DOCUMENTATION for **mlcoalsim** v1.42, multilocus coalescent simulations.

Sebastian E. Ramos-Onsins and Thomas Mitchell-Olds

March 18^{th}, 2008

Contents

1 *MLCOALSIM*: MULTILOCUS COALESCENT SIMULATIONS

The application program *mlcoalsim* (multilocus coalescent simulations) is designed to:

(i) Generate samples and calculate neutrality tests, and other statistics, under stationary model, several demographic models or strong positive selection by mean of coalescent theory.

(i) Perform coalescent simulations with the mutational phase given:

1. the population mutation rate θ ($\theta = 4N\mu$, where N is the effective population size and μ is the mutational rate).
2. a fixed number of mutations.
3. a distribution of θ values. A prior uniform (bounded) and a gamma distributions are enabled.
4. a fixed number of biallelic segregating sites taking into account the uncertainty of the population mutation rate (conditioning on biallelic segregating sites). A prior uniform (bounded) and a gamma distributions are enabled.

(iii) Perform coalescent simulations with recombination given:

1. the population recombination rate R ($R = 4Nr$, where r is the recombination rate).
2. a distribution of r values. A prior uniform (bounded) and a gamma distributions are enabled.
3. a fixed number of minimum recombination events (*Rm*) taking into account the uncertainty of the population recombination rate (fixing *Rm*). A prior uniform (bounded) and a gamma distributions are enabled.
4. a fixed number of minimum recombination events (*Rm*) and a fixed number of haplotypes, considering the uncertainty of the population recombination rate.

(iv) Perform multilocus analyses. Linked loci and unlinked loci are enabled. Multilocus statistics for unlinked loci are the average and the variance for each statistic.

(v) Include recurrent mutations (multiple hits) or not.

(vi) Include heterogeneity in mutation rate across the length of the sequence. A gamma distribution is used. Also, a number of invariant positions can also be defined.

(vii) Include heterogeneity in recombination rate across the length of the sequence. A gamma distribution is used. Hotspots or a constant value for all positions are possible.

This program is based on a previous version of Hudson's coalescent program *ms* (Hudson, 2002) and modified for the above purposes. The function to calculate minimum recombinant values is a modification of Wall's code (Wall, 2000). The gamma function was partially obtained from Grassly, Adachi and Rambaut code (Grassly et al., 1997).

This program is distributed under the GNU GPL License.

2 INSTALLATION

2.1 Installation of *mlcoalsim.*

The program is written in ANSI C, and can be compiled at any architecture with an ANSI C compiler. The program can be found at http://www.ub.edu/softevol/mlcoal. Download all the C code files and put together in the same folder:

mlsp_sm.h
get_mod_sm.c
neut_tests.c
ms2_sm.c
streec2_sm.c
main_sm2.c
input_hh3_sm.c
ran1.c

In case of using a Unix based platform and a gcc compiler, compile typing:

```
gcc *.c -lm -o mlcoalsim
```

In case of having Windows, use a compiler based on a gcc compiler.

2.2 Examples.

A folder with examples is available. The folder is called 'examples' and contains the example files contained in this documentation. Download this folder to see the examples and to verify the program compilation.

2.3 Perl scripts.

A couple of perl scripts are included. Simply download the folder called 'otherapps'. Instructions about how to use are in section 13 (Other Applicattions).

3 RUNNING *mlcoalsim.*

Simply type:

```
./mlcoalsim [input file] [output file with extension]
```

The inclusion of the paths for the input and the output files in the command line is optional. Otherwise, type *mlcoalsim* and the program will ask for the input and output files. The output file name must contain an extension name (*e.g.*, 'example.out').

4 INPUT FILE

mlcoalsim needs an input file for running. The input file must include the necessary information to conduct coalescent simulations in a single or multiple loci. In particular, this file should include the mutational method and the evolutionary model.

The input file must include the parameters names with their values separated by one or more spaces or tabulators. The different parameters should be separated by a return character. To run the program, the input file should include at least the following parameters:

- **seed1**: The random number seed. A positive integer to start the pseudorandom list of numbers.

- **print_neuttest** : type 1 to obtain an output with statistical and neutrality tests. In case of working with multilocus data, the average and variance is given in a single file. Type 2 if the user needs all statistics for each single locus plus the simulated average and variance. Otherwise 0.

- **print_matrixpol** : type 1 to obtain an output with the simulated sample sequences in phylip format. type 2 to have an output file in ms format. Type 3 to have an output file with samples in phylip format excluding multiple hits. Otherwise 0. **print_neuttest** and **print_matrixpol** are exclusives.

- **n_iterations**: number of iterations.

- **n_loci**: number of unlinked loci studied.

- **n_sites**: number of nucleotides studied per locus. The value for each locus must be separated by white spaces or tabs.

- **n_samples**: sample size per locus (separated by white spaces or tabs).

- **npop**: number of total populations.

- **recombination**: population recombination parameter ($4Nr$), where N is the population size and r the recombination rate per locus. The value for each locus must be separated by white spaces or tabs.

- **thetaw**: population mutation parameter ($4N\mu$), where N is the population size and μ the mutation rate per locus. Each locus must be separated by white spaces or tabs.

- **mutations**: the number of mutations (or biallelic segregating sites, depending on the opttion) per locus (separated by white spaces or tabs). These two parameters (thetaw, and mutations) should not be defined at the same time. A -1 value indicates that this parameter is not used.

- **factorn_chr**: Indicate the relative population size for each locus in relation to $4N$. For example, for loci located in autosome chromosomes in a diplod species, type 1. For loci located on the X chromosome type 0.75, etc.. The value for each locus must be separated by white spaces or tabs.

- **no_rec_males**: type 1 if recombination in males is inhibited (Drosophila), otherwise type 0 (default is zero).

- **likelihood_line**: type 1 if you desire the output in a single line. The output will be the log-likelihood value for the observed statistics (and combinations). The lines can be accumulated in the same file for each different condition parameters. Otherwise type 0 (default is zero).

Example of an input file:

```
seed1 7354
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
```

```
n_sites 1050
n_samples 25
npop 1

recombination 10
thetaw 5
mutations -1
factorn_chr 1
no_rec_males 0
likelihood_line 0
```

This input configuration will generate 1000 iterations under the standard neutral model for a sample of 25 lines of a single locus of 1050 nucleotides with recombination parameter 10 per locus and a fixed population mutation rate of 5 per locus. The output file will show a number of statistical tests calculated for each generated sample. Multiple hits are not considered.

5 OUTPUT FILES

Depending on the options included in the input file, the output files will be different. There are three major kinds of outputs:

5.1 DNA sequences matrix.

Those including a matrix with DNA sequence information of the sample for each locus (in ms and in fasta format). To obtain these outputs, you should include the parameter **print_neuttest** as 0 and **print_matrixpol** as 1, 2 or 3 for matrix in FASTA format, in ms format, or in FASTA format excluding the non-biallelic positions, respectively.

5.2 Statistical tests matrix.

Those including statistical and neutrality tests calculated at each iteration. In this case, the parameter **print_neuttest** will be 1 (or 2) and **print_matrixpol** will be 0. Note that all of these tests are calculated using only biallelic positions (*i.e.*, only positions that have two variants). The use of biallelic positions is justified because (i) many of these tests do not use tri- or tetra-allelic positions, (ii) the number of tri- and tetra-allelic positions are expected to be low in samples from the same species. The statistics and neutrality tests calculated are the following:

- *TD*: Tajima's D test (Tajima, 1989). This test basically looks at the differences between Watterson's estimate θ, (Watterson, 1975) and Tajima's estimate π (Tajima, 1983). Significant negative values indicate an excess of low frequency variants while positive values indicate an excess of intermediate frequency variants.
- *Fs*: Fu's F_s test (Fu, 1997). Use haplotype information and the nucleotide diversity to calculate this statistic. Powerful test when recombination parameter is known. Warning: this statistic can be somewhat bad calculated for large samples because a precision problem.
- *FD**: Fu and Li's D^* test without outgroup (Fu and Li, 1993). Fu and Li's tests are similar to Tajima's D test, but using different statistics related to the level of diversity.
- *FF**: Fu and Li's F^* test without outgroup (Fu and Li, 1993).

- *FD*: Fu and Li's D test with outgroup (Fu and Li, 1993).
- *FF*: Fu and Li's F test with outgroup (Fu and Li, 1993).
- *H*: Fay and Wu's H test (Fay and Wu, 2000).
- *B*: Wall's B test (Wall, 1999). Count the different haplotype structure at adjacent variants. Powerful to detect population subdivision.
- *Q*: Wall's Q test (Wall, 1999). Similar to B test, but try to take into account the recombination background.
- *ZA*: ZA statistic (Rozas et al., 2001). Statistic based on linkage disequilibrium at adjacent positions.
- *Fst*: Genetic differentiation among populations, F_{st} (Hudson et al., 1992). Here, the average differentiation among all populations is calculated.
- *Kw*: Number of haplotypes (Strobeck, 1987; Fu, 1996; Depaulis et al., 2001; Wall, 1999) divided by the sample size. Tests based on haplotype structure are usually quite powerful to detect positive selection events. On the other hand, a good estimate of recombination must be used.
- *Hw*: Haplotype diversity (Depaulis and Veuille, 1998) divided by the sample size.
- *R2*: Ramos and Rozas' R2 test (Ramos-Onsins and Rozas, 2002). Powerfult test to detect demographic expansion using small sample sizes in any recombinant or non-recombinant environment.
- *S*: Number of biallelic segregating sites.
- *pi_w*: Average nucleotide diversity within populations per locus (*e.g.,* Hudson et al., 1992).
- *pi_b*: Nucleotide diversity among populations per locus (*e.g.,* Hudson et al., 1992).
- *thetaWatt*: Watterson's estimate of nucleotide variation (θ) (Watterson, 1975).
- *thetaTaj*: Tajima's estimate of nucleotide variation per locus (nucleotide diversity, π) (Tajima, 1983).
- *thetaFW*: Fay and Wu's estimate of nucleotide variation (Fay and Wu, 2000). Useful for detection of high frequency variants produced by selective process.
- *D/Dmin*: Tajima's D divided by the minimum Tajima's D given S (Schaeffer, 2002; Schmid et al., 2005).
- *Hnorm*: Fay and Wu's H normalized (Zeng et al., 2006).
- *maxhap*: Number of lines in the most common haplotype (Depaulis et al., 2003) divided by the sample size.
- *maxhap1*: Simplified from (Hudson et al., 1994), count the number of lines with the most common haplotype but allowing a single segregating site within the biggest "haplotype" group (Rozas et al., 2001).
- *Rm*: Minimum number of recombinations of the sample (Hudson and Kaplan, 1985).
- *theta_fl*: Fu and Li's estimate of variability, based on singleton (Fu and Li, 1993).
- *theta_L*: Zeng *et al.* estimate of variability (Zeng et al., 2006).
- *Zeng_E*: Zeng *et al.* neutrality test (Zeng et al., 2006).
- *EW*: Ewens-Watterson test (Watterson, 1978).

- *Fstw*: Genetic differentiation among populations, F_{st} (Hudson et al., 1992). Here, the average differentiation among all populations is weighted by the number of samples per population.
- *Pwh*: Pwh test (unpublished) finds differences between the two more differentiated contiguous groups in the mismatch distribution.

Multilocus analyses will generate as many files as statistics calculated. The option **print_neuttest** set at 2 displays a multilocus summary with the average and variance for all loci.

Different statistics that estimate the level of variation are included: θ (Watterson, 1975), π (Tajima, 1983), and θ_H (Fay and Wu, 2000) for the entire sample. Although these statistics are calculated using different approaches, these values should be equivalent under the assumption of a stationary panmictic neutral model. The average levels of variation within and among different defined populations are also estimated (π_w, pi_b, *e.g.,* Hudson et al., 1992), as well as the average differentiation among populations with *Fst* statistic (*e.g.,* Hudson et al., 1992).

A description of the patterns of diversity is obtained using two main classes of statistics (Ramos-Onsins and Rozas, 2002): the Class I statistics, which use the information of the mutation frequency, and Class II statistics, which use information from the haplotype distribution.

Class I include the most popular test Tajima's D (*TD*, Tajima, 1989), and Fu and Li's tests (here *FD*,FF*,FD,FF,* Fu and Li, 1993). These tests compare different statistics in order to detect departures from neutrality when the statistical test is significantly different from zero. Also, Fay and Wu's H test (Fay and Wu, 2000) and *H_norm* (Zeng et al., 2006), designed to detect recent events of positive selection, $R2$ (Ramos-Onsins and Rozas, 2002), constructed to detect demographic expansions in populations, and weighted statistics for multilocus approach as *D/Dmin* (Schaeffer, 2002; Schmid et al., 2005).

Class II include the number of haplotypes, *Kw* (Strobeck, 1987) and the haplotype diversity, *Hw* (Depaulis and Veuille, 1998), both weighted by the number of samples for a better multilocus comparison, *Fs* (Fu, 1997) test, the statistics B and Q (Wall, 1999), *ZA* statistic (Rozas et al., 2001), *maxhap* (Depaulis et al., 2003), that count the number of lines that have the most common haplotype, and *maxhap1* (simplified from Hudson et al., 1994), which count the number of lines with the most common haplotype but allowing a single segregating site within the biggest "haplotype" group (Rozas et al., 2001).

5.3 Percentiles for statistical tests and probabilities for observed data.

When statistical tests are calculated, a file containing a table with percentiles for each statistical test for all loci considered and for averages and variances is printed. This file has the same name than the output file plus "_PPercentiles.out". In case the user include the values of statistical tests for observed data, the output will include the probability of the observed value be higher or equal than the simulated values. This option is useful for contrast of hypothesis.

5.4 Likelihood values

If the user decide to run a large number of simulations using different conditions, the option **likelihood_line**should be set at 1. Thus, the user can obtain in a single line the values of the likelihood for the observed(s) statistics and for combinations of statistics. The lines results can be accumulated in a single file simply by writing the same name to the output file to all the simulations.

5.5 Examples.

Example of an output file under the option **print_neuttest** set at 1:

```
mlcoalsim version 1.00 (20060725)
OUTPUT FILE: date Fri Aug 25 12:19:52 2006

Input data from the file:  ex00.txt

Print_neuttest:  1
Print_matrixpol:  0
N_iterations:  1000
Seed1:  7354

N_loci:  1
Factorn_chr:  1.000000
N_sites:  1050
N_samples:  25
Thetaw:  5
No_rec_males:  0
Recombination:  10


Neutral tests (excluding multiple hits positions)
value(TD) value(Fs) value(FD*) value(FF*) value(FD) value(FF) value(H) value(B) value(Q) value(ZA) value(Fst)
value(Kw) value(Hw) value(R2) value(S) value(pi_w) value(pi_b) value(ThetaWatt) value(ThetaTaj) value(ThetaFW*)
value(D/Dmin) value(H/Hmin) value(maxhap) value(maxhap1) value(Rm)
0.041944 -2.6104 0.018556 0.030184 -0.054837 -0.027135 0.78667 0.095238 0.13636 0.18003 na 0.56 0.92
0.12521 22 0 0 5.8263 5.8933 5.1067 0.016476 0.019433 0.24 0.24 3
0.48454 -2.521 0.44198 0.53257 0.44164 0.55181 -2.4867 0.21053 0.4 0.29455 na 0.56 0.91 0.14334 20 0
0 5.2967 6.0067 8.4933 0.19206 -0.067572 0.28 0.32 4
-0.066837 -2.5418 0.34763 0.25698 0.33247 0.24032 0.37667 0.090909 0.17391 0.23266 na 0.56 0.95 0.12377
23 0 0 6.0912 5.98 5.6033 -0.02615 0.0089004 0.16 0.24 1
0.64934 -1.6085 0.65893 0.7658 0.68234 0.80741 0.18667 0.16667 0.30769 0.26058 na 0.44 0.92667 0.15271
13 0 0 3.4428 4.0933 3.9067 0.27072 0.0078038 0.16 0.24 3
-1.0393 -5.6492 -1.4292 -1.5312 -1.7123 -1.7993 1.0033 0.071429 0.13333 0.084696 na 0.52 0.94 0.082645
15 0 0 3.9725 2.7933 1.79 -0.42531 0.036353 0.16 0.24 1
-0.17555 -5.045 -0.68905 -0.62252 -0.862 -0.76948 1.7967 0.071429 0.13333 0.15449 na 0.56 0.94 0.11497
15 0 0 3.9725 3.7733 1.9767 -0.071837 0.065097 0.16 0.2 2
0.61568 -1.7573 0.37765 0.52738 0.36589 0.54269 0.95333 0.11111 0.21053 0.31501 na 0.52 0.88 0.14793
19 0 0 5.0318 5.8933 4.94 0.24531 0.027269 0.32 0.36 3
-0.35553 -1.0232 -0.17185 -0.26545 -0.26349 -0.35228 -0.66 0.16667 0.23077 0.21715 na 0.36 0.88667 0.11221
13 0 0 3.4428 3.0867 3.7467 -0.14823 -0.027592 0.24 0.28 0
-0.10471 -2.5571 -0.71323 -0.614 -0.93008 -0.7888 0.99667 0.18519 0.25 0.26709 na 0.6 0.93 0.11569 28
0 0 7.4153 7.2067 6.21 -0.04032 0.019345 0.24 0.24 3
0.48023 -1.0838 -0.1231 0.066578 0.37024 0.48428 -0.43333 0 0 0.15335 na 0.36 0.83 0.1434 10 0 0 2.6483
3.0333 3.4667 0.2083 -0.023551 0.36 0.36 1
0.40825 -2.3845 0.78054 0.77933 0.83939 0.84217 3.3733 0.1 0.19048 0.20607 na 0.56 0.92 0.14243 21 0
0 5.5615 6.1867 2.8133 0.16106 0.087302 0.24 0.32 4
0.5961 -2.5299 -0.17185 0.070472 -0.26349 0.015428 1.1633 0 0 0.10229 na 0.48 0.90333 0.14583 13 0 0
3.4428 4.04 2.8767 0.24853 0.048634 0.24 0.28 2
1.3228 1.7859 0.051077 0.5118 -0.011691 0.50915 -0.47 0.35714 0.4 0.46216 na 0.32 0.72 0.17453 15 0 0
3.9725 5.4733 5.9433 0.54133 -0.017029 0.52 0.56 1
...
etc.
```

Each line (iteration) has a number of neutrality test. Each neutrality test is separated by a tabulator.

Example of an output file "*name*_PPercentiles.out" when statistical tests are calculated and observed data are included:

```
OBSERVED VALUES
TD Fs FD* FF* FD FF H B Q ZA Fst Kw Hw R2 S pi_w pi_b thetaWatt thetaTaj thetaFW D/Dmin H/Hmin maxhap
```

```
maxhap1 Rm
locus_0 -0.0494264 -1.59676 0.0200351 0.0131718 0.0134 -0.00123657 0.783333 0.125 0.205882 0.220047 x
0.6 0.946667 0.122479 37 x x 9.79884 9.54 9.13 -0.0189366 0.012183 0.16 0.2 3
average -0.0494264 -1.59676 0.0200351 0.0131718 0.0134 -0.00123657 0.783333 0.125 0.205882 0.220047 x
0.6 0.946667 0.122479 37 x x 9.79884 9.54 9.13 -0.0189366 0.012183 0.16 0.2 3
variance na na na na na na na na na na x na na na na x x na na na na na na na na

PROBABILITY THAT OBSERVED VALUE BE SMALLER OR EQUAL THAN SIMULATED VALUES. P(Sim <= Obs):  ACCURACY OF
+- 1e-6
TD Fs FD* FF* FD FF H B Q ZA Fst Kw Hw R2 S pi_w pi_b thetaWatt thetaTaj thetaFW D/Dmin H/Hmin maxhap
maxhap1 Rm
locus_0 0.5 0.501 0.501 0.5 0.501 0.5 0.5 0.509 0.5 0.5 na 0.531 0.532 0.5 0.522 na na 0.522 0.501 0.5
0.5 0.5 0.525 0.524 0.702
average 0.5 0.501 0.501 0.5 0.501 0.5 0.5 0.509 0.5 0.5 na 0.531 0.532 0.5 0.522 na na 0.522 0.501 0.5
0.5 0.5 0.525 0.524 0.702
variance na na na na na na na na na na na na na na na na na na na na na na na na na

PROBABILITY THAT OBSERVED VALUE BE EQUAL THAN SIMULATED VALUES. P(Sim = Obs):  ACCURACY OF +- 1e-6
TD Fs FD* FF* FD FF H B Q ZA Fst Kw Hw R2 S pi_w pi_b thetaWatt thetaTaj thetaFW D/Dmin H/Hmin maxhap
maxhap1 Rm
locus_0 0.001 0.002 0.003 0.001 0.002 0.001 0.003 0.013 0.005 0.001 na 0.156 0.036 0.001 0.043 na na
0 0.002 0.001 0.002 0.001 0.291 0.238 0.302
average 0.001 0.002 0.003 0.001 0.002 0.001 0.003 0.013 0.005 0.001 na 0.156 0.036 0.001 0.043 na na
0 0.002 0.001 0.002 0.001 0.291 0.238 0.302
variance na na na na na na na na na na na na na na na na na na na na na na na na na

Number of valid iterations:

TD Fs FD* FF* FD FF H B Q ZA Fst Kw Hw R2 S pi_w pi_b thetaWatt thetaTaj thetaFW D/Dmin H/Hmin maxhap
maxhap1 Rm
locus_0 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 0 1000 1000 1000 1000 1000 1000 1000 1000 1000
1000 1000 1000 1000 1000
average 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 0 1000 1000 1000 1000 1000 1000 1000 1000 1000
1000 1000 1000 1000 1000
variance 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PERCENTILES:

locus_0

TD Fs FD* FF* FD FF H B Q ZA Fst Kw Hw R2 S pi_w pi_b thetaWatt thetaTaj thetaFW D/Dmin H/Hmin maxhap
maxhap1 Rm
0.1% -1.88115 -12.9037 -2.91028 -3.01995 -2.70764 -2.73189 -34.3067 0 0 0.0141792 na 0.32 0.766667 0.0548533
13 0 0 3.44283 2.63333 0.533334 -0.737684 -0.368824 0.08 0.08 0
1.0% -1.57326 -9.18928 -2.1109 -2.19097 -2.30809 -2.47957 -14.4033 0 0 0.0547163 na 0.4 0.833333 0.06938
17 0 0 4.50217 3.40667 1.69 -0.605214 -0.223654 0.08 0.12 0
2.5% -1.38683 -7.95948 -1.76254 -1.86021 -1.84468 -1.87186 -11.27 0 0 0.077079 na 0.4 0.863333 0.0769878
20 0 0 5.29667 3.96 2.41667 -0.5263 -0.138479 0.12 0.12 1
5.0% -1.16859 -6.76538 -1.42408 -1.53295 -1.45447 -1.59022 -8.72333 0.025641 0.05 0.0919017 na 0.44 0.883333
0.0848728 23 0 0 6.09117 4.96667 3.12333 -0.443786 -0.118768 0.12 0.12 1
10.0% -0.928091 -5.05093 -1.02966 -1.0743 -1.11172 -1.20117 -6.04 0.037037 0.0714286 0.111726 na 0.48
0.9 0.0917654 26 0 0 6.88567 5.86667 4.31333 -0.354774 -0.0872859 0.12 0.16 1
50.0% -0.0494264 -1.59676 0.0200351 0.0131718 0.0134 -0.00123657 0.783333 0.125 0.205882 0.220047 na
0.6 0.946667 0.122479 37 0 0 9.79884 9.54 9.13 -0.0189366 0.012183 0.16 0.2 3
90.0% 0.790378 0.879629 0.821257 0.883204 0.927444 0.983939 4.07333 0.27027 0.375 0.358474 na 0.72 0.97
0.152882 52 0 0 13.7713 14.7 17.81 0.29551 0.0620819 0.28 0.32 5
95.0% 0.998518 1.55043 1.03484 1.10498 1.19105 1.24543 4.72 0.317073 0.428571 0.411504 na 0.76 0.976667
0.162257 57 0 0 15.0955 16.5467 20.05 0.377427 0.0711999 0.32 0.36 5
97.5% 1.18459 2.3811 1.17458 1.27042 1.34843 1.43784 5.56667 0.375 0.466667 0.451697 na 0.8 0.98 0.16749
61 0 0 16.1548 18.1933 22.97 0.44356 0.0777979 0.36 0.4 6
99.0% 1.45897 2.91104 1.33668 1.47486 1.52388 1.65796 6.34333 0.416667 0.522727 0.499242 na 0.84 0.983333
0.178815 68 0 0 18.0087 20.0133 25.1367 0.548307 0.0840853 0.36 0.44 6
99.9% 1.83042 4.56893 1.45024 1.61859 1.67698 1.85454 8.2 0.52381 0.590909 0.566158 na 0.88 0.99 0.192286
74 0 0 19.5977 23.5667 34.71 0.694189 0.0942726 0.44 0.52 7

average

TD Fs FD* FF* FD FF H B Q ZA Fst Kw Hw R2 S pi_w pi_b thetaWatt thetaTaj thetaFW D/Dmin H/Hmin maxhap
maxhap1 Rm
0.1% -1.88115 -12.9037 -2.91028 -3.01995 -2.70764 -2.73189 -34.3067 0 0 0.0141792 na 0.32 0.766667 0.0548533
13 0 0 3.44283 2.63333 0.533334 -0.737684 -0.368824 0.08 0.08 0
1.0% -1.57326 -9.18928 -2.1109 -2.19097 -2.30809 -2.47957 -14.4033 0 0 0.0547163 na 0.4 0.833333 0.06938
17 0 0 4.50217 3.40667 1.69 -0.605214 -0.223654 0.08 0.12 0
```

```
2.5% -1.38683 -7.95948 -1.76254 -1.86021 -1.84468 -1.87186 -11.27 0 0 0.077079 na 0.4 0.863333 0.0769878
20 0 0 5.29667 3.96 2.41667 -0.5263 -0.138479 0.12 0.12 1
5.0% -1.16859 -6.76538 -1.42408 -1.53295 -1.45447 -1.59022 -8.72333 0.025641 0.05 0.0919017 na 0.44 0.883333
0.0848728 23 0 0 6.09117 4.96667 3.12333 -0.443786 -0.118768 0.12 0.12 1
....etc.
```

The data showed are: observed values, probability for these values, number of valid iterations and percentiles for each statistic and locus.

Example of an output file under the option **print_matrixpol** set at 1:

```
mlcoalsim version 1.20 (20061210)
OUTPUT FILE: date Mon Dec 11 17:48:32 2006

Input data from the file:  ex00b.txt

Print_neuttest:  0
Print_matrixpol:  1
N_iterations:  10
Seed1:  7354

N_loci:  1
Factorn_chr:  1.000000
N_sites:  1050
N_samples:  25
Thetaw:  5
Heter_theta_alphag:  -1.000000
No_rec_males:  0
Recombination:  10
Heter_rec_alphag:  -1.000000


FASTA file:  locus 0, nsam 25, nsites 1050, mutations 22, iteration 1
>L0
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L2
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L3
AAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L4
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L5
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L6
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L7
AAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
.
.
.
>L21
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L22
AAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L23
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L24
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...

FASTA file:  locus 0, nsam 25, nsites 1050, mutations 20, iteration 2
>L0
AAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
>L1
AAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
...  etc.
```

In the sequence matrix, the nucleotide 'A' means the ancestral nucleotide. Note that a header with characteristics of the locus is included.

Example of an output file under the option **likelihood_line** set at 1:

```
Total TD[Total] FD[Total] Rm[Total] TD[0] TD[1] TD[2] TD[3] FD[0] FD[1] FD[2] FD[3] Rm[0] Rm[1] Rm[2]
Rm[3]
-95.2189 -30.412 -30.3514 -34.4555 -5.80084 -7.54452 -13.8175 -3.2491 -5.83754 -6.24713 -8.03477 -10.232
-9.01972 0 -13.8175 -11.6183
```

In this case, the statistics TD, FD and Rm were used to calculate the likelihood. Combined likelihoods for all TD values (for all loci) and for FD and Rm, as well for all values together are also given.

6 EVOLUTIONARY MODELS

The models included are (i) the neutral stationary panmictic model, (ii) the finite-island model, (iii) models where the population size changes across the time, (iv) refugia model and (v) deterministic positive selection. All these models can not be jointly used, with the exception of the neutral and the positive selection models for different independent loci and models that change the population size together with a finite-island model. In this case the same evolutionary processes operates in all loci and subpopulations.

6.1 Neutral stationary panmictic model.

This is the standard null model in population genetics. Only mutation and drift are considered, and recombination is included. Recombination must be expressed in $4Nr$ terms, where N is the effective population size and r is the recombination rate per locus.

Input example # 1:

```
seed1 786
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 25
npop 1

recombination 10
thetaw 10
mutations 0
factorn_chr 1
no_rec_males 0
```

6.2 Finite-island model.

In the finite-island migration model (Wright, 1931), the migration parameter M is symmetric and constant among islands ($M = 4Nm$, where N is the effective population size of the first defined sampled deme (subpopulation) and m the migration rate; each population was assumed to be sufficiently large to have one coalescent event per generation at the most). The number of the islands (subpopulations) remains fixed during the tree reconstruction.

The parameters used to indicate a finite island model are the following:

- **npop**: the total number of populations.

- **npop_sampled**: the number of populations sampled for each locus.

- **ssize_pop**: sample size of each population (separated by white spaces, comma and the next locus).

- **mig_rate**: $4Nm$. Compulsatory in case of more than one population.

- **factor_pop**: Population size of each population, in relation to the population size of the first indicated population (each value has to be separated by spaces or tabs). A value of 1 for all populations means that each population has the same population size than the first population.

- **ran_factorpop**: 1 if the relative population size of each population or refugia is randomly chosen before each iteration. Otherwise 0.

- **same_factorpop**: 1 if the relative population size of each population or refugia is fixed to 1 (where $N = 1$). Otherwise 0.

Input example # 2:

```
seed1 5685
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 55

recombination 10
thetaw 5
factorn_chr 1
no_rec_males 0


npop 10
npop_sampled 3
ssize_pop 20 10 25

mig_rate 5
factor_pop 1 0.1 1
ran_factorpop 0
same_factorpop 0
```

In this input example, the population mutation parameter **thetaw** indicates the level of variation per locus in the first defined population.

Warning: the output file will show the statistics values for the total sample, not considering the subdivision in population, except for F_{st} and π_B and π_W.

6.3 Models where the population size changes across the time.

Expansion, contraction, and bottlenecks are included. There are two options in this model: (i) the logistic curve or (ii) the instantaneous process; in case using a logistic curve (Fu, 1997), the population size changes with time following:

$$\left.\begin{array}{ll} N_t = N_0 & \text{if } t \leqslant t_0, \\ N_t = N_0 + \frac{N_1 - N_0}{1+e^{-\gamma(t+t_s-t_0-(\frac{t_1-t_0}{2}))}} & \text{if } t_0 < t < t_1 \\ N_t = N_1 & \text{if } t \geqslant t_1, \end{array}\right\} \tag{1}$$

Here N_t is the population size at time t (expressed in N_0 generations), N_0 is the population size at time t_0, and N_1 is the population size at time t_1. Here, $\gamma = 10/(t_1 - t_0)$. t_s is a measure of the slope of the curve and it is only available in the first defined event.

The specific parameters for these models are the following:

- **nintn**: Number of events. Each event will change the population size. When this value is 0, this module will not be used.
- **iflogistic**: 1 indicates that the null model follows a logistic curve, otherwise the changes in population size are instantaneous and only **nrec** is used (**npast** is useless).
- **ts**: used only in case of logistic growth. t_s is only available for the first event (that is, the most recent) and indicates the curve of the growth. For example, in case that the first event has a time duration of 0.1 and $t_s = 0.0$, the logistic grow shows its typical sigmoidal curve. If $ts = 0.05$, the curve is starting in the middle of the sigmoidal draw (showing an exponential-like curve) and continues for $0.1N_0$ generations (Figure 1).
- **nrec**: population size at the beginning of the event (recent), in relation to the population size N_0 (the population size at present time for the first population studied).
- **npast**: population size at the end of the event (past), in relation to the population size N_0.
- **tpast**: duration of the event in terms of $4N$ generations.

Input example # 3:

```
seed1 7864
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 20
npop 1

recombination 10
thetaw 5
factorn_chr 1
no_rec_males 0

nintn 3
iflogistic 1
ts 0.025
nrec 1 0.1 2
npast 0.1 0.1 2
tpast 0.05 0.01 1
```

Bottleneck is defined by including different consecutive events with different population sizes. This is the case for the above example. The population decrease from present to $0.05 \cdot 4N$ generations 10 times. Then, the population remains constant (population size of $0.1N$) for $0.01 \cdot 4N$

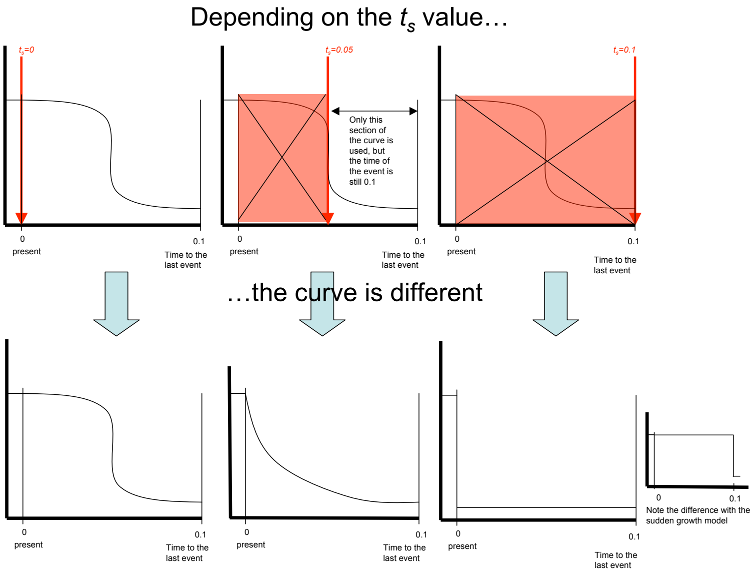
Depending on the t_s value…
t_s=0
t_s=0.05
t_s=0.1
Only this section of the curve is used, but the time of the event is still 0.1
0
present
0.1
Time to the last event
…the curve is different
Note the difference with the sudden growth model

Figure 1: Example of the use of parameter ts.

generations. Finally, the population changes to 2 times the present population for all the remaining time (Figure 2).

The (i) and (ii) models, that is, migration and change of the population size, can be jointly used. Nevertheless, this combined model is limited to change the population size for all populations at the same time.

6.4 Refugia model.

In this case, the current single population comes from several populations (refugias) that remained separated during a period of time. In the past, all the refugia came from a single population. The parameters are the following:

- **refugia**: type 1 if this model is used, otherwise 0.
- **npoprefugia**: number of refugia (subpopulations).
- **time_split**: time (in $4N$ generations) when the present population joined from all refugia from present to past.
- **time_scoal**: time (in $4N$ generations) when the population was splitted in several refugia from present to past.
- **factor_anc**: relative population size of the ancestral population, where N_0 is fixed at 1, in relative values.
- **freq_refugia**: average frequency contribution of each refugia to the current population (separated by white spaces or tabs).

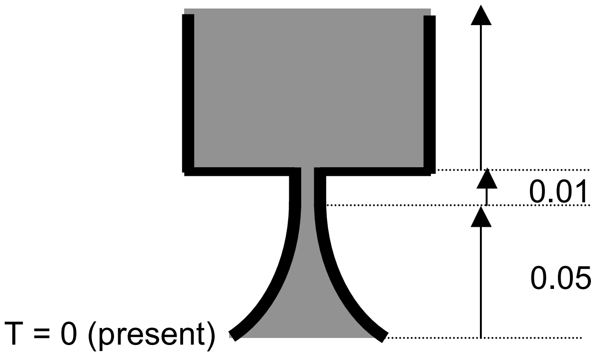

Figure 2: A representation of a population that changes the population size across the time.

- **mig_rate**: $4Nm$. migration level among refugia.

- **factor_pop**: population size in relation to the present population (1 means N for each population). Each value must be separated by white spaces or tabs.

- **ran_factor_pop**: 1 indicates that the relative population size of each population or refugia is randomly chosen before each iteration. Otherwise 0.

- **same_factor_pop**: 1 indicates that the relative population size of each population or refugia is fixed to 1 (where $N_0 = 1$). Otherwise 0.

Input example # 4:

```
seed1 659
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
thetaw 5
factorn_chr 1
no_rec_males 0

refugia 1
npoprefugia 2
time_split 0.0140
time_scoal 0.0165
factor_anc 0.1
freq_refugia 0.5 0.5

mig_rate 5
factor_pop 0.1 0.05
ran_factorpop 0
same_factorpop 0
```

6.5 Positive selection model.

Positive selection acts on a single position and the linked regions are affected by that process. The deterministic period is modeled in the selective phase in this program. Selection is considered out or inside the studied region, and complete or incomplete selective event can be modeled. We followed essentially the algorithm described in Braverman et al. (1995) to generate hitchhiking genealogies, and considering recombination within the locus of interest during the selective and neutral phases (see also Fay and Wu, 2000; Kim and Stephan, 2002; Przeworski, 2002).

The parameters are: the selection coefficient (s), the recombination rate between the selected locus and the studied locus (c), the intragenic recombination rate (r) and the time at which an advantageous mutation (t_f) is fixed. In our implementation of this model, we combined the recombination rates c and r (*i.e.* we used $4Nc + 4Nr$ as recombination parameter and therefore we needed the distance from the studied locus to the selected locus). For example, if $4Nc+4Nr = 110$ and the studied locus is 1000 nucleotides long and the selected position is 10000 nucleotides away from the studied locus, then $4Nc = 100$ and $4Nr = 10$. In the selective phase, we calculated directly the time to coalescent (instead of checking at small increments of time) for the selected and unselected population by using the reasoning of Nordborg (2001); equation 7. The selective phase starts at time t_s with a frequency of the selected allele of $1 - 1/2N$ and ends when the frequency of $x(t) < 1/N$. The value $x(t)$ was calculated deterministically using equation 1 in Kim and Stephan (2002) (see equation 3a in Stephan et al., 1992). Here, the parameter ϵ is fixed to $1/4Ns$. The computer code was tested by comparing the results of Table II in Stephan et al. (1992) and also by comparing some results with *ssw* program (Kim and Stephan, 2002).

The parameters used in this model are:

- **ifselection**: in case using the selective module type 1, otherwise 0.
- **pop_size**: Population size (N) of the present population (N remains always constant).
- **pop_sel**: The selective parameter $4Ns$, where s is the selective coefficient for the selected position.
- **sel_nt**: The selected position, starting from a value 0 in the first position of the studied locus. Higher or lower values than the length of the studied locus are allowed, indicating the selected position is out from the studied locus.
- **sinit**: indicates the time in $4N$ generations since the selective process finished (negative values are allowed and indicate that the selective process is not finished at present).

Input example # 5:

```
seed1 586
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10 "recombination for the 1050 sites"
thetaw 5
factorn_chr 1
no_rec_males 0
```

```
ifselection 1
pop_size 1E06
pop_sel 2E04
sel_nt -10000
sinit -0.00005
```

Here, an unfinished selective event ($0.04N$ generations before finished the deterministic phase) occurred at 10000 base pairs away from the studied region.

Models of positive selection on an previous neutral variant (standing variation) will be implemented in a future version.

7 THE MUTATIONAL PHASE

This program is designed to generate genetic data within-species, that is, the level of nucleotide variation within-population should not be too high - around 5% maximum - in order to avoid significant errors (on the contrary, a more sophisticated substitution model –not included here– should be used). For the same reason, it is recommended that he level of divergence with an outgroup species be not higher than 10-15%.

There are included three different methods to place mutations on the generated trees. In all methods we used sample data (n lines) obtained from a diploid species. The methods are as follows.

7.1 Methodology to place mutations.

7.1.1 Fix θ.

This is the standard coalescent method. A fixed value of the population mutation parameter θ ($4N\mu$) is used. A number of mutations according to the value of θ and the length of each tree is placed (according to a Poisson distribution).

Input example # 6:

```
seed1 4866
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
thetaw 5
factorn_chr 1
no_rec_males 0
```

Note that **thetaw** indicates the population mutation rate per locus ($\theta = 4N\mu$) in the defined population.

7.1.2 Fix the number of mutations.

This method place a fixed number of mutations on each tree (Hudson, 1993) generated under the models mentioned above.

Input example # 7:

```
seed1 759
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
thetaw 0
mutations 32
factorn_chr 1
no_rec_males 0
```

7.1.3 Take into account the uncertainty of θ.

There are included two kinds of distributions that consider the uncertainty of θ: the uniform and the gamma distributions. The uniform distribution can be used when the user has no information about the mutational rate; bounded values must be defined. The parameters used for these methods are the following:

- **range_thetant**: type 1 for using a uniform distribution and bound the θ values. Type 2 to use a log-uniform distribution and bound the θ values.

- **thetant_min**: in case **range_thetant** is 1, indicate the minimum level of variation (per nucleotide) considered. The same value will be used for all loci studied. A single value for the total loci or a value for each locus is allowed.

- **thetant_max**: in case **range_thetant** is 1, indicate the maximum level of variation (per nucleotide) allowed. The same value will be used for all loci studied. A single value for the total loci or a value for each locus is allowed.

- **ifgamma**: type 1 if the θ distribution is known and it is a gamma distribution. Otherwise type 0. The gamma function density depends on two parameters, α and p:

$$g(x) = \frac{p^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-px}, x \geqslant 0, \tag{2}$$

  and

$$\Gamma(\alpha) = \int_{0}^{\text{inf}} u^{\alpha-1} e^{-u} du. \tag{3}$$

  The mean and variance are: $E(x) = \frac{\alpha}{p}, Var(x) = \frac{\alpha}{p^2}$.

- **alpha_gamma** (α): in case **ifgamma** is 1, include for each locus (separated by spaces) the value of α in relation to variation per locus and not per nucleotide.

- **p_gamma** (p): in case **ifgamma** is 1, include for each locus (separated by spaces) the value of p in relation to variation per locus and not per nucleotide.

- **correct_gamma** (m): the result of the gamma distribution is multiplied by m. Then the expected value is $E(x) = \frac{\alpha}{p} m$. This value is specially useful when $\alpha = p$, because (in absence of m) the mean is 1, but the variance is reduced when $\alpha = p$ is enlarged. Thus, the expected θ value can be used to define m when $\alpha = p$. Use when **ifgamma** is 1, and include for each locus (separated by spaces) the value of m in relation to variation per locus and not per nucleotide.

Input example # 8:

```
seed1 4653
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
factorn_chr 1
no_rec_males 0

range_thetant 0
thetant_min 0.0005
thetant_max 0.05

ifgamma 1
p_gamma 1.0
alpha_gamma 1.0
correct_gamma 5.0
```

7.1.4 Fix the number of biallelic segregating sites, considering the uncertainty of θ.

It is included one method based on fixing the number of biallelic segregating sites considering the uncertainty of the value of θ ($FS\theta_{prior}$ methods).

The **R**ejection **A**lgorithm method (RA): This method employs the rejection algorithm #2 justified and described by Tavaré et al. (1997) and in Ramos-Onsins et al. (ress).
The RA method consider all possible θs and trees, but weight by their probabilities giving the number of observed biallelic segregating sites in the sample. A Gamma or a Uniform distribution for θ is enabled.

The methodology is as follows:

Let the parameter T be the vector of the coalescent times and Z the topology of the coalescent tree. $L_n = \sum_{k=1}^{n} nT_k$ is the total length of the coalescent tree (by summing the length of all branches) measured in units of $4N$ generations. Different coalescent iterations determine different values of T and Z, and therefore L_n.

Using a prior distribution of the mutation parameter θ (*e.g.*, uniform):

1. generate a vector of parameters $G = [\theta, T, Z]$,
2. compute the acceptance probability $g = \frac{Poisson(S,\theta L_n)}{Poisson(S,S)}$,
3. accept G with probability g, otherwise go to step 1,
4. output G and go to step 1. Exit the loop after i accepted rounds.

The parameters used for this method are the following:

- **sfix_allthetas**: 1 indicates the module is active.

- **method_samp**: 1 indicates RA.

- **mutations**: Note that in this case, this option refers to biallelic segregating sites instead of mutations. Another methods will be included in the future.

Also we have to consider those parameters defined in the subsection 7.1.3.

Input example # 9:

```
seed1 9871
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
thetaw 0
mutations 32
factorn_chr 1
no_rec_males 0

sfix_allthetas 1
method_samp 1

range_thetant 1
thetant_min 0.0005
thetant_max 0.05

ifgamma 0
p_gamma 0
alpha_gamma 0.0
correct_gamma 0.0
```

In this example, a sample with 32 biallelic segregating sites will be generated, assuming that the population mutation parameter is unknown, bounding the uniform distribution of θ values between 0.0005 and 0.05 mutations per nucleotide.
Extra outputs: When **sfix_allthetas** is defined, two extra output files are shown. The posterior probability of θ values per locus (file named equal than the output file including *_thetapost.out*), and the distribution of total length trees accepted (file named equal than the output file including *_treelength.out*).

7.2 Multiple hits module.

Although few tri- or tetra-allelic positions are expected in samples within species, multiple hits can affect importantly the distributions of tests with outgroup because the multiple hits can more probably occur in the long branch to the outgroup. As a consequence, the ancestral nucleotide might be erroneously assigned. When multiple hits are considered in the reconstruction of sample sequences, this effect is included in the distributions of the statistical tests.
To activate the option of multiple mutations in the same site (*i.e.,* finite sites model), it is necessary to include the distance (in relation to $4N$ generations) to an outgroup. In case fixing the number of biallelic segregating sites, the multiple hits module is only allowed when the parameter **sfix_allthetas** is active. The program will calculate the number of multiple hits in relation to the ratio transition/transversion. The model of nucleotide substitution here considered

is follows Kimura-2P (Kimura, 1980).
The parameters used for this option are:

- **mhits**: in case considering multiple hits type 1, otherwise type 0.

- **seed2**: seed for pseudorandom numbers. Mandatory in this option.

- **mhratio_sv**: the ratio of transition/transversion. s/v = 0.5 indicates no bias. **Important:** The mutational bias must be defined for each locus. Otherwise a value 0.5 will be assigned by default.

- **dist_out**: indicates the divergence, in $4N$ generations, of the studied population to an outgroup.

Input example # 10:

```
seed1 539
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
thetaw 10
mutations 0
factorn_chr 1
no_rec_males 0

mhits 1
seed2 546
mhratio_sv 0.5
dist_out 8
```

7.3 Heterogeneity in the mutation rate across the sequence.

Differences in mutation rate for different positions can be modeled using a gamma distribution. Here a gamma distribution with parameters $\alpha = p$ is used (see equations 2 and 3). For smaller values of α (*i.e.*, $\alpha < 1$), the distribution of values simulated resembles to a hotspot region, while large values of α (*i.e.*, $\alpha >> 1$) the distribution resembles to a region with uniform values. See also Figure 3 for a better understanding. Also, it is included an option to indicate the number of invariable sites. The parameters used are the following:

- **heter_theta_alphag**: in case not considering heterogeneity type 0 or a negative value, otherwise type the value of $\alpha = p$.

- **invar_mut_sites**: number of positions that are considered invariable. The algorithm will calculate (as an average) the number of invariable positions for each iteration.

Input example # 11:

```
seed1 1296
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
```

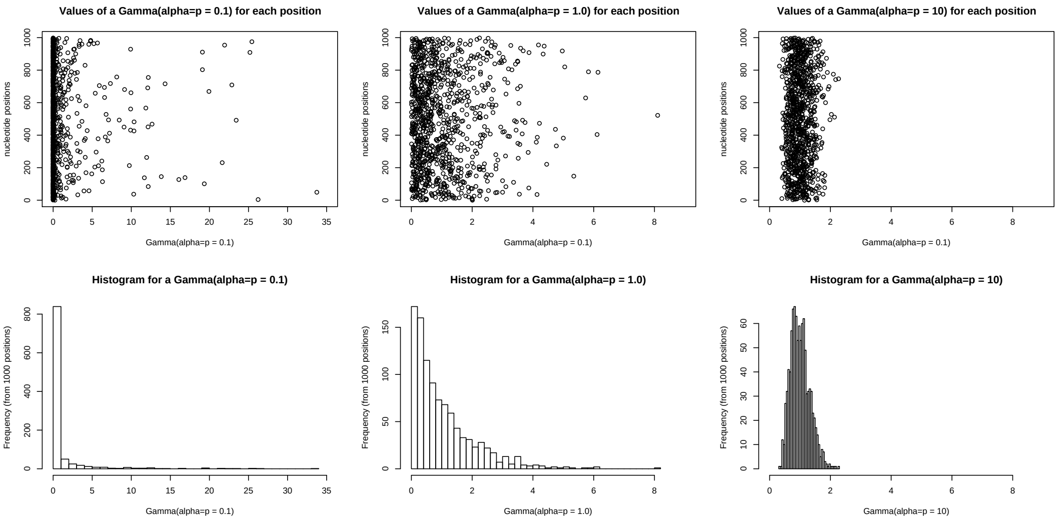

Figure 3: Heterogeneity in the mutation rate across the sequence. Values of θ/nt, considering a mean of 1.0 for a region of 1000 nucleotide positions. The plots above show the value of θ/nt in each position for different values of $\alpha = p$. Note that the left plot have different axis. The histograms below show the distribution of θ/nt values. For $\alpha = p < 1$, the heterogeneity would resemble a hotspot distribution (most have 0, but few have very high values), for $\alpha = p = 1$, we see an exponential distribution. For $\alpha = p >> 1$, the distribution is centered in 1 with small variance.

```
npop 1

recombination 10
thetaw 5

heter_theta_alphag 1.0
invar_mut_sites 630

factorn_chr 1
no_rec_males 0
```

8 RECOMBINATION PARAMETER

This section describes the necessary parameters to include the recombination parameter in simulations. There are three main options: (i) a fixed number of population recombination parameter R ($R = 4Nr$), (ii) a distribution of R, and (iii) conditioning on the minimum number of recombinants (Rm) (and optionally the number of haplotypes) of the sample taking into account the uncertainty of R.

8.1 Fix the population recombination parameter R.

A fixed value of the population recombination parameter R ($4Nr$) is used. A number of recombinations takes place according with the value of R and the length of each tree.

Input example # 12:

```
seed1 5698
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
thetaw 5
mutations 0
factorn_chr 1
no_rec_males 0
```

8.2 Take into account the uncertainty of the parameter *R*.

There are included two kinds of distributions to take into account the uncertainty of R: the uniform and the gamma distributions. The uniform distribution can be used when the user has no information about the recombination rate; bounded values must be defined. The parameters used for these methods are the following:

- **range_rnt**: type 1 for using a uniform distribution and bound the R values. Type 2 to use a log-uniform distribution and bound the θ values.

- **recnt_min**: in case **range_recnt** is 1, indicate the minimum level of R (per nucleotide) considered. The same value will be used for all loci studied. A single value for the total loci or a value for each locus is allowed.

- **recnt_max**: in case **range_recnt** is 1, indicate the maximum level of R (per nucleotide) considered. The same value will be used for all loci studied. A single value for the total loci or a value for each locus is allowed.

- **ifgammar**: type 1 in case the R distribution is known and it is a gamma distribution. Otherwise type 0. The gamma function density depends on two parameters, α and p (in this section named α_r and p_r), as indicated in the equations 2 and 3.

- **alpha_gammar** (α_r): in case **ifgammar** is 1, include for each locus (separated by spaces) the value of α_r in relation to population recombination parameter per locus and not per nucleotide.

- **p_gammar** (p_r): in case **ifgammar** is 1, include for each locus (separated by spaces) the value of p_r in relation to population recombination parameter per locus and not per nucleotide.

- **correct_gammar** (m_r): the result of the gamma distribution is multiplied by m_r. Then the expected value is $E(x) = \frac{\alpha_r}{p_r} m_r$. This value is specially useful when $\alpha_r = p_r$, because (in absence of m_r) the mean is 1, but the variance is reduced when $\alpha_r = p_r$ is enlarged. Thus, the expected R value can be used to define m_r when $\alpha_r = p_r$. Use when **ifgammar** is 1, and include for each locus (separated by spaces) the value of m_r in relation to variation per locus and not per nucleotide.

Input example # 13:

```
seed1 213
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

thetaw 10
factorn_chr 1
no_rec_males 0

range_rnt 0
recnt_min 0.0005
recnt_max 0.05

ifgammar 1
p_gammar 0.5
alpha_gammar 0.5
correct_gammar 5.0
```

8.3 Fix *Rm* considering the uncertainty of the parameter *R*.

It is included one method based on fixing the number of biallelic segregating sites taking into account the uncertainty of the value of R (FSR_{prior} methods).

The **R**ejection **A**lgorithm method (RA): This is a simple method to reject samples that do not have the same *Rm* value fixed by the user. Only those samples having the same *Rm* value will be counted and printed out. A Gamma or a Uniform distribution for R values is enabled as a prior.

The parameters used here are the following:

- **rmfix**: 1 indicates the module is active.
- **method_samp**: 1 indicates RA. Another methods will be included in the future.
- The parameters defined in the subsection 8.2.

Input example # 14:

```
seed1 4432
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

thetaw 10
factorn_chr 1
no_rec_males 0

rmfix 1
method_samp 1
Rm 2
```

```
range_rnt 1
recnt_min 0.0005
recnt_max 0.05

ifgammar 0
p_gammar 0.0
alpha_gammar 0.0
correct_gammar 0.0
```

In this example, a sample will be generated assuming that the population mutation parameter follows a uniform distribution of R with values ranging between 0.0005 and 0.05 mutations per nucleotide. Rm is fixed at 2.

8.4 Fix *Rm* and the number of haplotypes, considering the uncertainty of *R*.

Here we use the same methodology than the previous section, but the difference is that a fixed number of haplotypes is considered in the sample together with a fixed Rm value. Although the posterior distribution of R improves significantly, the use of both statistics is computationally very expensive in time.

The parameters used here are:

- **nhapl**: 0 indicates this parameter will not be used, an integer value > 0 will fix the number of haplotypes in the sample when **rmfix** is active.
- the parameters defined in the previous subsections (8.2 and 8.3).

Input example # 15:

```
seed1 9098
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

thetaw 10
factorn_chr 1
no_rec_males 0

rmfix 1
method_samp 1
Rm 2
nhapl 13

range_rnt 0
recnt_min 0.0005
recnt_max 0.05

ifgammar 1
p_gammar 1.0
alpha_gammar 1.0
correct_gammar 5.0
```

In this example, a sample will be generated assuming that the population mutation parameter follows a gamma distribution of R and only those samples with 13 haplotypes and $Rm = 2$ will be considered.

8.5 Heterogeneity in the recombination rate across the sequence.

Differences in recombination rate for different positions can be modeled using a gamma distribution. Here a gamma distribution with parameters $\alpha_r = p_r$ is used (see equations 2 and 3). For smaller values of α_r (*i.e.*, $\alpha_r < 1$), the distribution of values simulated resembles to a hotspot recombinant region, while large values of α_r (*i.e.*, $\alpha_r >> 1$) the distribution resembles to a region with uniform values. See also the previous Figure 3 for a better understanding. The parameter used is the following:

- **heter_rec_alphag**: in case not considering heterogeneity type 0 or a negative value, otherwise type the value of $\alpha_r = p_r$.

Input example # 16:

```
seed1 7787
print_matrixpol 0
print_neuttest 1

n_iterations 1000
n_loci 1
n_sites 1050
n_samples 30
npop 1

recombination 10
thetaw 5

heter_rec_alphag 0.5

factorn_chr 1
no_rec_males 0
```

9 COMBINING UNCERTAINTY IN MUTATIONAL AND RECOMBINATIONAL RATES

The combination of mutation and recombination rates, using a fixed value of θ or R, or using different distributions of θ or R are enabled. Also, it is enabled the combination of mutational and recombinational rates fixing the number of mutations and Rm (and optionally the number of haplotypes) considering prior distributions of the θ and R parameters.

Input example # 17:

```
seed1 1121
print_matrixpol 0
print_neuttest 1

n_iterations 100
n_loci 1
n_sites 1050
n_samples 30
npop 1

factorn_chr 1
no_rec_males 0

rmfix 1
Sfix_allthetas 1
method_samp 1

mutations 20
```

```
Rm 2

range_thetant 1
thetant_min 0.0005
thetant_max 0.05

ifgamma 0
p_gamma 10.0
alpha_gamma 10.0
correct_gamma 10.0

range_rnt 0
recnt_min 0.0005
recnt_max 0.01

ifgammar 1
p_gammar 1.0
alpha_gammar 1.0
correct_gammar 10.0

mhits 1
seed2 2871
mhratio_sv 2.0
dist_out 8
```

In this example, a sample will be generated assuming that the population mutation parameter follows a uniform distribution of θ values between 0.0005 and 0.01 mutations per nucleotide and the population recombination parameter follows a gamma distribution with parameters ($\alpha_r = 1, p_r = 1$ and $m_r = 10$). The samples will be conditioned to 20 biallelic segregating sites and $Rm = 2$. A RA methodology will be used to choose samples. Multiple hits are allowed.

10 MULTILOCUS ANALYSES

One of the main issues of *mlcoalsim* is the generation of samples and statistical tests for a set of multiple loci. There are two choices for multilocus analysis: (i) separated unlinked loci, or (ii) a single long region separated in several fragments.

Output files: It is important to note that in case having statistical tests outputs, the option 2 for **print_neuttest** enables the display of the average plus variance of of each statistic for all loci together, and the statistics for each locus separatedly.

10.1 Unlinked loci.

The number of loci must be defined and all the necessary parameters for each locus. Demographic models have to be defined only once, but positive selection must be defined for each locus.

Input example # 18:

```
seed1 670
print_matrixpol 0
print_neuttest 2

n_iterations 1000
n_loci 4
n_sites 1050 560 2456 1000
n_samples 30 10 24 12
npop 1

recombination 10 0 30 56
thetaw 10 5 35 8.5
factorn_chr 1 1 1 1
no_rec_males 0
```

```
ifselection 0 0 1 0
pop_size 1E06
pop_sel 0 0 2E04 0
sel_nt 0 0 200 0
sinit 0 0 -0.01 0
```

In this case, a neutral model is used for the loci #0, #1 and #3, and positive selection model is used for the locus #2.

Input example # 19:

```
seed1 671
print_matrixpol 0
print_neuttest 2

n_iterations 1000
n_loci 4
n_sites 1050 560 2456 1000
n_samples 30 10 24 12
npop 1

recombination 10 0 30 56
thetaw 0
mutations 20 11 46 17
factorn_chr 1 1 0.75 0.75
no_rec_males 0

sfix_allthetas 1
range_thetant 1
thetant_min 0.0005
thetant_max 0.05

ifgamma 0
p_gamma 0
alpha_gamma 0.0
correct_gamma 0.0

refugia 1
npoprefugia 2
time_split 0.0140
time_scoal 0.0165
factor_anc 0.1
freq_refugia 0.5 0.5

mig_rate 5
factor_pop 0.1 0.05
ran_factorpop 0
same_factorpop 0
```

Here, a rejection algorithm is used for a multilocus analyses at four loci considering a refugia model. Note that two loci are autosomal and two are linked to the X chromosome.

10.2 Linked loci.

Use this option in case having linked loci from a long region or to study a long region analyzing some regions separately. In this option, it is better to type the parameter **print_neuttest** the value 2 (the averages and variances are in this case mostly useless).

The necessary parameters for using this option are:

- **nlinked_loci**: type 1 in to study a region using sliding windows. Type the number of linked loci to study separated fragments. Otherwise type 0.

- **pos_linked**: in case **nlinked_loci** is $>$ 1, indicate the first and the last positions for each fragment (note that the first nucleotide is 0), then comma and define the next fragment.

- **linked_segsites**: in case **nlinked_loci** is > 1, it is possible to fix the number of segregating sites for each linked fragment. Indicate the number of segregating sites separated by spaces or tabs. This option is only allowed when sfix_allthetas (uncertainty in θ value) is activated.

- **linked_rm**: in case **nlinked_loci** is > 1, it is possible to fix the number of minimum recombinant events (Rm) for each linked fragment. Indicate the number of Rm separated by spaces or tabs. This option is only allowed when rmfix (uncertainty in R value) is activated.

- **linked_nhapl**: in case **nlinked_loci** is > 1, it is possible to fix the number of haplotypes for each linked fragment. Indicate the number of haplotypes separated by spaces or tabs. This option is only allowed when rmfix (uncertainty in R value) is activated.

- **displ**: in case **nlinked_loci** is 1, the sliding windows option is activated. **displ** indicates the distance that the starting position of a fragment is separated from the next.

- **window**: in case **nlinked_loci** is 1, **window**indicates the length of each studied fragment.

Warning: All the parameters will be defined for the complete region. The output file(s) will only show the results for the linked fragments.

Input example # 20:

```
seed1 672
print_matrixpol 0
print_neuttest 2

n_iterations 1000
n_loci 1
n_sites 12350
n_samples 12
npop 1

recombination 60
thetaw 98.3
mutations 0
factorn_chr 1
no_rec_males 0

nlinked_loci 5
pos_linked 12 500, 1056 2300, 5020 7050, 9590 10900, 11200 12349
displ 0
window 0

ifselection 1
pop_size 1E06
pop_sel 2E04
sel_nt 2000
sinit 0
```

The above example shows an analysis of five fragments included in region of 12350 nucleotides.

Input example # 21:

```
seed1 672
print_matrixpol 0
print_neuttest 2

n_iterations 100
n_loci 1
n_sites 1350
n_samples 12
npop 1

factorn_chr 1
no_rec_males 0

nlinked_loci 2
```

```
pos_linked 12 500, 1056 1349
linked_segsites 10 5
linked_rm 0 0
"in case n_locilinked = 1"
displ 0
window 0

range_thetant 1
thetant_min 0.005
thetant_max 0.01

ifgamma 0
p_gamma 1.0
alpha_gamma 1.0
correct_gamma 1.0

range_rnt 0
recnt_min 0.0005
recnt_max 0.05

ifgammar 1
p_gammar 1.0
alpha_gammar 1.0
correct_gammar 5.0

sfix_allthetas 1
rmfix 1
method_samp 1
```

The above example shows an analysis of two fragments included in region of 1350 nucleotides where the number of segregating sites and the Rm values where fixed for both regions.

11 OBTAINING *P*-VALUES FOR OBSERVED DATA

The researcher can contrast their observed statistics values with the distribution of simulated statistics values including the statistic values in the input file. The results will be printed in the file *name_output*_PPercentiles.out. The input file must include the name of the observed statistic plus '_obs' (note the exceptions bellow) as a parameter, where the first column indicates if the statistic in question will be included (1) or not (0). Then, the observed values must be typed and separated by spaces or tabs. A non-available value should be written with a value of -10000. A maximum precision of $\pm 10^5$ is used to calculate probabilities.

This option calculates:

- the probability of the observed value be higher or equal than the simulated values.
- the probability of the observed value be equal than the simulated values. This probability is calculated to facilitate the analysis in discrete distributions.
- the percentiles 0.1%, 1%, 2.5%,5%,10%,25%,50%,75%,90%,95%,97.5%,99% and 99.9% for all statistics for all loci, and for average and variance statistics.

The list of the parameters included in this option are the following:

- **TD_obs**: Tajima's D test (Tajima, 1989).
- **Fs_obs**: Fu's F_s test (Fu, 1997).
- **FDn_obs**: Fu and Li's D^* test without outgroup (Fu and Li, 1993).
- **FFn_obs**: Fu and Li's F^* test without outgroup (Fu and Li, 1993).
- **FD_obs**: Fu and Li's D test with outgroup (Fu and Li, 1993).

- **FF_obs**: Fu and Li's F test with outgroup (Fu and Li, 1993).
- **H_obs**: Fay and Wu's H test (Fay and Wu, 2000).
- **B_obs**: Wall's B test (Wall, 1999).
- **Q_obs**: Wall's Q test (Wall, 1999).
- **ZA_obs**: ZA statistic (Rozas et al., 1999).
- **Fs_obst**: Genetic differentiation among populations, F_{st} (Hudson et al., 1992).
- **Kw_obs**: Number of haplotypes (Strobeck, 1987; Fu, 1996; Depaulis et al., 2001; Wall, 1999) divided by the sample size.
- **Hw_obs**: Haplotype diversity (Depaulis and Veuille, 1998) divided by the sample size.
- **R2_obs**: Ramos and Rozas' R2 test (Ramos-Onsins and Rozas, 2002).
- **S_obs**: Number of biallelic segregating sites.
- **pi_w_obs**: Average nucleotide diversity within populations per locus (*e.g.,* Hudson et al., 1992).
- **pi_b_obs**: Nucleotide diversity among populations per locus (*e.g.,* Hudson et al., 1992).
- **thetaWatt_obs**: Watterson's estimate of nucleotide variation (θ) (Watterson, 1975).
- **thetaTaj_obs**: Tajima's estimate of nucleotide variation per locus (nucleotide diversity, π) (Tajima, 1983).
- **thetaFW_obs**: Fay and Wu's estimate of nucleotide variation (Fay and Wu, 2000).
- **D_Dmin_obs**: Tajima's D divided by the minimum Tajima's D given S (Schaeffer, 2002; Schmid et al., 2005).
- **Hnorm_obs**: Fay and Wu's H normalized (Zeng et al., 2006).
- **maxhap_obs**: Number of lines in the most common haplotype (Depaulis et al., 2003) divided by the sample size.
- **maxhap1_obs**: Number of lines in the most common haplotype (and adding a maximum of 1 biallelic site within) (Rozas et al., 2001)
- **Rm_obs**: Minimum number of recombinations of the sample (Hudson and Kaplan, 1985).
- **thetafl_obs**: Fu and Li's estimate of variability (Fu and Li, 1993).
- **thetaL_obs**: Zeng *et al.* estimate of variability (Zeng et al., 2006).
- **ZengE_obs**: Zeng *et al.* neutrality test (Zeng et al., 2006).
- **EW_obs**: Ewens-Watterson test (Watterson, 1978).
- **Fstw_obs**: Genetic differentiation among populations, F_{st} (Hudson et al., 1992). Here, the average differentiation among all populations is weighted by the number of samples per population.
- **Pwh_obs**: Pwh test (unpublished) finds differences between the two more differentiated contiguous groups in the mismatch distribution. First, all pairwise comparisons are sorted, then, two groups are formed (first is 1vs n-1 ... until n-1vs 1). For the first two groups formed, the difference between the highest

and lowest value within group is calculated for each group, and the difference between the two groups is stored ($with$). The difference between the minimum value in the second group and maximum value in the first group is calculated ($betw$). If $betw$ ¿ $with$, the value of $betw + with$ is calculated. The maximum value for all comparisons is the value of *Pwh*.

Input example # 22:

```
seed1 5474
print_matrixpol 0
print_neuttest 2

n_iterations 1000
n_loci 4
n_sites 1050 560 2456 1000
n_samples 30 10 24 12
npop 1

recombination 10 0 30 56
thetaw 10 5 35 8.5
factorn_chr 1 1 1 1
no_rec_males 0

ifselection 0 0 1 0
pop_size 1E06
pop_sel 0 0 2E04 0
sel_nt 0 0 200 0
sinit 0 0 -0.01 0

TD_obs 1 -1.18213 -1.91896 -2.19236 0.540555
FD_obs 1 -1.42122 -1.71123 -1.57737 1.59608
FF_obs 0 -1.56732 -1.99677 -2.09661 1.38405
Rm_obs 1 0 0 0 0
```

In this example, the observed data for four loci are given. Statistics with the first value of 1 are calculated (TD,FD,Rm) and 0 not calculated (FF). The rest of statistics are neither calculated.

12 CALCULATION OF LIKELIHOOD VALUES

The P-values are calculated according to the number of coincidences between the observed and the simulated values. That is, a probability of 0.1 will be obtained if TD_obs is equal to the 10% of values in the simulation. The likelihood is given as $2log(Pvalue)$ in the output file. A single line for the output is the only output showed (the user has to remember the parameter conditions for the given line). In case the user run several simulations with the same name in the output file, the results will be accumulated consecutively in different lines of the same file (adding, not replacing file). In case no coincidence is found, the program will assign a $Pvalue$ equal to $1/(niter + 1)$ in order to calculate the likelihood. Note that in case of no coincidence, if more iterations, lower likelihood values.

A new option is included in order to avoid no coincidence by non-discrete values. The comparison is considered coincidence when the simulated value is included in an interval centered on the observed value, the ”error” value given by the user (+/- the value given). This value has to be defined for each statistic and is the same for all loci in the same statistic.

The list of the parameters included in this option are the following:

- All parameters described in the previous section (**statistic_obs**).
- **likelihood_line**: set to 1 to calculate likelihood.
- **TD_err**: Interval error for Tajima's D test (Tajima, 1989).

- **Fs_err**: Interval error for Fu's F_s test (Fu, 1997).
- **FDn_err**: Interval error for Fu and Li's D^* test without outgroup (Fu and Li, 1993).
- **FFn_err**: Interval error for Fu and Li's F^* test without outgroup (Fu and Li, 1993).
- **FD_err**: Interval error for Fu and Li's D test with outgroup (Fu and Li, 1993).
- **FF_err**: Interval error for Fu and Li's F test with outgroup (Fu and Li, 1993).
- **H_err**: Interval error for Fay and Wu's H test (Fay and Wu, 2000).
- **B_err**: Interval error for Wall's B test (Wall, 1999).
- **Q_err**: Interval error for Wall's Q test (Wall, 1999).
- **ZA_err**: Interval error for ZA statistic (Rozas et al., 1999).
- **Fs_errt**: Interval error for genetic differentiation among populations, F_{st} (Hudson et al., 1992).
- **Kw_err**: Interval error for number of haplotypes (Strobeck, 1987; Fu, 1996; Depaulis et al., 2001; Wall, 1999) divided by the sample size.
- **Hw_err**: Interval error for haplotype diversity (Depaulis and Veuille, 1998) divided by the sample size.
- **R2_err**: Interval error for Ramos and Rozas' R2 test (Ramos-Onsins and Rozas, 2002).
- **S_err**: Interval error for number of biallelic segregating sites.
- **pi_w_err**: Interval error for average nucleotide diversity within populations per locus (*e.g.,* Hudson et al., 1992).
- **pi_b_err**: Interval error for nucleotide diversity among populations per locus (*e.g.,* Hudson et al., 1992).
- **thetaWatt_err**: Interval error for Watterson's estimate of nucleotide variation (θ) (Watterson, 1975).
- **thetaTaj_err**: Interval error for Tajima's estimate of nucleotide variation per locus (nucleotide diversity, π) (Tajima, 1983).
- **thetaFW_err**: Interval error for Fay and Wu's estimate of nucleotide variation (Fay and Wu, 2000).
- **D_Dmin_err**: Interval error for Tajima's D divided by the minimum Tajima's D given S (Schaeffer, 2002; Schmid et al., 2005).
- **H_Hmin_err**: Interval error for Fay and Wu's H normalized (Zeng et al., 2006).
- **maxhap_err**: Interval error for number of lines in the most common haplotype (Depaulis et al., 2003) divided by the sample size.
- **maxhap1_err**: Interval error for number of lines in the most common haplotype (and adding a maximum of 1 biallelic site within) (Rozas et al., 2001)
- **Rm_err**: Interval error for minimum number of recombinations of the sample
- **thetafl_err**: Interval error for Fu and Li's estimate of variability (Fu and Li, 1993).
- **thetaL_err**: Interval error for Zeng *et al.* estimate of variability (Zeng et al., 2006).
- **ZengE_err**: Interval error for Zeng *et al.* neutrality test (Zeng et al., 2006).

- **EW_err**: Interval error for Ewens-Watterson test (Watterson, 1978).
- **Fstw_err**: Interval error for genetic differentiation among populations, F_{st} (Hudson et al., 1992). Here, the average differentiation among all populations is weighted by the number of samples per population.
- **Pwh_err**: Interval error for Pwh test (unpublished) finds differences between the two more differentiated contiguous groups in the mismatch distribution (with at least one group exhibiting low differences within).

Input example # 23:

```
seed1 5474
print_matrixpol 0
print_neuttest 2

n_iterations 1000
n_loci 4
n_sites 1050 560 2456 1000
n_samples 30 10 24 12
npop 1

recombination 10 0 30 56
thetaw 10 5 35 8.5
factorn_chr 1 1 1 1
no_rec_males 0

ifselection 0 0 1 0
pop_size 1E06
pop_sel 0 0 2E04 0
sel_nt 0 0 200 0
sinit 0 0 -0.01 0

TD_obs 1 -1.18213 -1.91896 -2.19236 0.540555
FD_obs 1 -1.42122 -1.71123 -1.57737 1.59608
Rm_obs 1 0 0 0 0

likelihood_line 1
TD_err 0.2
FD_err 0.2
Rm_err 0.01
```

In this example, the likelihood values for TD, FD and Rm are calculated for each locus and statistic and for the combinations (*i.e.*, for TD, for all loci together, for the other statistics and for all together).

13 A COMPREHENSIVE EXAMPLE OF INPUT FILE

The following is an example of an input file with all the parameters defined in the program (input.txt):

```
"This is a comment:  input test at July 13th 2006"
"Seed to start coalescent simulations"
seed1 111

"print_matrixpol.  0:  does not show DNA sequences.  1:  DNA sequences in phylip format.  2:  ms format
(+mhits) 3:  phylip excluding mhits"
"print_neuttest Calculates and prints neutral tests.  0:  no calculatation.  1=avg+var 2=values/loci
+ avg+var"
"Using the sliding windows or liked loci, avg+var file is not very informative.  Then use option 2"
print_matrixpol 0
print_neuttest 1

"Basic parameters:  For more than 1 locus separate the values with spaces or tabs"
"npop indicates the number of populations studied"
```

```
n_iterations 100
n_loci 1
n_sites 1000
n_samples 20
npop 1

recombination 10
thetaw 5

"Likelihood option.  set at 1 only if the likelihood value is required."
likelihood_line 0

"Heterogeneity for mutation and recombination rates is allowed across the positions.  the value of alpha(=p)
must be given.  0 or less indicates that a uniform will be used.  A value (0,inf) ia allowed, being close
to zero extreme hotspot distribution, 1 an exponential distribution, and >> 1 close to constant."
"invariable_mut_sites indicates the number of sites that are considered invariable (it is used as a proportion)."
heter_rec_alphag -1
heter_theta_alphag -1
invar_mut_sites 0

"mhits:  in case of considering multiple hits type 1, otherwise 0"
"seed2:  add in case of mhits"
"mhratio_sv is the ratio transition/transversion, s/v = 0.5 indicates no bias.  The mutational bias must
be defined for each locus.  Otherwise will be assigned 0.5 by default."
"dist_out indicates the divergence, in 4No generations, of the studied species to the outgroup"
mhits 0
seed2 123
mhratio_sv 0.5
dist_out 8

"factor that correct each locus for population size considering the chromosomal dotation in relation
to 4N: e.g., An autosomal locus is 1 (4N), a locus located in the X chromosome has a factor of 0.75 (3N)."
"in case recombination is supressed in males, set no_rec_males at 1, otherwise type 0."
factorn_chr 1
no_rec_males 0

"to increase the speed of simulations, the user can eliminate recombination after a tlimit time (in 4N
generations).  tlimit 1 means that after 4N generations the recombination has a value of zero."
tlimit 1000

"To use in case of linked loci:  0 no calculation, 1 use this option with sliding windows, > 1 for a
number of separated fragments."
"pos_linked:  type the first and the last nt positions (the first nucleotide is zero), comma and the
next locus."
nlinked_loci 0
pos_linked 0
"in case n_locilinked = 1"
displ 0
window 0



"in case the theta distribution is a gamma, set ifgamma to 1.  In case using a uniform distribution of
theta values, set range_thetant to 1.  Type 2 to use a log-uniform distribution."
"In case of using a uniform distribution:  In order to limit the theta values to biologically realistic
values, set range_thetant at 1 and specify the limits of the theta per nucleotide values and the same
values will be used for all the loci studied."
"in case the theta distribution is a known gamma distribution, set ifgamma at 1 and include FOR EACH
LOCUS (separated by spaces) the value of p and alpha parameters in relation to variation per locus (NOT
per nucleotide)"
"correct_gamma multiplies the result of the gamma distribution for the indicated value."
range_thetant 0
thetant_min 0.0005
thetant_max 0.05
ifgamma 0
alpha_gamma 0.0
p_gamma 0.0
correct_gamma 0.0

"in case the recombination distribution is a gamma, set ifgammar to 1.  In case using a uniform distribution
of R values, set range_rnt at 1.  Type 2 to use a log-uniform distribution."
"In case of using a uniform distribution:  In order to limit the R values to biologically realistic values,
set range_rnt at 1 and specify the limits of the R per nucleotide values and the same values will be
used for all the loci studied."
```

```
"in case the R distribution is a known gamma distribution, set ifgammar at 1 and include FOR EACH LOCUS
(separated by spaces) the value of p and alpha parameters in relation to variation per locus (NOT per
nucleotide)"
"correct_gammar multiplies the result of the gamma distribution for the indicated value."
range_rnt 0
recnt_min 0.0005
recnt_max 0.05
ifgammar 0
alpha_gammar 0.0
p_gammar 0.0
correct_gammar 0.0

"sfix_allthetas:  use this parameter in case conditioning over S and given a distribution of theta values
(otherwise 0).  rmfix must be 1 when conditioning on Rm (and alternatively the number of haplotypes)."
sfix_allthetas 0
mutations 0
rmfix 0
Rm 2
nhapl 0

"in case sfix_allthetas or/and rmfix be 1(defined), the method for obtaining samples conditioning values
must be defined using method_samp:  1 indicates RA."
method_samp 1

"WARNING: each model is incompatible with the others, with the exception of selection plus neutral stationary
model and migration plus changes in population size."

"ifselection:  in case of selection type 1, otherwise 0"
"selection, we need N (pop_size), 4Ns (pop_sel), sel_nt is the position of selected nt (negative values
are allowed)"
"sinit is the time in 4N generations since the selective process finished.  Negative values are allowed
and indicate unfinished processes."
"In case of selection, the recombination parameter value has a different meaning than above:  recombination
value for the studied region and all the region until the selective position"
ifselection 0
pop_size 1E06
pop_sel 2E04
sel_nt -10000
sinit 0

"some subdivision parameters:  define in case of more than one population"
"npop_sampled, set the number of pops studied for each locus"
"ssize_pop = sample size of each pop separated by spaces, comma and the next locus."
npop_sampled 1
ssize_pop 20

"Refugia parameters:  refugia 1 if used, otherwise 0"
"npoprefugia:  number of refugia"
"time_split:  from present to past, time in 4N generations the present population joined from all refugia."
"time_scoal:  from present to past, time in 4N generations the population was splitted in several refugia."
"factor_anc:  relative population size of the ancestral population, where No is 1."
"freq_refugia:  average frequency contribution of each refugia to the present population (separated by
spaces or tabs)."
refugia 0
npoprefugia 2
time_split 0.0140
time_scoal 0.0165
factor_anc 0.1
freq_refugia 0.5 0.5

"Define in case of refugia or subdivision:"
"mig_rate = 4NNm.  In case of more than one population, migration is needed."
"factor_pop = Population size in relation to N. 1 means N for each population or refugia (separated by
spaces or tabs)."
"ran_factor_pop:  1 indicates that the relative population size of each population or refugia is randomly
chosen before each iteration.  Otherwise 0"
"same_factor_pop:  indicates that the relative population size of each population or refugia is fixed
on 1 (where No = 1).  Otherwise 0"
mig_rate 0
factor_pop 1 0.1
ran_factorpop 0
same_factorpop 0
```

```
"Changes in N: set nintn at 0 if is not used."
"iflogistic 1 indicates that a logistic curve is used, otherwise (iflogistic 0) the changes in N are
instantaneous and only nrec is used (npast is useless in this last case)"
"in case logistic growth, ts can be defined:  ts is only available for the FIRST event (from present)
and indicates the curve of the growth; for example:"
"in case the first event has a duration 0.1 and ts=0.0, the logistic grow shows a typical sigmoidal curve;
if ts=0.05, the curve is starting in the middle (showing an exponential-like curve) but continues for
0.1 (!)  No generations."
"nintn = number of events."
"nrec = relative pop size at the beggining of the event (recent)"
"npast = relative pop size at the end of the event (past)"
"tpast = duration of the event in 4N generations"
iflogistic 1
ts 0.0
nintn 0
nrec 1
npast 1
tpast 0.15

"Observed values for statistics"
"The first column indicates if Pvalues are calculated (1) or not (0)."
"After the first column, include the values of each locus, separated by spaces or tabs.  A maximum precission
of +- 1e-06 is used."
"Note that the statistics FD* and FF* are replaced by FDn and FFn, and D/Dmax and H/Hmax by D_Dmin and
H_Hmin, respectively."

TD_obs 0 -0.0494264
Fs_obs 0 -1.59676
FDn_obs 0 0.0200351
FFn_obs 0 0.0131718
FD_obs 0 0.0134
FF_obs 0 -0.00123657
H_obs 0 0.783333
B_obs 0 0.125
Q_obs 0 0.205882
ZA_obs 0 0.220047
Fst_obs 0
Kw_obs 0 0.6
Hw_obs 0 0.946667
R2_obs 0 0.122479
S_obs 0 37
pi_w_obs 0 0
pi_b_obs 0 0
thetaWatt_obs 0 9.79884
thetaTaj_obs 0 9.54
thetaFW_obs 0 9.13
D_Dmin_obs 0 -0.0189366
H_Hmin_obs 0 0.012183
maxhap_obs 0 0.16
maxhap1_obs 0 0.2
Rm_obs 0 3

"In case likelihood_line 1, then a interval for coincidence of the observed value (+/- the value) is
required for each statistic used."
TD_err 0
Fs_err 0
FDn_err 0
FFn_err 0
FD_err 0
FF_err 0
H_err 0
B_err 0
Q_err 0
ZA_err 0
Fst_err 0
Kw_err 0
Hw_err 0
R2_err 0
S_err 0
pi_w_err 0
pi_b_err 0
thetaWatt_err 0
thetaTaj_err 0
```

```
thetaFW_err 0
D_Dmin_err 0
H_Hmin_err 0
maxhap_err 0
maxhap1_err 0
Rm_err 0
```

In this example of input file, all the parameters defined in the program are included, although most of them are inactivated. This input file will generate 100 iterations for a sample of 20 lines of a single locus of 1000 nucleotides with recombination 10 per locus and a fixed population mutation rate of 5 per locus. The standard neutral model will be used and the output will show a number of statistical tests calculated for each generated sample.

14 OTHER APPLICATIONS

mlcoalsim encompass two scripts written in perl to help in the analysis of the results of coalescent simulations. Remember to install perl in your computer before running the scripts.It is advisable to run the simulations and the perl scripts in the same computer in order to avoid problems with the characters of carriage returns.

14.1 Calculating the statistical power for statistical tests of neutrality.

A perl script to calculate the statistical power of the tests is available. This script is useful in case analyzing the confidence and the power of some test given alternative models.

The usage of this script (calc_power_chose.pl) is:

```
perl calc_power_chose.pl -std (file with null model) -dir (directory with alternative
files.out) -perc0 (proportion [0,0.5] to calculate power) -perc1 (proportion [0,0.5]
to calculate power)
```

This script enables the calulation of the statistical power for a number of *mlcoalsim* output files (with the extension .out), for two different probabilities (*perc0* and *perc1* options).
Example:

```
perl calc_power_chose.pl -std L1n20nullmodel.out -dir ./ -perc0 0.025 -perc1 0.050
```

It will generate a results file with the calculation of the statistical power of each neutrality test for the alternative models (located in the current directory) in relation to the null model.

14.2 Calculating the probability of observed values for statistical tests of neutrality.

A perl script to calculate the probability of the neutrality tests is available. Although this script is redundant to the option indicated in section 11, it is useful in case analyzing alternative data. Note that the precision of the values is not taking into account, so results might be some different to the results obtained using the parameters in section 11.

The usage of this script (probstats.pl) is:

```
perl probstats.pl -obs (file with observed values) -sim (file with simulated values)
```

This script calculates the probabilities (higher than and lower than, separately) for all the statistics present in the output of simulated values. The file with the observed values must have the same header (the line indicating all the statistics) than the *mlcoalsim* output file, and the observed values in the same position one line below (with the same number of tabulators).
Example:

```
perl probstats.pl -obs obsvalues.out -sim simsfile.out
```

It will generate a results file with the calculation of the probabilities for each neutrality test of the observed values given the simulated values.

14.3 Calculating composite probabilities.

This perl script was constructed by Pablo Librado under the supervision of M. Aguadé and S. Ramos-Onsins. This script calculates the composite probabilities for several statistics together, or with the same statistic for several loci together. This script contains its own documentation in the folder "output_selector" (output_selector4.doc).

The usage of this script (output_selector4.pl) is:

```
perl output_selector4.pl -cols [number of the column to use] -in [mlcoalsim output
file] -obs [file with obs values] -out [output file] -emp [y/n]
```

This script enables the calculation of the composite probability for a number of *mlcoalsim* output files given a file with the observed data.
Example:

```
perl output_selector4.pl -cols 9:24 -in simulations.out -obs obsvalues.txt -out
results.txt -emp y
```

It will generate a results file with the calculation of the composite probability for the observed data in comparison to the simulated data.

14.4 Running a large number of different scenarios and calculating the likelihood.

14.4.1 Grid:

Here we include a perl script that construct a grid for a number of parameters in the range of values chosen by the user. The results will be added in two files (file with likelihoods and file with parameters). Remember that the option *likelihood_line* must be 1.
The usage of this script (mlcoalsim_grid.pl) is:

```
perl mlcoalsim_grid.pl -infile (path to the initial mlcoalsim input file, with NO
comments) -num_par (number of parameters to modify) -outfile (name of the output
file) -parfile (name of the parameters output file -os (pc, linux, mac)
```

This script calculates the likelihood for a number of *mlcoalsim* output files given observed data. Example:

```
perl mlcoalsim_grid.pl -infile ex01_input_likelihood.txt -num_par 2 -outfile ex01grid.out
-parfile ex01par_grid.out -os mac < ragepar_grid_ex01.txt > ex01res_grid.out
```

It will generate a output file with all likelihood valuesand an additional ouptut file with all the parametric conditions used. In this example (with only 100 iterations per simulation), we have the following result (here plotted in Figure 4):

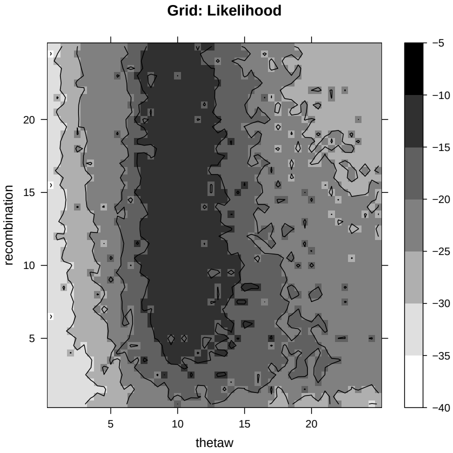

Figure 4: A representation of the likelihood values across all the studied parametric space.

14.4.2 MCMC:

Here we include a perl script that construct a posterior distribution for a number of parameters in the range of values chosen by the user. The results will be added in two files (file with likelihoods and file with parameters). Remember that the option *likelihood_line* must be 1.
The usage of this script (mlcoalsimgrid.pl) is:

```
perl mlcoalsim_mhmcmc.pl -infile (path to the initial mlcoalsim input file, with
NO COMMENTS) -numpar (number of parameters to modify) -burniter (burn-in iterations)
-mcmciter (iterations in the chain) -mlcoalsimout (mlcoalsim output file name)
-mcmcout (MHMCMC output file name) -parfile (name of the parameters output file
-os (pc, linux, mac)
```

This script calculates the likelihood for a number of *mlcoalsim* output files given observed data, and gives as a result a distribution of the accepted parameters.
Example:

```
perl ./mlcoalsim_mhmcmc.pl -infile ./ex01_input_likelihood.txt -numpar 2 -mlcout
ex01allsims2.out -burniter 500 -mcmciter 4500 -mcmcout ex01acceptedmcmc2.out -parfile
ex01par_mcmc2.out -os linux < rangepar_mcmc_ex01b.txt > ex01res_mcmc2.out
```

It will generate a output file with all likelihood values for the accepted parameters and an additional ouptut file with all the parametric conditions used. In this example (with only 100 iterations per simulation), we obtained the following result (here plotted in Figure 5):

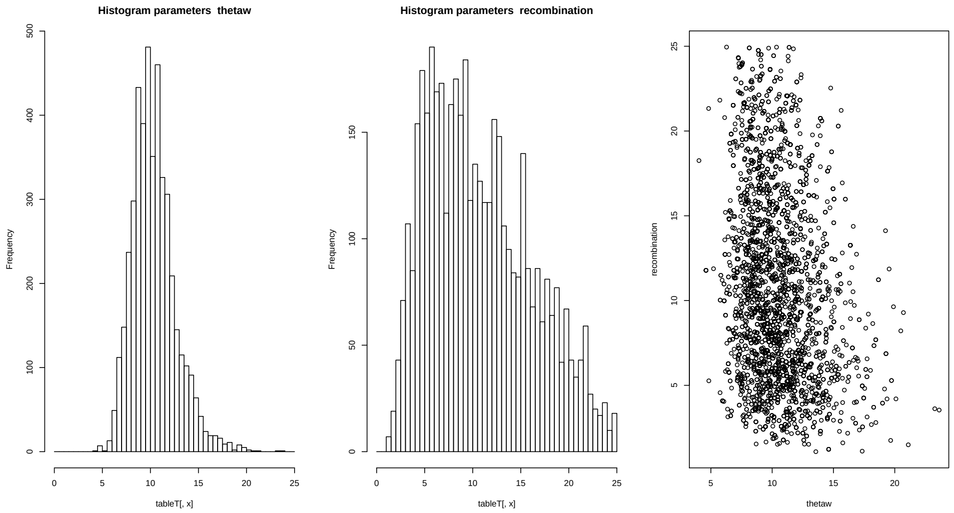

Figure 5: Posterior distribution of $4N\mu$ and $4Nr$ parameters using a variant of MHMCMC.

14.5 Running a large number of different scenarios and calculating statistics.

The same perl script that construct a grid for a number of parameters in the range of values chosen by the user is used (mlcoalsim_grid.pl). In this case, the same procedure must be made than for calculating likelihood values (that is, include an input file with the general variables, see the previous section) with the exception that the variable ”n_iter” must be set at the value 1 (and also set the variable ”likelihood_line” at 1, although likelihood is not calculated here). The results will be added in two files (file with likelihoods and file with parameters).

15 OTHER TECHNICAL INFORMATION

The generation of deviates from uniform, Binomial, Poisson, and Gamma distributions, and the finding of roots of functions were based on bibliography (Press et al., 1992; Press and Teukolsky, 1992; Lanczos, 1964; Cheng and Feast, 1979; Atkinson, 1979; Fishman, 1979; Ridders, 1979).

16 UPDATES, BUGS AND OTHER DETECTED PROBLEMS

- versions 1.06 and previous have an error calculating multiple hits events. Fixed on Nov 14th, 2006.
- versions from 1.05 to 1.08 have a error on rejection algorithm module. Fixed on Dec 11th, 2006.

- versions 1.08 and previous have an error in the selection module. It affects only those parameters with "sinit" negative in an small range of values
- The version 1.20 have changed the meaning of recombination for selective models. Now the value of recombination indicates the region studied and not all the length until the selected position.
- version 1.20 and previous had an error calculating R2 statistic. Fixed in version 1.205 and in version 1.095 on Feb 6th, 2007.
- versions from 1.20 to 1.208 had a typographical error in the "_PPercentile" output. Fixed on Feb 11th, 2007.
- version 1.21 changed the maximum precision value for comparing observed vs simulated values to 10^{-5} instead 10^{-6}. Changed on February 17th, 2007.
- version 1.23 changed an output mistake in the header (ts value should not be printed in some cases).
- version 1.24 modified the precision values for the F_s statistic. A warning is included in documentation. For large samples (*e.g.*, 300 lines), the F_s statistic can not be exactly calculated.
-version 1.25 detected a bug in the recombination module when uncertainty in θ is active and mutations are fixed, and at the same time is active the uncertainty in the population recombination parameter but Rm is undefined. Fixed on July 5th, 2007.
-version 1.26 detected a bug printing the recombination or θ values used in simulations. Specifically, the program did not print the used values when a distribution was given but no condition (S or R_m) was defined.
-version 1.27 Bug detected in compilations for Fedora. Error in precision. Example ex05.txt did not work. Bug detected in the refugia model in specific conditions. Fixed on July 9th, 2007. Thanks to Li-Guo Wang (Beijing Genomics Institute).
-version 1.28 Versions from 1.20 to 1.27 had a bug in multiple hits module. Thanks to Marta Mele (UPF, Barcelona). Fixed Aug 19th, 2007.
-version 1.29 included a message and a break when the number of sites is higher than the number of mutations and no multiple hits is allowed. Thanks to Felipe Martins. Changed on December 1st, 2007.
-version 1.37 Updated version including fixing the number of biallelic segregating sites instead of mutations. Also, it is included a number of new statistics (thetaFL, thetaL, Hnorm, Fstw, Pwh) and the option to calculate likelihood values for observed data.
-version 1.38. A bug in conditioning for mutations was fixed. A log-uniform distribution for θ and for R is included.
-version 1.39. A bug in printing the matrix of mutations when θ or R are fixed was found. Also some other errors in printing headers were fixed. Thanks to Pavlos Pavlidis (LMU, Munich).
version 1.40. A bug in managing loci with S=0 was detected and fixed. The scripts for running a large number of different scenarios and calculating the likelihood were modified and some bugs were fixed. The algorithm for the unpublished statistic Pwh contained a bug and was modified.

17 SOFTWARE CITATION

Sebastian E. Ramos-Onsins and Thomas Mitchell-Olds. *mlcoalsim*: Multilocus Coalescent Simulations. *Evolutionary Bioinformatics* 2007:**2** 1-4.

18 ACKNOWLEDGMENTS

Many thanks to all people who contributed to improving and debugging this application program. Many thanks to people from the labs of M. Aguadé, W. Stephan and T. Mitchell-Olds. Thanks to J. Rozas for his invaluable help. Thanks in special to A. Sánchez-Gracia, A. Amador, F. Vieira, H. Quesada, D. Orengo, S. Guirao, A. Vilella, P. Librado, and M. Aguadé from Barcelona, L. Ometo, S. Mousset, D. de Lorenzo and W. Stephan from Munich and K. Schmid, J. de Meaux and C. Wheat from Jena. Thanks to D. Posada and one anonymous reviewer for very useful comments and suggestions. Thanks also to Y. Kim for helping with the selective model. Also thanks to Anna Palmé. This work is partially supported by 'Distinció per la Promoció de la Recerca Universitària' from Generalitat de Catalunya to M. Aguadé and grant BFU2004-02253 from DGICYT, Spain to M. Aguadé and by the Max-Planck Gesselschaft.

19 REFERENCES

Atkinson, A. C. (1979). The computer generation of poisson random variables. *Appl. Statist. 28:1*, 29–35.

Braverman, J. M., R. R. Hudson, N. L. Kaplan, C. H. Langley, and W. Stephan (1995). The hitchhiking effect on the site frequency spectrum of DNA polymorphisms. *Genetics 140*(2), 783–796.

Cheng, R. C. and G. M. Feast (1979). Some simple gamma variate generators. *Appl. Statist. 28:3*, 290–295.

Depaulis, F., S. Mousset, and M. Veuille (2001). Haplotype tests using coalescent simulations conditional on the number of segregating sites. *Mol. Biol. Evol. 18*(6), 1136–1138.

Depaulis, F., S. Mousset, and M. Veuille (2003). Power of neutrality tests to detect bottlenecks and hitchhiking. *J. Mol. Evol. 57 Suppl 1*, S190–200.

Depaulis, F. and M. Veuille (1998). Neutrality tests based on the distribution of haplotypes under an infinite-site model. *Mol. Biol. Evol. 15*(12), 1788–1790.

Fay, J. C. and C.-I. Wu (2000). Hitchhiking under positive Darwinian selection. *Genetics 155*(3), 1405–1413.

Fishman, G. S. (1979). Sampling from the binomial distribution on a computer. *J. Am. Statist. Ass. 74:366*, 418–423.

Fu, Y.-X. (1996). New statistical tests of neutrality for DNA samples from a population. *Genetics 143*, 557–570.

Fu, Y.-X. (1997). Statistical tests of neutrality of mutations against population growth, hitchhiking and background selection. *Genetics 147*(2), 915–925.

Fu, Y. X. and W. H. Li (1993). Statistical tests of neutrality of mutations. *Genetics 133*(3), 693–709.

Grassly, N. C., J. Adachi, and A. Rambaut (1997). Pseq-gen: an application for the monte carlo simulation of protein sequence evolution along phylogenetic trees. *Computer Applications in the Biosciences 13(5)*, 559–560.

Hudson, R. R. (1993). The how and why of generating gene genealogies. In N. Takahata and A. Clark (Eds.), *Mechanisms of Molecular Evolution*, pp. 23–36. Sunderland, MA: Sinauer Assoc.

Hudson, R. R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics 18*(2), 337–338.

Hudson, R. R., K. Bailey, D. Skarecky, J. Kwiatowski, and F. J. Ayala (1994). Evidence for positive selection in the *superoxide dismutase* (*sod*) region of *Drosophila melanogaster. Genetics 136*(4), 1329–1340.

Hudson, R. R. and N. L. Kaplan (1985). Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics 111*(1), 147–164.

Hudson, R. R., M. Slatkin, and W. P. Maddison (1992). Estimation of levels of gene flow from DNA sequence data. *Genetics 132*(2), 583–589.

Kim, Y. and W. Stephan (2002). Detecting a local signature of genetic hitchhiking along a recombining chromosome. *Genetics 160*(2), 765–777.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitution through comparative studies of nucleotide sequences. *J. Mol. Evol. 16*, 111–120.

Lanczos, C. (1964). A precision approximation of the gamma function. *J. SIAM 1*, 86–96.

Nordborg, M. (2001). Coalescent theory. In D. Balding, M. Bishop, and C. Cannings (Eds.), *Handbook of Statistical Genetics*, pp. 179–212. Chichester: John Wiley & Chichester Sons.

Press, W. and S. Teukolsky (1992). Portable random number generators: 6 (5), 522. *Computers in Physics 6*, 522–524.

Press, W. H., W. T. Teukolsky, S. A. ans Vetterling, and B. P. Flannery (1992). *Numerical Recipes in C. The Art of Scientific Computing.* Cambridge University Press.

Przeworski, M. (2002). The signature of positive selection at randomly chosen loci. *Genetics 160*(3), 1179–1189.

Ramos-Onsins, S. E., S. Mousset, T. Mitchell-Olds, and W. Stephan (in press). Population genetic inference conditioning on the number of segregating sites - a reassessment. *Genetical Research* -, –.

Ramos-Onsins, S. E. and J. Rozas (2002). Statistical properties of new neutrality tests against population growth. *Mol. Biol. Evol. 19*(12), 2092–2100.

Ridders, C. J. F. (1979). A new algorithm for computing a single root of a real continuous function. *IEEE Transactions on Circuits and Systems 26:11*, 979–980.

Rozas, J., M. Gullaud, G. Blandin, and M. Aguade (2001). DNA variation at the *rp49* gene region of *Drosophila simulans*: evolutionary inferences from an unusual haplotype structure. *Genetics 158*(3), 1147–1155.

Rozas, J., C. Segarra, G. Ribo, and M. Aguadé (1999). Molecular population genetics of the rp49 gene region in different chromosomal inversions of drosophila subobscura. *Genetics 151*(1), 189–202.

Schaeffer, S. (2002). Molecular population genetics of sequence length diversity in the *Adh* region of *Drosophila pseudoobscura. Genetical Research 80*, 163–175.

Schmid, K. J., S. E. Ramos-Onsins, H. Ringys-Beckstein, B. Weisshar, and T. Mitchell-Olds (2005). A multilocus sequence survey in *Arabidopsis thaliana* reveals a genome-wide departure from a neutral model of dna sequence polymorphism. *Genetics 169*, 1601–1615.

Stephan, W., T. Wiehe, and M. W. Lenz (1992). The effect of strongly selected substitutions on neutral polymorphism - analytical results based on diffusion theory. *Theor. Popul. Biol. 41*(2), 237–254.

Strobeck, C. (1987). Average number of nucleotide differences in a sample from a single subpopulation: a test for population subdivision. *Genetics 117*, 149–153.

Tajima, F. (1983). Evolutionary relationship of DNA sequences in finite populations. *Genetics 105*(2), 437–460.

Tajima, F. (1989). Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics 123*(3), 585–595.

Tavaré, S., D. J. Balding, R. C. Griffiths, and P. Donnelly (1997). Inferring coalescence times from DNA sequence data. *Genetics 145*(2), 505–518.

Wall, J. (1999). Recombination and the power of statistical tests of neutrality. *Genet Res 74*, 65–79.

Wall, J. D. (2000). A comparison of estimators of the population recombination rate. *Mol. Biol. Evol. 17*, 156–163.

Watterson, G. (1975). On the number of segregating sites in genetical models without recombination. *Theor. Popul. Biol. 7*, 256–276.

Watterson, G. (1978). The homozygosity test of neutrality. *Genetics 88*, 405–417.

Wright, S. (1931). Evolution in mendelian populations. *Genetics 16*, 97–159.

Zeng, K., Y.-X. Fu, S. Shi, and C.-I. Wu (2006). Statistical tests for detecting positive selection by utilizing high-frequency variants. *Genetics 174*, 1431–1439.