SUMMARY
Manual for VarEff (Variation of effective population size) package: R software

VERSION 1.2

Natacha Nikolic and Claude Chevalet

(INRAE)

This package works with R software (version 3.0.2 or later), the library(mcmc) and library(grDevices). Load VarEff, mcmc, and grDevices.

VarEff file (.zip) can be install on R manually using "Packages -> Install package(s) from local zip files" from menu bar.

Table 1. Functions of VarEff package.

Functions	**Description**	**File used**	**Files created**
Theta()	Estimates the mean of current (Theta0), intermediate (Theta1) and ancestral (Theta2) theta values, asks for parameters and builds a script to run VarEff (“job.R”).	infile.txt	job.Thet0 job.R
VarEff()	Model VAREFF. Applies the method to get the posterior distribution of *Ne* at past times, and creates a Batch file	infile.txt	Job.Theta (summary results) job.Batch (core of the estimates)
NatSizeDist()	Calculates estimates of effective size (*Ne*) at a number of times from 0 to a certain time ago (given by the user), plots and saves results on files.	job.Batch	job.Nstat job.Ndist job.Nsize
LogSizeDist()	Same as NatSizeDist() in the *Log(Ne)* scale.	job.Batch	job.Lstat job.Ldist job.Lsize
NTdist()	Graphical summary of the posterior distribution of estimates of *Log(Ne)* in the past (a 2D plot).	job.Batch	job.2D
plotNdistrib()	Plots the posterior distributions of the estimates of *Ne* (or *Log(Ne)*) at a number of times in the past.	job.Ndist or job.Ldist	
Tmrca()	Calculates and plots the posterior distribution of the Time to Most Recent Common Ancestor	job.Batch	job.Tmrca

A. DEFINITIONS OF VARIABLES

1) FROM VarEff()

Variable of VarEff()	Definition
parafile	Name that you give to the job and to the output files created by the model.
infile	Name of the data file.
NBLOC	Number of Loci.
JMAX	Number of time when the effective size has changed, used to generate step functions simulating the past demography. Ex: JMAX=2, if we think that the population took 3 different effective sizes in the past.
MODEL	Choice one mutation model in: S = Single Step Model, T = Two Phase Model, G = Geometric Model, and provide an additional coefficient (C) for T and G models.
MUTAT	Mutation rate, assumed the same for all loci.
NBAR	Prior value for the effective size.
VARP1	Variance of the prior log-distribution of effective sizes. Ex: VARP1=3 allows for searches with 20 to 40-fold relative variations of effective size.
RHOCORN	Coefficient of correlation between effective sizes in successive intervals
GBAR	Number of generations since the assumed origin of the population.
VARP2	Variance of the prior log-distribution of time intervals during which the population is assumed of constante size.
DMAXPLUS = DMAX+1	DMAX is the maximal distance between alleles (number of microsatellite motifs) that is used in the estimation algorithm).
Diagonale	A smoothing parameter to balance the observed covariance structure with a theoretical diagonal variance matrix and avoid numerical instability. Diagonale = 0.5 is a robust choice.
NumberBatch	Number of batch (nbatch) for metrop in MCMC.
LengthBatch	Length of batch (blen) for metrop in MCMC.
SpaceBatch	Space of batch (nspac) for metrop in MCMC.
Burnin	Length of the burnin period.
AccRate	Acceptation rate.

2) FROM NatSizeDist(), LogSizeDist(), NTdist() and Tmrca():

Variable of NatSizeDist(), LogSizeDist(), NTdist() and Tmrca()	Definition
TMAX	Length of the period (phase) for which the distributions of Ne or Log(Ne) in the past are generated.
NBT	Number of time intervals. Ex: If TMAX=1000 generations, and NBT=100, estimates are calculated every tenth generation until 1000 generations ago.
TMAXin and LogTMAXin	Maximal values of the period for which the distributions of TMRCA and of Log(TMRCA) are calculated by the function Tmrca().

B. FIRST STEP: THETA FUNCTION

Examples in the following made use of data drawn from the simulation of a sharp bottleneck in a population of size 1000, reduced to size 100 since 200 generations. Estimations made use of 20 microsatellite markers submitted to a (very) high rate of mutation (μ = 0.01).

The Theta() function allows you to get a first view of data, and to prepare a parameter file to run VarEff(). The function can be used in two ways:

1) DIRECTLY

Example with data file “InputTest.txt”:

```
> Theta(parafile="Test",infile='InputTest.txt', NBLOC=20, Burnin=10000,
AccRate=0.25)
[1] Estimation of effective sizes Ne(T) in the past
[1] from microsatellites data
[1]
[1] This function builds a parameter file 'parafile.R'
[1] allowing you to run VarEff with the command: source('parafile.R')
[1]
[1] *** Data ***
[1] Name of the data file= InputTest.txt
[1] Number of microsatellite markers= 20
[1] Name of the job = Test
```

Then, Theta() asks the user to provide values for the other variables of VarEff().

2) ANSWERING THE QUESTIONS

```
> Theta()
[1] Estimation of effective sizes Ne(T) in the past
[1] from microsatellites data
[1]
[1] This function builds a parameter file 'parafile.R'
[1] allowing you to run VarEff with the command: source('parafile.R')
[1]
[1] *** Data ***
 Name of the data file= InputTest.txt
 Number of microsatellite markers= 20
 Give a name for this job = Test
```

Then Theta() reads the data, shows the distribution of allele distance frequencies in the sample (Fig.1), provides global estimates of theta and write them on a .Theta file,

…

```
[1] Theta (4*Ne*u) is estimated using 3 estimators correlated to
[1] Present, Intermediate and Ancestral population size;
[1] results are written to a .Theta file
[1]
[1] Mean values of estimates Theta_0, Theta_1, Theta_2
[1] 6.746623 14.359955 15.753006
[1] Imbalance indices ln(Theta_1/Theta_0) and ln(Theta_2/Theta_0)
[1] 0.7554014 0.8479892
```

…

asks the user to provide values for the other variables of VarEff(), and writes a .R file to run the VarEff() function (Screen 1).

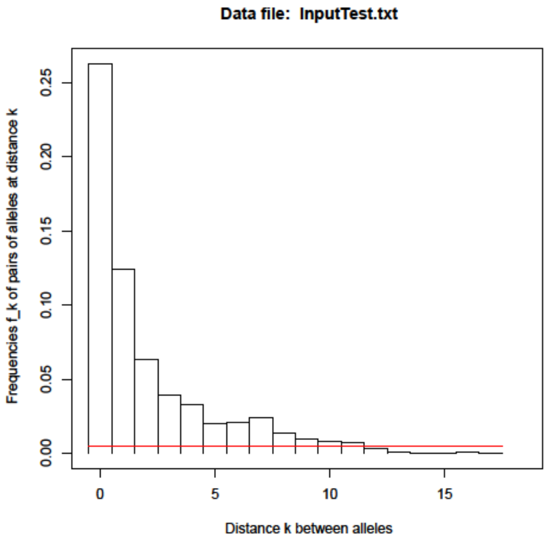


Figure 1. Distribution of the frequencies of allele distances

```
VarEff(parafile = 'Test',
infile = 'InputTest.txt',
NBLOC = 20,
JMAX = 4,
MODEL = ‘S’,
MUTAT = 0.01,
NBAR = 1000,
VARP1 = 3,
RHOCORN = 0,
GBAR = 5000,
VARP2 = 3,
DMAXPLUS = 13,
Diagonale = 0.5,
NumberBatch = 10000,
LengthBatch = 1,
SpaceBatch = 1
Burnin = 10000,
AccRate = 0.25)
```

Screen 1. The Test.R file generated by Theta()

C. SECOND STEP: VAREFF FUNCTION

The VarEff function can be used in two ways:

1) DIRECTLY

Enter the command providing values to all the variables

```
> VarEff(parafile = 'Test', infile = 'InputTest.txt', NBLOC = 20,
JMAX = 4, MODEL = 'S', MUTAT = 0.01, NBAR = 1000, VARP1 = 3,
RHOCORN = 0, GBAR = 5000, VARP2 = 3, Diagonale = 0.5,
NumberBatch = 10000, LengthBatch = 1, SpaceBatch = 1, Burnin=10000, AccRate=0.25,
DMAXPLUS = 13)
```

Or, when Theta() has been previously run, launch with:

```
> source(“Test.R”)
```

Test.R is the file (“name of job”) give during the process of Theta() (see section B.2).

2) ANSWERING THE QUESTIONS

Call the function VarEff(), then answer the questions (red) thanks to the explanation (blue) (Screen 2). In that case, it begins like Theta(), providing the global theta estimates, showing the distribution of allele distances, giving guidelines to choose parameters, and building a .R file containing all the chosen parameters.

At the end of the calculations, VarEff() returns global theta values (as Theta() does) and summaries of adjustment criteria of data to model and the distribution of posterior probabilities, which are added to the .Theta file.

```
[1] Mean values of estimates Theta_0, Theta_1, Theta_2
[1] 6.746623 14.359955 15.753006
[1] Imbalance indices ln(Theta_1/Theta_0) and ln(Theta_2/Theta_0)
[1] 0.7554014 0.8479892
[1] Mean and standard deviation of Quadratic deviations from data
[1] 10.163939 2.622715
[1] Mean and standard deviation of log prior probabilities
[1] -17.995083 1.517846
```

The main result of VarEff() is the .Batch file, which reports a list of demographic evolutions described by step functions. Each line includes:

```
Column 1: the number i of the simulated state (from 1 to Numberbatch)
Column 2: quadratic deviation of data from the i-th simulated state
Column 3: natural logarithm of the prior probability the i-th state
Columns 4 to JMAX+4: the JMAX+1 population size in the i-th state
Columns JMAX+5 to 2 JMAX+4: times of size changes in the i-th state
Columns 2 JMAX + 5: value of the C parameter of the mutation model
```

Results are kept in the Batch files in reduced scales: theta's for population sizes, products of generation numbers times mutation rate for times of size changes. The additional C parameter is a constant set to 0 for the Single Step Mutation Model, positive for geometrical model or negative for the Two Phase Model (TPM).

Screen 2. Questions and Answers when launching VarEff():

```
> VarEff()
[1] Estimation of effective sizes Ne(T) in the past
[1] from microsatellites data
[1] Note that this R code makes use of the mcmc library
[1]
[1] *** Data ***
 Name of the data file= InputTest.txt
 Number of microsatellite markers= 20
 Give a name for this job = Test
[1] *** Demographic model ***
[1] Choose the number of transitions between periods with constant Ne
 Value of JMAX = 4
[1] *** Mutation model ***
[1] Fix the mutation model (S, T or G)
[1] (S = Single Step Model, T = Two Phase Model, G = Geometric Model)
[1] and the prior value of its C parameter, 0 < C < 1
[1] e.g. answer ' T 0.15 ' for the Two Phase Model with a proportion 15% of two
motifs steps
 Mutation model and C coefficient? S
[1] Give the mutation rate MUTAT
 MUTAT = 0.01
[1] *** Priors about population sizes ***
[1]
[1] Prior expected value of Effective Population Size Ne:
 Prior Ne = 1000
 Variance of log(Ne)= 3
[1] Prior correlation between successive population sizes:
 correlation= 0
[1] *** Priors about time and time intervals ***
[1]
```

```
[1] Prior time since the Origin (number of generations))
 Origin time (generations) = 1000
[1] Prior about time intervals:
 Variance of log(time interval)= 3
[1] *** Smoothing the Covariance matrix ***
[1]
[1] Calculation of an approximate likelihood assumes the theoretical
[1] variance-covariance matrix V of observations is known.
[1] It is in fact replaced by a combination of the sample estimate Vs
[1] and of the diagonal matrix Vd made of theoretical variances:
[1] V = (1-D) * Vs + D * Vd
[1] D must be > 0 and less than 1
[1] D = 1 means that only theoretical variances are considered
[1] and correlations are ignored
[1] D = 0 means that only observed data are considered, a choice
[1] that may raise numerical instability
[1] For more detail read the help and article
[1] A medium choice, D = 0.5 , is generally robust
 D = 0.5
[1] *** Length of the MCMC chain (look at the metrop specifications) ***
[1] Suggested initial values:
[1] nbatch = 10000, blen=10, nspac=10 , Burnin=10000, AccRate=0.25
[1] Enter the five values you wish to use
 nbatch = 10000
 blen = 1
 nspac = 1
 Burnin = 10000
 AccRate = 0.25
Read 240 items
[1] Effective Number of markers = 20
[1] Plot of the (mean) distribution of distances between alleles
[1] Choose the range 0:DMAX of allele distances to be analysed
[1] e.g. so that f_k < 0.005 for k > DMAX (under the red line)
 Enter DMAX = 12
[1] Mean values of estimates Theta_0, Theta_1, Theta_2
[1] 6.746623 14.359955 15.753006
[1] Imbalance indices ln(Theta_1/Theta_0) and ln(Theta_2/Theta_0)
[1] 0.75540135 0.84798912
[1] Mean and standard deviation of Quadratic deviations from data
[1] 9.979541 2.662895
[1] Mean and standard deviation of log prior probabilities
[1] -18.563440 1.760705
```

D. THIRD STEP: OUTPUTS FROM RESULTS FILES

To obtain the distribution of **(1)** effective size (*Ne*) or **(2)** *Log*(*Ne*) at a number of generations in the past, run the function **(1)** NatSizeDist() or **(2)** LogSizeDist()

Both functions work the same. They use the job.Batch file created by VarEff(). Results can be shown in natural census values of population sizes, or in reduced theta scale if mutation rate is set to 0 in these functions.

1) DIRECTLY

Example:

NatSizeDist(NameBATCH= Test.Batch, MUTAT=0.01, TMAX=1000, NBT=20)

In this example the estimates of *Ne* will be calculated every 50-th generation from 0 to 1000 generations ago.

2) ANSWERING THE QUESTIONS

Call the function NatSizeDist(), then answer the questions (same answers, using the reduced scales):

```
> NatSizeDist()
[1] VarEff - View past effective sizes Ne(T)
[1] and save posterior distributions
 Name of the batch file= Test.Batch
[1] Mutation rate: Enter 0 to use reduced scales Theta's and G*µ
 mutation rate = 0
[1] Enter the length of the period to be analysed
[1] give time in the reduced time scale G*µ
 Length of the period = 10
 Number of time intervals= 20
[1] Results are expressed in reduced scales Theta's and G*µ
[1] Plot of mean results
```

Results include a plot (Fig.2) :

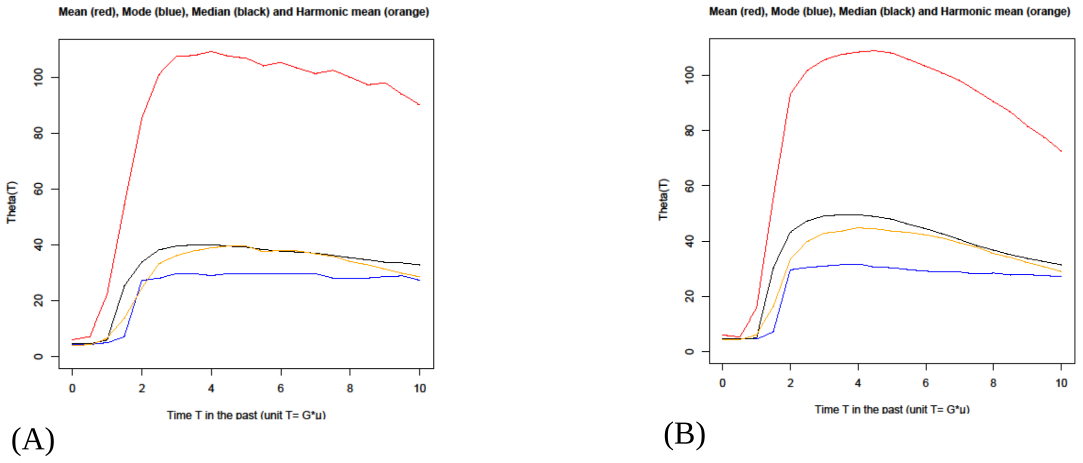


Figure 2. Estimates of *Ne* in the past (reduced scale).
(A) using a short MCMC chain as in Table 1; (B) using a long chain (10000, 10, 10).

A file “job.Nstat (or job.Lstat)” gives statistics at the different times:

1. Generations (if MUTAT is not 0) or the corresponding reduced time
2. Arithmetic means of *Ne* or *Log(Ne)*
3. Harmonic means of *Ne* (not provided for *Log(Ne)*)
4. Mode
5. Median
6. Quantile 5%
7. Quantile 95%
8. Standard deviation of *Ne* of *Log(Ne)*

Two others files are produced:

- job.Ndist (or job.Ldist) containing the detailed densities of the posterior distributions of *Ne* (or Log(*Ne*)), at the NBT+1 considered times. Such files are used by the plotNdistrib() function (see below).
- job.Lsize (or job.Lsize) gives, for each of the nbatch simulated states, the NBT+1 values of Ne at the considered times in the period.

E. THIRD STEP: PLOTTING POSTERIOR DENSITIES

1) TWO DIMENSIONAL SUMMARY: function NTdist():

The function NTdist() uses the job.Batch file and makes a figure to summarize the posterior distribution of *Ne* or Log(*Ne*) in a certain period (TMAX) given by the user.
The graph is plotted and saved as a text.

Example:

Directly using the function as:

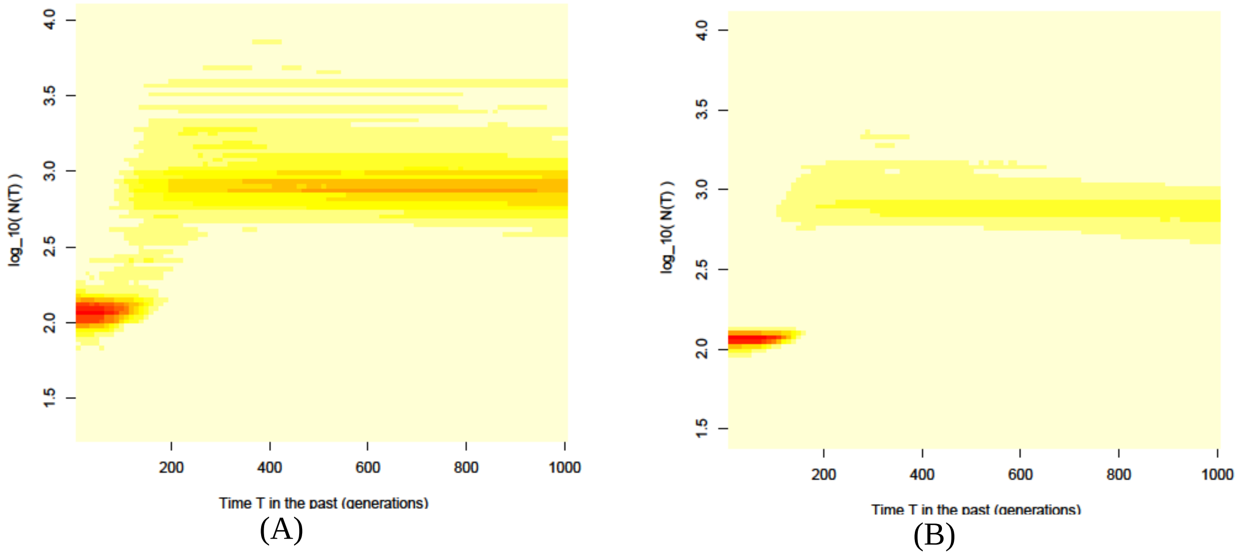
```
> NTdist(NameBATCH= Test.Batch, , MUTAT=0.01, TMAX=1000)
```

In this example the summary will be from 0 to 1000 generations ago.

By answering the questions (example giving a “true” mutation rate):

```
> NTdist()
[1] VarEff - View joint distribution of T and Ne(T)
[1] and save this joint posterior distributions
 Name of the batch table= Test.Batch
[1] Mutation rate: Enter 0 to use reduced scales Theta's and G*µ
 mutation rate = 0.01
[1] Enter the length of the period to be analysed
[1] give time in number of generations
 Length of the period = 1000
[1] Assumed mutation rate= 0.01
[1] Plotting the 2D ( g ,log_10( Ne(g) ) ) distribution
[1] Time range: from 0 to 1000 (generations)
[1] Minimum and Maximum values of log_10( Ne ) in the plot:
[1] Minimum = 1.20233626012957
[1] Maximum = 4.08949542800206
[1] Coverage = 97.2457 %
```

The result is shown as a plot (Fig. 3):



(A) (B)

Figure 3. Global plot of the posterior distribution of *Ne* in the past
(A) using a short MCMC chain as in Table 1; (B) using a long chain (10000, 10, 10).

2) PLOTTING DENSITIES OF POSTERIOR DISTRIBUTIONS

The function plotNdistrib() makes use of files .Ndist or .Ldist previously built by NatSizeDist() or LogSizeDist(). It allows the user to exhibit the density of the posterior distribution of *Ne* (or of *Log(Ne)*) at several times in the past.

Compared to NTdist() that gives a rough but global view of these densities, plotNdistrib() gives a precise view of these densities at a small number of times. The global 2D plot given by NTdis() may help choosing the times when plotNdistrib() is used.

The function is interactive, allowing the user to check several plots. Plots can be saved by the the user (Figure 4)

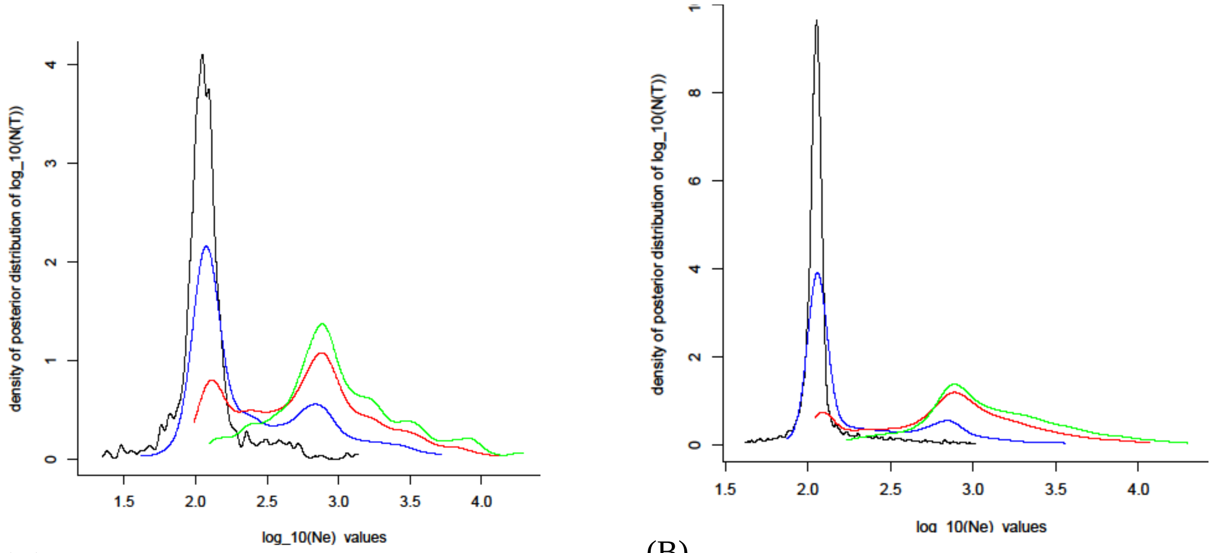
Call the function as plotNdistrib(“job.Ndist”), or as plotNdistrib() and answer the questions:

NOTE: the example refers to the same previous example, using extractions from LogSizeDist(), to provide densities of Log(Theta).

In Fig. 4-A, posterior densities are quite erratic, because the MCMC chain used in the example was very short: MCMC parameters were (10000,1,1) whereas getting good results generally requires 100 times longer runs, with parameters set at (10000,10,10) for example, as in Fig. 4-B.

```
> plotNdistrib()
[1] *** Plot posterior distributions of N(T) ***
 Name of the 'parafile.Ndist' file = test.Ldist
[1] 1 0
[1] 2 50
[1] 3 100
```

```
[1] 4 150
[1] 5 200
[1] 6 250
[1] 7 300
[1] 8 350
[1] 9 400
[1] 10 450
[1] 11 500
[1] 12 550
[1] 13 600
[1] 14 650
[1] 15 700
[1] 16 750
[1] 17 800
[1] 18 850
[1] 19 900
[1] 20 950
[1] 21 1000
[1] Choose the number of times (suggestion: <=5) you wish to plot the
distribution
 number of times (stop on 0): 4
 case # 1 , instant number = : 1
 case # 2 , instant number = : 3
 case # 3 , instant number = : 4
 case # 4 , instant number = : 5
[1] Choosing the range of Ne values
[1] note that a log_10 scale of sizes is used here
[1] maximal Ne value is 4.287239. Give an upper value :
 enter maximum Ne value= 4.5
[1] minimal Ne value is 1.351914. Give a lower value :
 enter minimum Ne value= 1.3
[1] Choose the number of times (suggestion: <=5) you wish to plot the
distribution
 number of times (stop on 0): 0
```



(A)

(B)

Figure 4. Posterior distributions of *Log(Ne)*, at several times in the past.
Black: T= 0 (current time), Blue: T=50 ; Red: T=150 ; Green: T=200.
(A) using a short MCMC chain as in Table 1; (B) using a long chain (10000, 10, 10).

F. AN ADDITIONAL LOOK: PLOTTING THE DISTRIBUTION OF TMRCA

The function Tmrca() derives the posterior distribution of the Time to the Most Recent Common Ancestor of two alleles drawn from the current population. The function uses the job.Batch file and makes a figure to summarize the posterior distribution of *TMRCA* or Log(*TMRCA*) in a certain period (TMAX) given by the user. Working on Log distribution allows to work on larger time scale.

Like plotNdistrib(), it is an interactive function that allows the user to check several options.

Graphs are plotted and can be saved on files by the user (Figure 5).

The job.Tmrca file records the trials with the probabilities that TMRCA is less than the chosen upper times.

TMRCA can be equal at = $g*\mu$ (generation time * mutation rate) or only g. It depends on whether the mutation rate is provided or no.
By providing the mutation rate, TMRCA = g.
By not providing the mutation rate, TMRCA = $g*\mu$.

Example, answering the questions:

```
> Tmrca()
[1] VarEff suite - function Tmrca()
[1] Retrieving the posterior distribution of TMRCA
 Give the name of a .Batch file Test.Batch
 Enter the mutation rate 0.01
[1] Mean, St.dev and Max of T_JMAX
[1] 2175 3116 35958
[1] Most ancient proposed time = 8407
[1] Plotting the posterior density of T_MRCA
[1] Enter upper limit for ancient times (generations ago)
 (type 0 to turn to Log(TMRCA) densities) TMAX = 0
```

Answering “0” skips calculations about natural TMRCA, to turn to the distribution of Log(TMRCA):

```
 [1] Plotting the posterior density of log(T_MRCA)
[1] Enter upper limit for ancient times (generations ago)
 (stop on 0) TMAX = 200
[1] Prob( TMRCA < TMAX ) = (approx) 0.435
[1] Plotting the posterior density of log(T_MRCA)
[1] Enter upper limit for ancient times (generations ago)
 (stop on 0) TMAX = 2000
[1] Prob( TMRCA < TMAX ) = (approx) 0.869
[1] Plotting the posterior density of log(T_MRCA)
[1] Enter upper limit for ancient times (generations ago)
 (stop on 0) TMAX = 20000
[1] Prob( TMRCA < TMAX ) = (approx) 1
```

```
[1] Plotting the posterior density of log(T_MRCA)
[1] Enter upper limit for ancient times (generations ago)
 (stop on 0) TMAX = 0
```

<u>The function can also be called providing its arguments, for example:</u>

```
> Tmrca(batchfile=”Test.Batch”, MUTAT=0.01, TMAXin=0, LogTMAXin=200)
```

In that case, calculations are restricted to the specified values of the TMAX parameters (in this example, density of natural times are skipped and the plot includes only the right part of Figure 5).

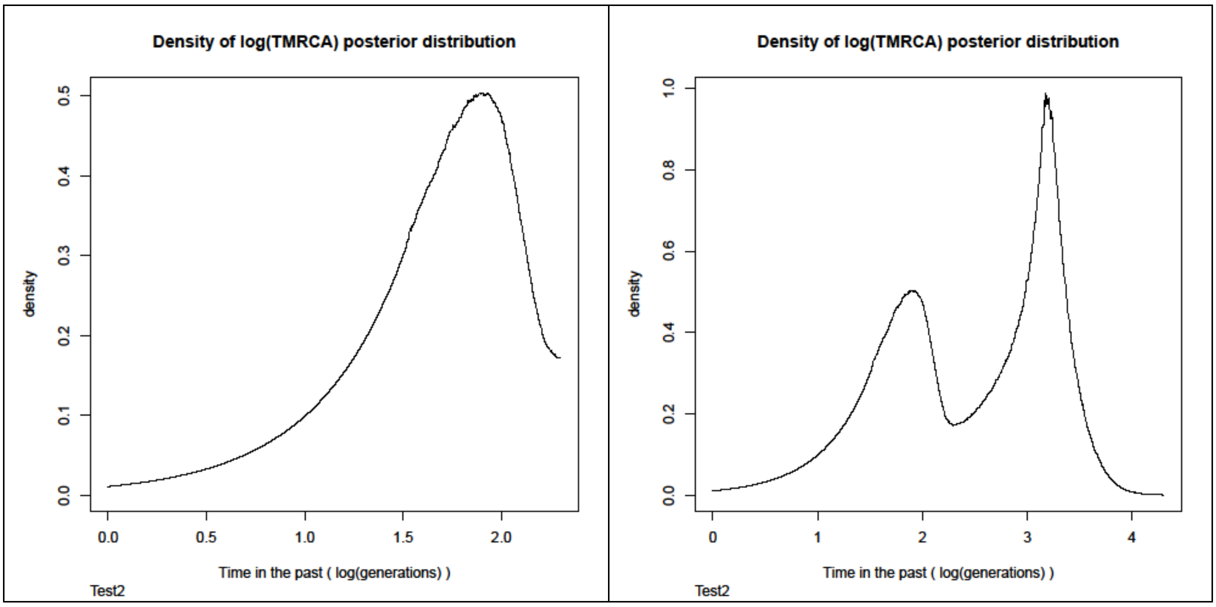


Figure 5. Distribution of the TMRCA of two alleles drawn at present time.

Figure 5 shows the plots obtained from the first and third options.

- For the first option TMAX = 200, as we gave the mutation rate, the x-axis = Log10(g) and the maximum of the x-axis is Log10(200) = 2.3.
- For the first option TMAX = 20000, as we gave the mutation rate, the x-axis = Log10(g) and the maximum of the x-axis is Log10(200) = 4.3.

The first chosen time (TMAX = 200) corresponds to the time in which the population underwent its severe bottleneck. The Figure shows that coalescent events occurred mainly in two periods: a recent period when the effective population size was 100, and an ancient period when the effective population size was 1000 before the bottleneck. It also indicates that about half of coalescent events occurred in the recent period.

<u>To estimate the time of a bottleneck in number of generation (g):</u>

- By providing the mutation rate (example here 0.01) and turning the distribution in Log (see our example above), the x-axis is equal to Log10(*g*). Figure 5 shows a first peak (bottleneck) at 1.8, so Log10(*g*) = 1.8. In generation time, it is equal to g = 10^1.85 = 63 generations ago.
- By not providing the mutation rate (not the case in our example) and turning the distribution in Log, the x-axis is equal to Log10(*g*μ*). In this case, g = (10^x)/μ, with x equal to the x-axis value of the peak.

G. UTILITIES

Additional scripts are provided on the web site. The script is called “VarEffMSVAR”. Two functions change data file formats between VarEff and MSVAR software, that use the same kind of data and the basic calculation of allele distance frequencies that are expected after population size variations.

msvar2vareff() allows an MSVAR input file to be transformed into a VarEff input data file.

vareff2msvar() allows a VarEff input data file to be converted to an MSVAR data infile.

H. RECOMMENDATIONS

Preliminary test:

Start with short batches (1000 * 1 * 10) and a DMAX not too large, and make different trials

1. JMAX (example: JMAX=2, JMAX=3, JMAX=4)
2. NBAR (start with the value between Theta0-Theta2 and a medium VARP1. Check stability for larger VARP1. Choose NBAR such that the results revealed narrow quantiles (5%-95%, columns 6 and 7 in .NSTAT file), but keep the pair NBAR, MUTAT in accordance with the global Theta values.
3. GBAR (start with a large value and a large VARP2, then increase or decrease GBAR to check the best values, then use smaller VARP2 values)
4. Increase DMAX to the maximum, to check robustness (note that calculation time increases with DMAX, and large DMAX may lead to numerical problems)

Final result with:

- Batch with (NumberBatch = 10000, LengthBatch = 10, SpaceBatch = 10)
- RHOCORN is often set to 0
- Diagonale is often set to 0.5

I. ANNEXE - TEST FILE “InputTest.txt”

```
9
4 12 2 0 14 21 15 3 9
12
3 30 10 17 15 0 1 0 0 0 0 4
18
2 0 0 0 0 0 0 0 0 0 0 1 29 16 1 3 24 4
5
8 3 43 16 10
9
13 17 17 1 0 1 16 9 6
5
1 3 16 58 2
5
9 19 46 5 1
11
1 8 3 7 35 13 0 0 0 12 1
17
1 0 0 0 0 0 0 0 0 1 15 5 53 0 2 0 3
9
13 38 10 0 3 5 0 6 5
```

```
14
1  21  0  0  1  24  2  0  0  0  0  7  21  3
11
7  1  0  0  0  1  3  12  11  34  11
13
7  33  9  1  0  0  0  5  4  1  12  5  3
11
2  9  0  0  6  0  0  18  32  11  2
10
18  8  1  8  6  1  17  20  0  1
10
8  17  8  5  2  3  4  2  21  10
15
5  11  19  9  3  0  0  0  0  0  9  8  2  9  5
14
1  2  6  0  0  0  2  11  5  25  13  1  13  1
8
3  35  26  6  7  0  1  2
14
1  4  0  0  0  38  18  7  9  1  0  0  1  1
```