProteinEvolverABC_Phase4.pl
       documentation       ProtSeq1.phy ProteinEvolverABC_Phase1.pl
       README.html  license      ProteinEvolverABC_1.0_examples
       ProteinEvolverABC_Phase2.pl      Settings.txt       scripts
       ProteinEvolverABC_AllPhases.sh   ProteinEvolverABC_Phase3.pl      bin
       source
```

-> Step 1 of *ProteinEvolverABC* is executed with *ProteinEvolverABC_Phase1.pl* and indicating the settings file. First the script checks the directories, files and the OS in use,

```
ProteinEvolverABC_proof$ perl ProteinEvolverABC_Phase1.pl Settings.txt

******************************************************************************
**********************************
ProteinEvolverABC_Phase1.pl
- Read Settings file
- Compute the summary statistics for the target data
- Analyze the evolutionary scenario
- Compute the prior distributions (requires R with libraries: lattice,
MCMCpack, ape, graphics)
******************************************************************************
**********************************

> Input file uploaded: Settings.txt

> ProteinEvolverABC directory detected: .

> Settings directory detected: .

> OS detected: darwin (Mac OS X):
  darwin
  darwin-thread-multi-2level

> R output file created: "./MakePriorsProteinEvolverABC.r" ...
> Reading Settings from input main file ...
  Target alignment file: ProtSeq1.phy
    ./ProtSeq1.phy was detected... Ok!
  Number of simulations: 250
  Indels (gaps) are ignored
  Number of processors to run the simulations: 1
  Simulated data will not be saved
  Running information (simulations and summary statistics) is not shown on the
screen
  Haploid/diploid: 2
  Population size according to: fix 1000
  Recombination rate according to: uniform 0 5.56e-6
  Substitution rate according to: uniform 0 1.67e-4
  Substitution model: JTT
  Amino acid frequencies (1x20): fix 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
```

Afterwards, the script computes the summary statistics of the target multiple sequence alignment (more information about the summary statistics of the input multiple sequence alignment can be shown on the screen if the user has previously activated the option *ShowInformationScreen=1* in the settings file),

```
> Reading input protein sequences file ...
 Reading input file "./ProtSeq1.phy"...
 ProtSeq1.phy with sample size 35 and length 604 amino acids

> Computing summary statistics from input protein sequences file ..
  Target alignment file: ProtSeq1.phy
  "./ProtSeq1.phy" was detected... Ok!
  ProtSeq1.phy with sample size 35 and length 604 amino acids

> Calculating summary statistics from the target data set ..
SSs dataset: "./ProtSeq1.phy"Summary statistics for the target data were
successfully computed!
```

Then the perl script creates an R script to compute the prior distributions defined in the settings file (specification error in the Settings file is checked at this point),

```
> Writing R output file "./MakePriorsProteinEvolverABC.r" ...

> Writing executable output file for simulations
"./ProteinEvolverABC_Phase2.sh", "./ProteinEvolverABC_arguments.txt" and
"./ProteinEvolverABC_Pop_arguments.txt" ...
```

Finally, the script computes the specified prior distributions,

```
> Computing prior distributions ...
This phase requires the programming language R with the installed libraries:
lattice, MCMCpack, ape, graphics
An output file is created with the R outputs ...

> Successful!
```

-> Step 2 of *ProteinEvolverABC* is executed with *ProteinEvolverABC_Phase2.pl* and indicating the settings file. The amount of information about the computer simulations printed on the screen depends on the previous activation of the option *ShowInformationScreen* in the settings file (in the following example the information of computer simulations on the screen was not activated to save computer time). This step will perform the simulations and can run in parallel if the user specifies it in the settings file,

```
ProteinEvolverABC_proof$ perl ProteinEvolverABC_Phase2.pl Settings.txt
```

```
******************************************************************************
**********************************
ProteinEvolverABC_Phase2.pl
ProteinEvolverABC Phase 2 does:
- Read Settings file
- Perform simulations
******************************************************************************
**********************************

> OS detected: darwin (Mac OS X):
  darwin
  darwin-thread-multi-2level

  Number of simulations: 250
  Indels (gaps) are ignored
  Number of processors to run the simulations: 1
  Simulated data will not be saved
  Running information (simulations and summary statistics) is not shown on the
screen

> Performing simulations..
  "./ProteinEvolverABC_Phase2.sh" was detected... Ok!
  "./ProteinEvolverABC_arguments.txt" was detected... Ok!
Simulation # 250

> Generating files "ProteinEvolverABC_Phase3.sh" and "ArgumentsSS.txt" for the
phase 3...

> Successful!
```

This step can take a considerable amount of time depending on the number of simulations to perform, the size of the simulated alignments (number of sequences and sequence length) and the recombination rate; increasing those numbers increases the computer time. Running simulations in parallel (when possible) greatly reduces the computer time.

-> Step 3 of *ProteinEvolverABC* is executed with *ProteinEvolverABC_Phase3.pl* and indicating the settings file. The amount of information about the computed summary statistics that is printed on the screen depends on the previous activation of the option *ShowInformationScreen* in the settings file (in the following example the information of computer simulations on the screen was not activated to save computer time). This script computes summary statistics from the previously simulated multiple alignments of protein sequences,

```
ProteinEvolverABC_proof$ perl ProteinEvolverABC_Phase3.pl Settings.txt

******************************************************************************
**********************************
ProteinEvolverABC_Phase3.pl
ProteinEvolverABC Phase 3 does:
- Compute the summary statistics for the simulated data
******************************************************************************
**********************************

 Indels (gaps) are ignored
 Simulated data will not be saved
 Running information (simulations and summary statistics) is not shown on the
screen

> Computing summary statistics from the simulated data "./sequences*"..
  "./ProteinEvolverABC_Phase3.sh" was detected... Ok!
  "./ArgumentsSS.txt" was detected... Ok!
SSs dataset: "./sequences000000250"
```

```
> Concatenating summary statistics from each simulated data ..

> Summary statistics placed in the file "./SSsimulations.ss"

> Checking output files and preparing files for the ABC estimation phase..

Working on files "./ProteinEvolverABC_Pop_arguments.txt" and
"./SSsimulations.ss"
Simulations with incomplete calculation of Summary Statistics will be
skipped..
Done!

> Successful!
```

Simulations for which it was not possible to compute all the summary statistics are skipped (e.g. due to division by zero errors, etc). This would appear during the execution of this step,

```
SSs dataset: "./sequences000000036"
ERROR: Too few informative sites to test significance.  Try decreasing
windowsize or increasing alignment length


Exiting...
rm: Phi.poly.unambig.sites: No such file or directory
```

An at the end of this execution,

```
Simulations with incomplete calculation of Summary Statistics will be
skipped..
skipped lines: 36
Done!
```

Skipped simulations will not be considered in the estimation phase. This may not be a problem for the estimation if the number of non-skipped simulations is large enough. In case of a large number of skipped simulations, one should revise the prior distributions (unrealistic prior distributions could be problematic, for example if simulated data do not display substitution events) and/or increase the number of simulations to ensure a proper sampling of the parametric space.

Depending on the specifications in the settings file, simulated data can be either removed or compressed into a file “SimulatedData.tar.gz” stored in the “OtherOutputs” output folder.

-> Step 4 of *ProteinEvolverABC* is executed with *ProteinEvolverABC_Phase4.pl* and indicating the settings file. This script computes ABC estimations of substitution and recombination rates in the target multiple alignment of protein sequences by considering the previously obtained summary statistics of the simulated and input target datasets,

```
ProteinEvolverABC_proof$ perl ProteinEvolverABC_Phase4.pl Settings.txt

******************************************************************************
**********************************
ProteinEvolverABC_Phase4.pl
ProteinEvolverABC Phase 4 does:
- Read Settings file
- Perform ABC analysis using simulated data and target data (requires R with
library: abc)
******************************************************************************
*********************************

> Reading Settings from input file ...
  Number of simulations: 250
  Number of iterations (simulations considered for the estimation phase): 250
  Tolerance number: 50
  ABCMethod: loclinear
  ABCTransf: logit
  ABCHCorr: 0
  SummaryStatistics: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
  MultiPage: 1

> Writing R output file "./PerformABC.r" ...

> R output file created: "./PerformABC.r" ...

> Performing ABC...
This phase requires the programming language R with the library: abc
 Summary Statistics file read
 Parameters file read
 Sequences data file read

> ABC method performed, printing results ...

Estimates from the rejection step:
                    Rho      Theta
Min.:          1.472322  59.561340
2.5% Perc.:    2.488356  67.776100
Median:        7.669024 179.916235
Mean:          8.059174 197.589595
Mode:          6.265842 129.685646
97.5% Perc.:  13.103307 364.699451
Max.:         13.337113 378.145885

Estimates from the local linear regression step:
                             Rho     Theta
Min.:                   13.41035 319.66342
Weighted 2.5 % Perc.:   13.41035 336.99676
Weighted Median:        13.41036 339.70968
Weighted Mean:          13.41036 339.27703
Weighted Mode:          13.41036 339.67371
Weighted 97.5 % Perc.:  13.41036 340.85080
Max.:                   13.41036 343.38374

 Plots of posterior distributions from the rejection step created
 Plots of posterior distributions from the local linear regression step
created
 Plots to assess ABC analysis integrity created

> Successful!
```

4.4. Output Files

Several output files are generated by *ProteinEvolverABC* during the different stages of the estimation. These files are saved in the output folders "*MainOutputs*" and "*OtherOutputs*" that will be placed in the working directory.

4.4.1. Output files from simulations

A variety of information is saved from the simulations in the working directory, output folder "*OtherOutputs*":

- The file "*ProteinEvolverABC_arguments.txt*" shows the *ProteinEvolver* arguments used to perform the simulations (further details in the *ProteinEvolver* documentation, https://github.com/MiguelArenas/proteinevolver). The file "*ProteinEvolverABC_Pop_arguments.txt*" includes the same information but considering the population recombination ρ and amino acid substitution θ rates

(Section 1.1). The file "*PSimulations.csv*" shows the values sampled from the prior distributions for the parameters to be estimated (ρ and θ).

- The simulated alignments are compressed into the file "*SimulatedData.tar.gz*" if the user specified in the settings file the option of save the simulations.
- The files "*Histogram_PriorRecombination.pdf*" and "*Histogram_PriorSubstitution.pdf*" show illustrative histograms of the prior distributions used for the simulations of recombination and substitution rates, respectively. In addition, histograms are also provided for ρ and θ.
- The summary statistics computed from each simulated data are printed in files "*SSsimulations.ss*" and "*SSsimulations.csv*".
- Some temporal scripts and outputs (namely, *MakePriorsProteinEvolverABC.r*, *ProteinEvolverABC_Phase2.sh*, *ProteinEvolverABC_Phase3.sh, Output_MakePriorsProteinEvolverABC.txt*, *PerformABC.r*) are also stored in the folder "*OtherOutputs*". These scripts can be helpful for debugging any running-error of *ProteinEvolverABC* (e.g., checking errors from incorrect specification of parameters for the simulations).

4.4.2. Summary statistics from the target multiple alignment of protein sequences

- The summary statistics computed from the target multiple alignment of protein sequences are printed in the file "*SSRealData.ss*" and saved in the folder "*OtherOutputs*".

4.4.3. Output files from estimations

The information on the ABC estimations is saved in the folder "*MainOutputs*":

- The file "*Histograms_SStats.pdf*" shows a histogram of the values of summary statistics from the retained simulations for each summary statistic considered. Vertical blue lines corresponding to the value of the summary statistic of the target sequences alignment are added. Ideally the histograms should have a Gaussian-like shape with the blue line in the centre of the distribution. These plots are used to assess if the model assumed for the simulations fits well the data.
- The file "*Scaterplots_SStatsVSParams.pdf*" shows a scatter plot of the values of the summary statistics and corresponding values of the parameters assumed for the simulation (i.e. recombination and substitution rates) for each pair of considered summary statistic and parameter. Again, a blue line corresponding to the value of the summary statistic of the target sequences alignment is added. Ideally the blue line should be somewhere in the centre of the y-axis. These plots can give a rough visualization of the local linear univariate relation between the parameters and the summary statistics. Note, however, that the adjustment via the regression-step is performed via a multiple regression, which considers all the summary statistics together.
- The file "*PCA_SStats.pdf*" shows a scatter plot of the two first principal components of a principal component analysis (PCA) on the considered summary statistics. The black points correspond to a sample of all the simulations, the red points correspond to all the retained simulations, and the yellow point correspond to the target sequences alignment. Ideally the red area should be well inside the black area and the yellow point should be well inside

the red area. This plot is also used to assess if the model assumed for the simulations fits well the data.

- The files "*Posteriors_rejection.pdf*" and "*Posteriors_loclinear.pdf*" show the posterior distributions of the estimated parameters for the *regression* and *rejection* methods, respectively. Black lines represent the prior distributions, and they are obtained by considering a random sample of the simulations. Histograms and their smooth curve representation (blue lines) represent the posterior distributions.
- The files "*Summary_ABC_rejection.csv*" and "*Summary_ABC_loclinear.csv*" present the estimations of recombination and substitution rates under the *rejection* and *regression* methods, respectively. These files are in a "comma separated values" formatted and can be easily opened by excel or other spreadsheet software.

4.5. Re-analysing data

The *ProteinEvolverABC* package allows choose a few different settings for the ABC estimation. These settings can be changed for a better tuning of the estimations. Among these, the most important to consider are the following:

- ABC iterations. This setting defines the number of simulations to consider in the ABC analysis. By considering different values of this setting, the user may get some insights on how well the entire space of the simulations is being characterized. For estimations using a target dataset that was simulated, in which one can assume for the simulations the exact model that originated the target alignment, I showed that as little as 50,000 simulations are enough for consistent estimations. However, when considering "real" target sequences alignments, many more points may be required (e.g. 100,000).

- ABC tolerance. This setting defines the number of simulations closest to the target sequences alignment that are retained. Choosing the tolerance is not trivial: it has to be large enough so that to provide a good characterization of the posterior distribution, but on the other hand, it has to be small enough so that the simulations retained are not very far from the target sequences alignment. The use of the regression-step adds another concern: performing multiple regressions using small number of points can introduce some problems. For estimations using simulated target alignments I found that a tolerance of 100 simulations worked fine for several different total number of simulations. However, when considering "real" target alignments, a wider tolerance can be needed (e.g. 1,000 simulations).

- ABC transformation. This setting defines which, if any, transformation will be performed to the parameters' values before adjusting them with the regression method. These transformations have the advantage of ensuring that the regression-adjustment does not project points outside the region of interest. Particularly, the "log" transformation ensures that the adjusted values of the retained simulations are positive, while the "logit" transformation ensures that they are inside the prior distribution range.

- ABC correction. This setting defines whether to correct for the heteroscedasticity of the parameters when performing the regression adjustment. This correction was introduced by Blum and Francois [21] and was shown to have considerable impact on the ABC estimations in particular cases [21].

- Summary statistics to use. This setting defines which summary statistics to consider when performing ABC estimations. As the number of simulations and the tolerance interval, the choice on the summary statistics is not trivial in ABC. For the estimation of recombination and substitution rates, we showed that the use of all the summary statistics provided in *ProteinEvolverABC* works fairly well.

Due to the advantages of re-analysing the same simulated data with different settings of the ABC estimation, we suggest a simple way to perform these re-analysis:
1. Rename the "*MainOutputs*" folder.
2. Create a new Settings file with different ABC settings.
3. Re-run step 4 of the *ProteinEvolverABC* pipeline with the new Settings file.
Note, however, that you should keep track on which Settings file correspond to which renamed "*MainOutputs*" folder.

4.6. Examples

The package includes the following examples in the folder "*examples*" including all input and output files:

- **Example1rapid:** Analysis of a simulated real target alignment [35 sequences, 604 amino acids using only 250 simulations (a few simulations with the aim of only performing a rapid exploration of the framework and so results may not be accurate) with the following prior distributions [ρ = Uniform(0,13.43) and θ: = Uniform(0,403.47)] while assuming an effective population size *N* = 1000. The simulations were not saved to save space in the hard disk.
- **Example2:** Analysis of a simulated target protein sequences alignment [25 sequences, 500 amino acids, *N* = 1,000, *Rho* = 60 ($r = Rho/4Nl = 3.0E^{-5}$), *Theta* = 200 ($\mu = Theta/4NL = 1.0E^{-4}$)] using 60,000 simulations (from which 50,000 were used to perform the ABC estimation) with the following prior distributions [*Rho* = Uniform(0,120), *Theta*: = Uniform(0,500)]. Running this example can take more than 1 day.
- **Example3:** Analysis of a Pancreatic ribonuclease real dataset (4 sequences, 113 amino acids) [22], using 50,000 simulations with the following prior distributions [*Rho* = Uniform(0,120) and *Theta*: = Uniform(0,500)]. Running this example on a single processor can take around 2.5 hours.
- **Example4:** Analysis of a Glucokinase real dataset (4 sequences, 319 amino acids) [22], using 50,000 simulations with the following prior distributions [*Rho* = Uniform(0,120) and *Theta*: = Uniform(0,500)]. Running this example on a single processor can take around 5 hours.
- **Example5:** Analysis of a Homogentisate 1,2-dioxygenase real dataset (4 sequences, 419 amino acids) [22], using 50,000 simulations with the following prior distributions [*Rho* = Uniform(0,120) and *Theta*: = Uniform(0,500)]. Running this example on a single processor can take around 5.5 hours.
- **Example6:** Analysis of a Cytochrome P450 real dataset (4 sequences, 465 amino acids) [22], using 50,000 simulations with the following prior distributions [*Rho* = Uniform(0,120) and *Theta*: = Uniform(0,500)]. Running this example on a single processor can take around 6.5 hours.
- **Example7:** Analysis of a Protease HIV-1 real dataset downloaded from the Pfam database (Pfam ID: PF09668; 10 sequences, 124 amino acids), using 50,000 simulations with the following prior distributions [*Rho* = Uniform(0,120)

and *Theta*: = Uniform(0,500)]. Running this example on a single processor can take around 5.5 hours.

Note that Example2 can require a considerable amount of time to run, thus for example it can be run on 8 processors. *The number of processors would depend on the machine capabilities.* For further details on the assumed models for each example see the specifications of the respective input Settings file.

4.7. Message errors and recommendations

Errors generated from incorrect settings are usually shown on the screen and the program will suggest abort the execution by typing CTRL+C. **For beginners, I recommend copying a provided input file "Settings.txt" from the "examples" folder and edit it with the desired settings or use the GUI to generate the input file "Settings.txt".**

Running Phase 1

The input multiple sequence alignment must be presented in *standard phylip sequential* format (section 4.2). In order to avoid errors with the format, we recommend use the format alignment converter *ALTER* (http://sing.ei.uvigo.es/ALTER/) [20]. Therein after uploading your input alignment, go to "*Select program*" and in the section "*Population genetics*" please select "*CodABC*". Then select "*PHYLIP*" and finally, covert.
Importantly, the amino acid sequences of the input multiple sequence alignment should only include any of the 20 amino acids (one-letter code) or indels (as "-"). Other letters or symbols (i.e., X, $, ., etc) will produce an error displayed on the screen.

Running Phase 2

If simulations start but stop before the R execution, check that the R libraries (lattice, MCMCpack, ape, graphics, abc) are correctly installed (see section 3.1). The R function "suppressPackageStartupMessages" should also be installed for this. Also check that you have compiled *ProteinEvolverABC* and the executable files appear in the "*bin*" folder (see sections 3.2 and 4.3).

If simulations start but stop after the R execution, check the documentation of *ProteinEvolver* but most likely a setting is incorrectly specified (revise settings following the section 4.1).

In general, note that the simulation phase can be computational costly with long running times. This is particularly found when performing a large number of simulations, when the simulated data includes a large number and long sequences, or when considering high recombination rates. Therefore, obviously, reducing the number of simulations and the size of the input alignment of protein sequences can dramatically reduce computer time. Sometimes reducing the input dataset is not be possible and then one can try to reduce the number of simulations. In this direction, one can reduce the prior distribution of the recombination and substitution rates (assuming the user has an idea about the possible estimates of these parameters) to have an enough number of simulations covering the parametric landscape of the prior distributions. Indeed, running the simulations in parallel (if the machine has more than one processors) can reduce computer times.

Another concern is the lack of RAM memory when running the computer simulations. This can occur in scenarios with a user-specified very high recombination rate. Note that recombination increases the number of nodes of the ARG (ancestral recombination graph [16, 17]). Thus, when ρ is very high (e.g. above 150), the ARG needs to save in memory a high amount of information. As the simulation time exponentially increases with ρ, it may lead to lack of available memory and the program stops with an error message. In this situation we recommend reduce the user-specified values of ρ (in its prior distribution) or population size. Of course, another option is just use a machine with more RAM memory.

Running Phase 3

When computing the summary statistics from the simulations one can find that for some simulations one or more summary statistics cannot be computed (e.g. due to division by zero errors, etc). For example, the Phi summary statistics cannot be computed if there is not an enough number of substitutions in the input data and one can find on the screen a text like,

```
SSs dataset: "./sequences000000036"
ERROR: Too few informative sites to test significance.  Try decreasing
windowsize or increasing alignment length


Exiting...
rm: Phi.poly.unambig.sites: No such file or directory
```

This is not problematic, those simulations will be automatically excluded for the estimation phase.

The computation of summary statistics can take a long time if there are many simulated data and in the simulated data is large (many and long sequences). Unfortunately, by the moment this computation could not be parallelized and I hope this will be done in a future version of the framework.

Running Phase 4

One can find that the estimation with the regression approach could not be performed. In this situation this phase can be executed again (section 4.5) varying the ABC tolerance in the input settings file. See section 5.3 for details about the rejection and regression approaches. The rejection approach can be more robust (in terms of being able to work) than the regression approach to analyze real data but in theory the regression approach can (not always) provide more accurate estimates than the rejection approach due to its complexity (section 5.3). In addition to varying the tolerance, one can increase the number of simulations to obtain results from the regression approach but this implies repeat all the phases (1-4) with the new number of simulations.

One can also find that the results from the rejection approach can disagree with the results from the regression approach. This can occur especially when analyzing real data and there is not a direct answer to decide which approach is more accurate for the study dataset [23]. For a given real dataset the rejection approach can be more accurate than the regression approach, and *vice versa* for another real dataset. As indicated above, the rejection approach is more robust (in terms of being able to work) than the regression approach to analyze real data but in theory the regression approach can (not always) provide more accurate estimates than the rejection approach. Therefore the basis could be using the results from the rejection approach if the results from the regression approach are not available or if they are unrealistic (i.e., falling out of the priors, negative, etc). A formal procedure to select the estimation method would be performing computer simulations (mimicking the real dataset) where the true values of the parameters under study (sampled from the corresponding prior distributions) are known, but this can take a long time and also the simulated data could still differ from the real data (i.e., if the real data suffered a complex evolutionary process and presents a complex pattern of molecular diversity, which is a limitation of ABC).

If you find any unexpected error, or there is any doubt, do not hesitate to contact miguelmmmab@gmail.com. Thanks for your contribution!

5. Models and Methods

ProteinEvolverABC is a framework to estimate recombination and substitution rates from protein sequence alignments using an ABC procedure. The simulator includes a variety of evolutionary models that may help to better mimic real scenarios. Then, informative summary statistics are calculated from the simulated data. Finally, the computation of the posterior distributions is performed using ABC methods.

5.1. Evolutionary Models and Computer Simulations

The first step of *ProteinEvolverABC* is based on the simulator *ProteinEvolverProtABC*, which is an adapted version of the program *ProteinEvolver* (https://github.com/MiguelArenas/proteinevolver) [1] to perform ABC.
ProteinEvolverProtABC implements a variety of evolutionary models and generates protein sequence alignments (see below) collected at same or different times (temporal longitudinal sampling or tip dates [see, 24]), from a population evolved under complex recombination by using the coalescent approach [25-27]. Protein sequences are evolved under a variety of empirical substitution models of protein evolution *Blosum62* [28], *CpRev* [29], *Dayhoff* [30], *DayhoffDCMUT* [31], *HIVb* [32], *HIVw* [32], *JTT* [33], *JonesDCMUT* [31], *LG* [34], *Mtart* [35], *Mtmam* [36], *Mtrev24* [37], *RtRev* [38], *VT* [39], *WAG* [40] and any user-defined model. Both haploid and diploid data can be simulated. Indeed, the program also implements heterogeneity across sites according to a gamma distribution (+G) and proportion of invariable sites (+I) [41].

5.2. Summary Statistics

The package implements a total of 16 summary statistics (Table 2).
Three of the summary statistics are fast statistical tests for detecting recombination: the maximum chi-squared method [χ^2, [42]], the neighbor similarity scope [*NSS*, [43]], and the pairwise homoplasy index [*PHI*, [15]], available in the *PhiTest* software [15]. Other calculated summary statistics are the mean, standard deviation (sd), skewness (sk) and kurtosis (ku) of the amino acid (aa) diversity, aa heterozygosity (*H*) and aa pairwise sequence identity (si). Another summary statistic is the number of aa segregating sites (*S*).

Table 2. The implemented summary statistics, their code and their identifiers (Id).

Description	Code	Id
pairwise homoplasy index PHI	Phi	1

neighbor similarity scope NSS	NSS	2
maximum chi-squared χ^2	ChiSq	3
mean of aa diversity	p_av	4
sd of aa diversity	p_sd	5
sk of aa diversity	p_sk	6
ku of aa diversity	p_ku	7
mean of aa heterozygosity	H_av	8
sd of aa heterozygosity	H_sd	9
sk of aa heterozygosity	H_sk	10
ku of aa heterozygosity	H_ku	11
number of aa segregating sites	S	12
mean of pairwise aa sequence identity	si_av	13
sd of pairwise aa sequence identity	si_sd	14
sk of pairwise aa sequence identity	si_sk	15
ku of pairwise aa sequence identity	si_ku	16

5.3. ABC Methods

Two ABC methods are implemented in *ProteinEvolverABC*, a simple rejection method [4] and a weighted multiple linear regression method as presented in [5]. The detailed algorithms of these methods are described below (further details can be found in the references given, as well as in various review papers such as [23] or [44]).
These methods are performed using the R package "*abc*", further details on their implementation can be seen in [45].

5.3.1. Rejection method

The basic algorithm, as introduced by Pritchard and co-workers [4] can be written as following:

(1) Sample (vector valued) parameter, Φ, from the prior: $\Phi_i \sim p(\Phi)$;
(2) Simulate data, D, given Φ_i : $D_i \sim p(D|\Phi_i)$;
(3) Summarize D_i with a set of chosen summary statistics to obtain S_i; go to (1) until N sample points from the joint distribution $p(S, \Phi)$ have been created;
(4) Accept δ points whose S_i have the smallest distance to s', the real data summarized by the same set of summary statistics.

In steps (1) to (3), one simulates independent pairs (Φ_i, S_i), $i = 1, 2, \ldots, N$, where each Φ_i is an independent draw from the prior distribution of Φ, and S_i are values that summarize simulated values of D with $\Phi = \Phi i$. The (Φ_i, S_i) are then random draws from the joint density. Step (4), the rejection-step, provides an estimate of the conditional density, when $S = s'$, that is, the posterior distribution $p(\Phi|S = s')$. The idea is that the Φ_i for which $|S_i - s'|$ is small forms an approximation of a random sample from the desired posterior distribution $p(\Phi|D = d')$.

5.3.2. Regression method

In the regression method two changes are proposed at step (4): smooth weighting and regression adjustment. The aim is to improve the sampling of the posterior density by weighting the Φ_i according to its distance from the real data by evaluating $|S_i - s'|$ and to adjust the Φ_i using a local-linear regression to weaken the effect of that discrepancy [5].

(4) Apply a weighting scheme to the points according to their distance to the *s'*. And then perform a linear multiple regression with points assigned non-zero weight. Adjust them according to $\Phi_i^* = \Phi_i + R_i$, where R_i is the residual of the i^{th} point.

As noted by Beaumont and co-workers [5] the rejection method can be viewed as a special case of the local-linear regression approach when using a uniform kernel and a local-constant regression. Actually, as δ tends to zero, the regression and rejection methods become equivalent.

5.4. ProteinEvolverABC Performance

There is a lack of evolutionary frameworks implementing the estimation of recombination rate from protein sequences, which does not allow a direct comparison of *ProteinEvolverABC* with other frameworks. Still, I can mention that the methodology used by *ProteinEvolverABC* but applied on DNA coding sequences showed more accurate estimations of the recombination rate than common likelihood-based tools such as *OmegaMap* [9] or *Lamarc* [8], see [2].

I preformed an evaluation of *ProteinEvolverABC* using computer simulations with *ProteinEvolver* [1]. I defined a total of 20 evolutionary scenarios combining the values of ρ = 0, 30, 60 and 90 with the values of θ = 50, 100, 200, 300 and 400 (producing sequence identity between 0.92 and 0.59), which are values commonly observed in nature [e.g., 2, 7]. Each simulated dataset included 25 sequences and 500 amino acids of sequences length and evolved under the JTT empirical substitution model of protein evolution [33]. A total of 100 simulated datasets (hereafter named as test datasets) were performed for each evolutionary scenario. The ABC estimation in the test datasets was performed using 50,000 simulations (hereafter named as model datasets, that are independent of the test datasets) and under both rejection (Rej) and regression (Reg) methods. The estimations of recombination and substitution rates are shown in Figures 1 and 2, respectively.

The results indicate that both parameters can be accurately estimated using 50,000 computer simulations. As expected, the recombination rate can be more difficult to estimate at low levels of substitution rate (i.e., see scenario 2) although this effect can be reduced by increasing the number of simulations (using 100,000 simulations the estimation values fall within the confidence intervals in all the studied scenarios).

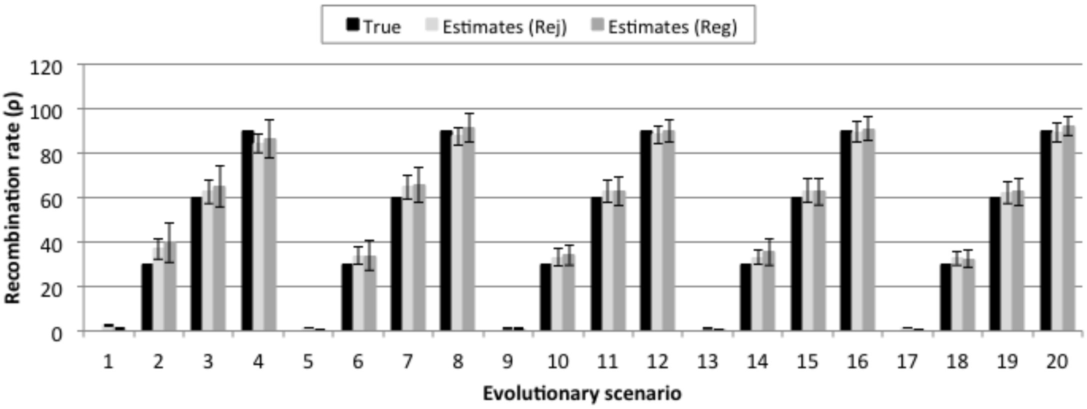


Figure 1. Evaluation of the estimation of the population recombination rate (ρ). Estimation performed under the rejection (Rej, clear grey bars) and regression (Reg, dark grey bars) methods and comparison with the true (simulated, black bars) values for each studied evolutionary scenario (a combination of ρ = 0, 30, 60 and 90 with θ = 50, 100, 200, 300 and 400), where every scenario includes 100 test datasets. Error bars indicate 95% confidence intervals from the mean. Study based on 50,000 computer simulations.

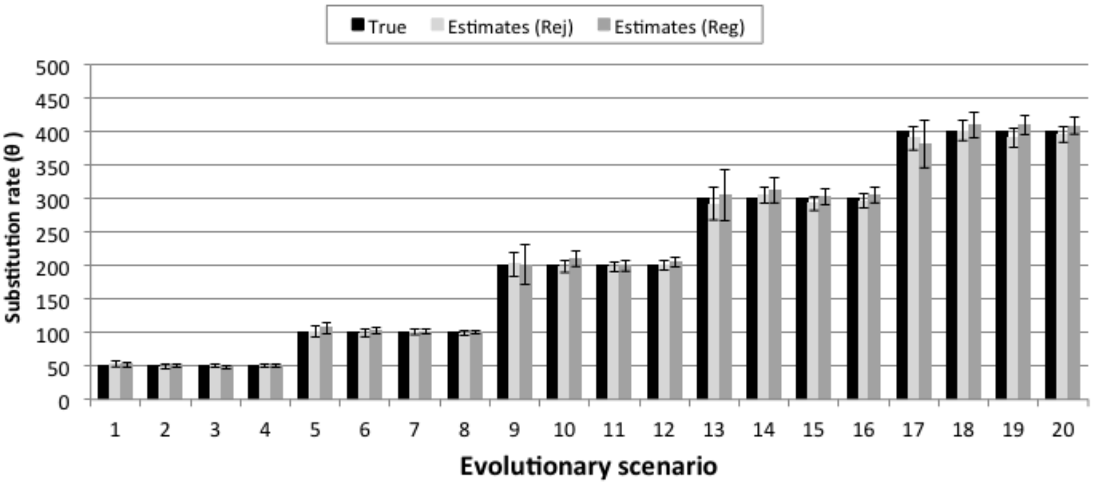


Figure 2. Evaluation of the estimation of the population substitution rate (θ). Estimation performed under the rejection (Rej, clear grey bars) and regression (Reg, dark grey bars) methods and comparison with the true (simulated, black bars) values for each studied evolutionary scenario (a combination of ρ = 0, 30, 60 and 90 with θ = 50, 100, 200, 300 and 400), where every scenario includes 100 test datasets. Error bars indicate 95% confidence intervals from the mean. Study based on 50,000 computer simulations.

Indeed, the main advantage of *ProteinEvolverABC* is being robust to influences of recombination on substitution estimates, and *vice versa*. This marks a difference between analytical tools based on ABC and based on other approaches [2, 5, 23, 46].

5.5. Assumptions and Limitations

ABC assumes that simulated data can mimic the real data. Simulations are based on models and are never fully representative of the real process. This is actually a limitation present in any estimation approach based on a model of evolution. The idea is to apply simulations as realistic as possible. The data-generating process assumed by *ProteinEvolverABC* consists of drawing a sample genealogy from the standard coalescent with recombination [26, 47], and then, given the genealogy, generating the sequences evolved under a substitution model of protein evolution [1]. This is a well-established methodology to simulate protein sequences upon evolutionary histories [48].

Recombination rate is a parameter that can be problematic to estimate in samples with low substitution rates. Indeed recombination events can be difficult to detect if there is not enough diversity in the data [49]. Thus, simulated data with low substitution rates may not reflect their recombination rates, which can complicate estimations of the latter parameter [50]. Nevertheless, in contrast to other analytical approaches, ABC can account for this uncertainty and reduce the bias in recombination estimates imposed by low substitution rates [2].

6. Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness and Ministry of Science and Innovation through the Grants [RYC- 2015-18241] and [PID2019-107931GA-I00/AEI/10.13039/501100011033] to MA.

7. References

1. Arenas M, Dos Santos HG, Posada D, Bastolla U: **Protein evolution along phylogenetic histories under structurally constrained substitution models**. *Bioinformatics* 2013, **29**(23):3020-3028.
2. Lopes JS, Arenas M, Posada D, Beaumont MA: **Coestimation of Recombination, Substitution and Molecular Adaptation rates by approximate Bayesian computation**. *Heredity* 2014, **112**(3):255-264.
3. Arenas M, Lopes JS, Beaumont MA, Posada D: **CodABC: A Computational Framework to Coestimate Recombination, Substitution, and Molecular Adaptation Rates by Approximate Bayesian Computation**. *Mol Biol Evol* 2015, **32**(4):1109-1112.
4. Pritchard JK, Seielstad MT, Perez-Lezaun A, Feldman MW: **Population growth of human Y chromosomes: a study of Y chromosome microsatellites**. *Mol Biol Evol* 1999, **16**(12):1791-1798.
5. Beaumont MA, Zhang W, Balding DJ: **Approximate Bayesian computation in population genetics**. *Genetics* 2002, **162**(4):2025-2035.
6. Arenas M, Araujo NM, Branco C, Castelhano N, Castro-Nallar E, Perez-Losada M: **Mutation and recombination in pathogen evolution: Relevance, methods and controversies**. *Infect Genet Evol* 2018, **63**:295-306.
7. Arenas M, Lorenzo-Redondo R, Lopez-Galindez C: **Influence of mutation and recombination on HIV-1 in vitro fitness recovery**. *Mol Phylogenet Evol* 2016, **94**(Pt A):264-270.
8. Kuhner MK: **LAMARC 2.0: maximum likelihood and Bayesian estimation of population parameters**. *Bioinformatics* 2006, **22**(6):768-770.

9. Wilson DJ, McVean G: **Estimating diversifying selection and functional constraint in the presence of recombination**. *Genetics* 2006, **172**(3):1411-1425.
10. McVean GA, Myers SR, Hunt S, Deloukas P, Bentley DR, Donnelly P: **The fine-scale structure of recombination rate variation in the human genome**. *Science* 2004, **304**(5670):581-584.
11. Martin DP, Murrell B, Golden M, Khoosal A, Muhire B: **RDP4: Detection and analysis of recombination patterns in virus genomes**. *Virus Evol* 2015, **1**(1):vev003.
12. Arenas M, Posada D: **Coalescent simulation of intracodon recombination**. *Genetics* 2010, **184**(2):429-437.
13. Arenas M, Posada D: **The effect of recombination on the reconstruction of ancestral sequences**. *Genetics* 2010, **184**(4):1133-1139.
14. Darriba D, Taboada GL, Doallo R, Posada D: **ProtTest 3: fast selection of best-fit models of protein evolution**. *Bioinformatics* 2011, **27**(8):1164-1165.
15. Bruen TC, Philippe H, Bryant D: **A simple and robust statistical test for detecting the presence of recombination**. *Genetics* 2006, **172**(4):2665-2681.
16. Arenas M: **The Importance and Application of the Ancestral Recombination Graph**. *Front Genet* 2013, **4**:206.
17. Griffiths RC, Marjoram P: **An ancestral recombination graph**. In: *Progress in population genetics and human evolution.* Edited by Donelly P, Tavaré S, vol. 87. Berlin: Springer-Verlag; 1997: 257-270.
18. Arenas M: **Trends in substitution models of molecular evolution**. *Front Genet* 2015, **6**:319.
19. Felsenstein J: **Inferring phylogenies**. MA: Sinauer Associates: Sunderland; 2004.
20. Glez-Pena D, Gomez-Blanco D, Reboiro-Jato M, Fdez-Riverola F, Posada D: **ALTER: program-oriented conversion of DNA and protein alignments**. *Nucleic Acids Res* 2010, **38**(Web Server issue):W14-18.
21. Blum MGB, François O: **Non-linear regression models for Approximate Bayesian Computation**. *Statistics and Computing* 2010, **20**:63-73.
22. Bordner AJ, Mittelmann HD: **A new formulation of protein evolutionary models that account for structural constraints**. *Mol Biol Evol* 2013, **31**(3):736-749.
23. Beaumont MA: **Approximate Bayesian Computation in Evolution and Ecology**. *Annu Rev Ecol Evol Syst* 2010, **41**:379-405.
24. Navascues M, Depaulis F, Emerson BC: **Combining contemporary and ancient DNA in population genetic and phylogeographical studies**. *Mol Ecol Resour* 2010, **10**(5):760-772.
25. Kingman JFC: **The coalescent**. *Stochastic Processes and their Applications* 1982, **13**:235-248.
26. Hudson RR: **Properties of a neutral allele model with intragenic recombination**. *Theor Popul Biol* 1983, **23**(2):183-201.
27. Hudson RR: **Generating samples under a Wright-Fisher neutral model of genetic variation**. *Bioinformatics* 2002, **18**(2):337-338.
28. Henikoff S, Henikoff JG: **Amino acid substitution matrices from protein blocks**. *Proc Natl Acad Sci U S A* 1992, **89**(22):10915-10919.
29. Adachi J, Waddell PJ, Martin W, Hasegawa M: **Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA**. *J Mol Evol* 2000, **50**(4):348-358.

30. Dayhoff MO, Schwartz RM, Orcutt BC: **A model of evolutionary change in proteins**. In: *Atlas of protein sequence and structure.* Edited by Dayhoff MO, vol. 5, Suppl. 3. Washington D. C.; 1978: 345-352.
31. Kosiol C, Goldman N: **Different versions of the Dayhoff rate matrix**. *Mol Biol Evol* 2005, **22**(2):193-199.
32. Nickle DC, Heath L, Jensen MA, Gilbert PB, Mullins JI, Kosakovsky Pond SL: **HIV-specific probabilistic models of protein evolution**. *PLoS One* 2007, **2**(6):e503.
33. Jones DT, Taylor WR, Thornton JM: **The rapid generation of mutation data matrices from protein sequences**. *Comput Appl Biosci* 1992, **8**(3):275-282.
34. Le SQ, Gascuel O: **An improved general amino acid replacement matrix**. *Mol Biol Evol* 2008, **25**(7):1307-1320.
35. Abascal F, Posada D, Zardoya R: **MtArt: A New Model of Amino Acid Replacement for Arthropoda**. *Mol Biol Evol* 2007, **24**(1):1-5.
36. Yang Z, Nielsen R, Masami H: **Models of amino acid substitution and applications to mitochondrial protein evolution**. *Mol Biol Evol* 1998, **15**(12):1600-1611.
37. Adachi J, Hasegawa M: **MOLPHY version 2.3: programs for molecular phylogenetics based in maximum likelihood**. *Comput Sci Monogr* 1996, **28**:1-150.
38. Dimmic MW, Rest JS, Mindell DP, Goldstein RA: **rtREV: an amino acid substitution matrix for inference of retrovirus and reverse transcriptase phylogeny**. *J Mol Evol* 2002, **55**(1):65-73.
39. Muller T, Vingron M: **Modeling amino acid replacement**. *J Comput Biol* 2000, **7**(6):761-776.
40. Whelan S, Goldman N: **A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach**. *Mol Biol Evol* 2001, **18**(5):691-699.
41. Yang Z: **Among-site rate variation and its impact on phylogenetic analysis**. *Trends Ecol Evol* 1996, **11**(9):367-372.
42. Smith JM: **Analyzing the mosaic structure of genes**. *J Mol Evol* 1992, **34**(2):126-129.
43. Jakobsen IB, Easteal S: **A program for calculating and displaying compatibility matrices as an aid to determining reticulate evolution in molecular sequences**. *Comput Appl Biosciences* 1996, **12**:291-295.
44. Blum MGB, Nunes MA, Prangle D, Sisson SA: **A comparative review of dimension reduction methods in approximate Bayesian computation**. *Stat Sci* 2013, **28**(2):189-208.
45. Csillery K, Francois O, Blum MGB: **abc: an R package for approximate Bayesian computation (ABC)**. *Methods in Ecology and Evolution* 2012, **3**(3):475-479.
46. Bertorelle G, Benazzo A, Mona S: **ABC as a flexible framework to estimate demography over space and time: some cons, many pros**. *Mol Ecol* 2010, **19**(13):2609-2625.
47. Arenas M: **Computer programs and methodologies for the simulation of DNA sequence data with recombination**. *Front Genet* 2013, **4**:9.
48. Yang Z: **Computational Molecular Evolution**. Oxford, England.: Oxford University Press; 2006.

49. Posada D, Crandall KA: **Evaluation of methods for detecting recombination from DNA sequences: computer simulations**. *Proc Natl Acad Sci U S A* 2001, **98**(24):13757-13762.

50. Myers SR, Griffiths RC: **Bounds on the minimum number of recombination events in a sample history**. *Genetics* 2003, **163**(1):375-394.