Epos2plot 1.2 : Plot epos Results

Bernhard Haubold
Max-Planck-Institute for Evolutionary Biology, Plön, Germany

2019-08-27

1 Introduction

The package epos2plot contains two programs: epos2plot itself and plotSum. Epos2plot summarizes multiple epos results into quantile plots. Figure 1 shows such a quantile plot computed from 1000 epos (Lynch et al., 2019) results, which in turn were computed from 1000 haplotype samples simulated from the simplest population model, constant size. PlotSum summarizes multiple data epos2plot data sets by computing their mean and standard deviation or standard error.

In the following sections I first explain how to set up epos2plot and then give a tutorial-style introduction to the usage of epos2plot and plotSum

2 Getting Started

Epos2plot is written in Go and distributed via github.

- Obtain the program

  ```
  git clone https://github.com/evolbioinf/epos2plot
  ```

- Change into the new directory

  ```
  cd epos2plot
  ```

- Make

  ```
  make
  ```

- Test

  ```
  make test
  ```

- Install

  ```
  make install
  ```

- The documentation is typeset in LaTeX. Make the documentation

  ```
  make doc
  ```

  The manual is now in doc/epos2plot.pdf.

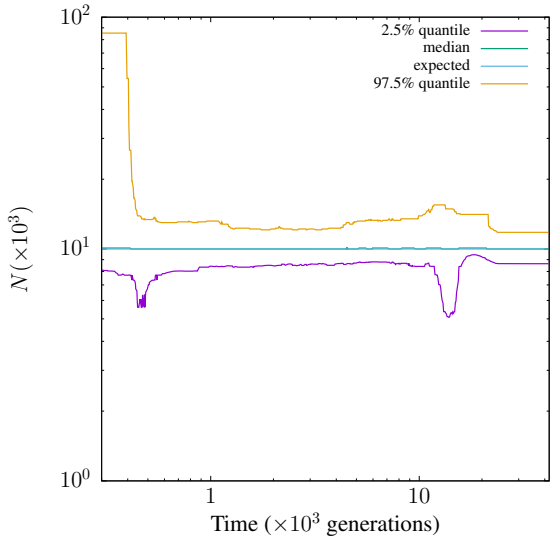

Figure 1: Population size estimation using `epos` on 1000 samples of size 30 drawn from a population of 10,000, followed by `epos2plot`.

3 Tutorial

3.1 `epos2plot`

The example data displayed in Figure 1 is based on a scenario taken from (Liu and Fu, 2015, Figure 2a): Assume a population of constant size 10,000 individuals and draw 1000 samples of 30 haplotypes, each consisting of 10 Mbp. Such data can be simulated using

```
mspms 30 1000 -t 4800 -r 3800 1e7 |
ms2sfs                            |
epos -l 1e7 -U -u 1.2e-8 > example.epos
```

The output of the fast coalescent simulator `mspms` (Kelleher et al., 2016) is converted to site frequency spectra by `ms2sfs`[1]. `Epos`[2], finally, estimates population sizes from these spectra.

- Get the precomputed example data, uncompress it, and move it into the `data` directory

  ```
  wget guanine.evolbio.mpg.de/epos2plot/example.epos.bz
  bunzip2 example.epos.bz2
  mv example.epos data/
  ```

- Look at the first sample, which happens to occupy 11 lines in the uncompressed data file:

  ```
  head -n 11 data/example.epos
  #InputFile: stdin
  #Polymorphic sites surveyed:      18966
  #Monomorphic sites surveyed:    9981034
  #m = 1; maximum Log(Likelihood):    -159.489374 {2}
  #m = 2; maximum Log(Likelihood):    -156.442818 {2,3}
  #m = 3; maximum Log(Likelihood):    -156.330973 {2,3,11}
  #Final Log(Likelihood):             -156.442818
  ```

[1] `https://github.com/evolbioinf/sfs/`
[2] `https://github.com/evolbioinf/epos/`

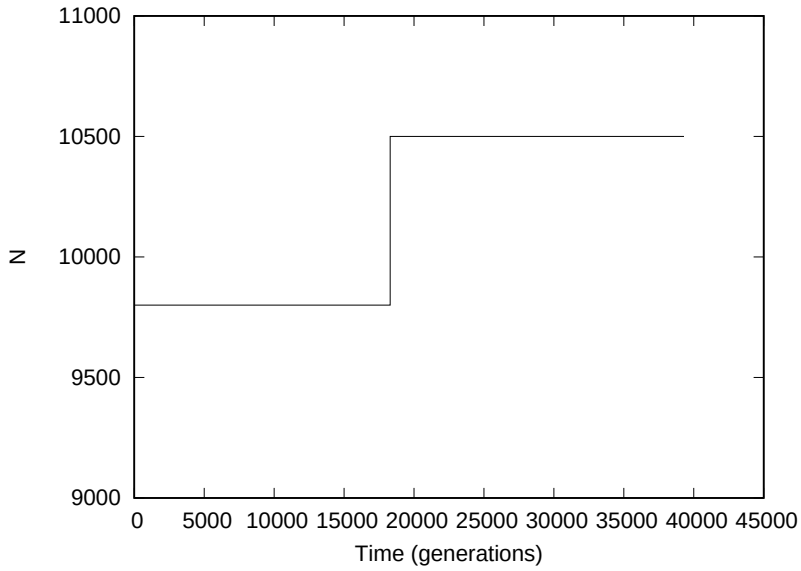

Figure 2: Plot of single demography.

```
#d^2: 0.00352383
#Level T[Level] N[Level]
3 1.83e+04 9.80e+03
2 3.93e+04 1.05e+04
```

The part that concerns us here are the two lines without leading hashes at the bottom. We read them from the bottom up, which means going from the past toward the present. "Level" 2, the root of the coalescent, is located 39,400 generations in the past, at which point the population size, N, was 10,500 individuals. This size stayed constant until generation 18,500 in the past, when $N = 9900$, which remained unchanged until the present.

- Instead of reading the data upside down, as we have just done, it is easier to extract it automatically into two columns, time and size

```
head -n 11 data/example.epos | epos2plot -r
0 9800
18300 9800
18300 10500
39300 10500
```

- and plot it using `pipePlot`[3]

```
head -n 11 data/example.epos |
epos2plot -r                 |
pipePlot -x "Time (generations)" -y N -X 0:45000 -Y 9000:11000
```

to get Figure 2.

- Plot all 1000 demographies in the example data set

```
epos2plot -r data/example.epos |
pipePlot -x "Time (generations)" -y N
```

[3] http://github.com/evolbioinf/pipeplot

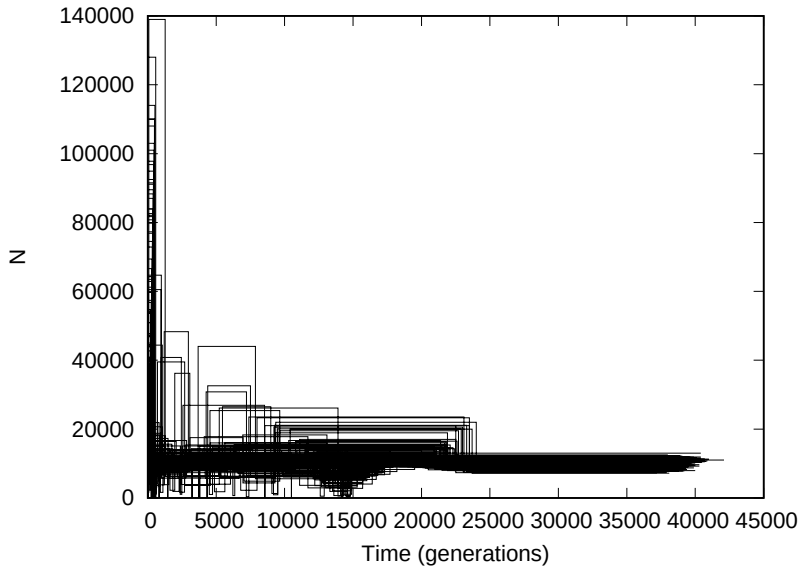

Figure 3: Plot of all demographies in the example data.

to get Figure 3.

- Notice the ragged right hand side of Figure 3 due to samples coalescing at different points in time. This leads to a fundamental problem with our analysis: The expected time for a sample of n haplotypes to reach its most recent common ancestor, $E[T_{\mathrm{MRCA}}]$, is proportional to the population size (Wakeley, 2009, p. 76):

$$E[T_{\mathrm{MRCA}}] = 4N\left(1 - \frac{1}{n}\right).$$

  As we move from the present into the past, samples successively find their most recent common ancestor, and might be expected to drop out of the quantile computation. However, this would lead to a strong upward bias in the results, as only samples that induce large population size estimates endure into the more distant past. To avoid this bias, `epos` by default, i. e. without `-r`, extends the population size measured at the most recent common ancestor of each sample into the past until the last sample has coalesced.

- Instead of plotting raw demographies, `epos2plot` by default summarizes them by computing 2.5% and 97.5% quantiles around the median:

```
epos2plot data/example.epos | head
#Time   LowerQ Median UpperQ
0       1       10000   76900
0.0046  1       10000   76900
0.0046  1       10000   76900
0.00952 1       10000   76900
0.00952 544     10100   76900
0.0148  544     10100   76900
0.0148  1000    10100   76900
0.0205  1000    10100   76900
0.0205  1090    10100   76900
```

  where `Time` contains the time in generations, `LowerQ` the lower quantile of the population size, `Median` its median, and `UpperQ` its upper quantile.

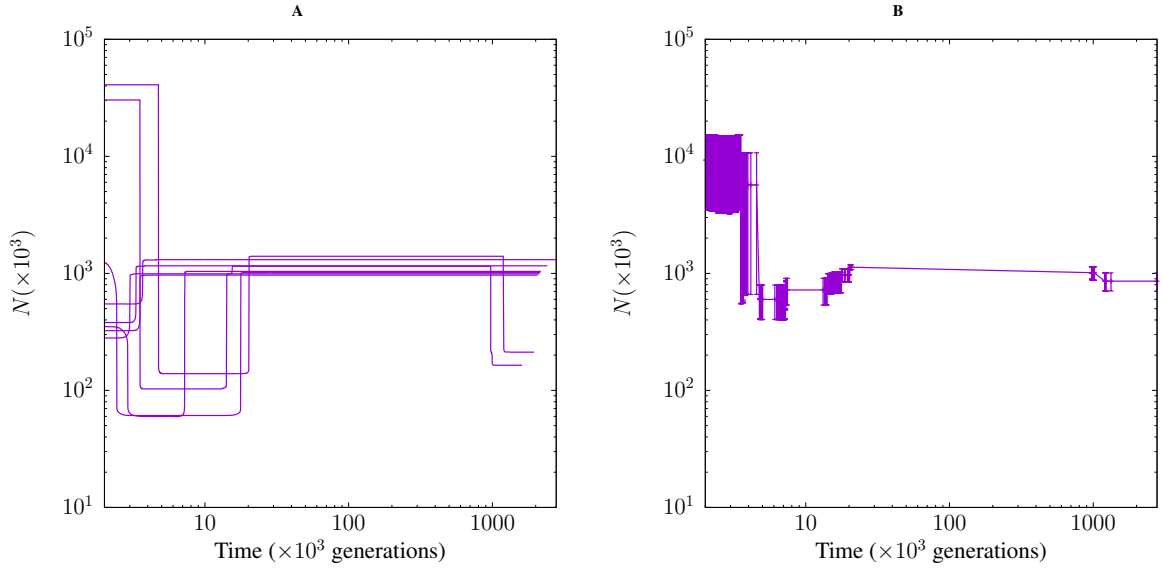

Figure 4: Medians of eight bootstrapped runs of `epos2plot` on the Chq population of *D. pulex* (**A**), mean±SEM of that plot.

- In case you are wondering about time points like 0.0046 generations, they result from very early coalescent events and make little biological sense. `Epos2plot` allows the user to choose a minimum step length between the data points it prints using the `-t` option. By default this is 0, but if we now set it to 1, we get

```
epos2plot -t 1 data/example.epos | head
#Time    LowerQ Median UpperQ
0        1       10000   76900
1.4      1140    10100   76900
1.4      1260    10100   76900
2.4      1260    10100   76900
2.4      1280    10100   76900
4.03     1280    10100   76900
4.03     1340    10100   76900
5.18     1340    10100   76900
5.18     1370    10100   76900
```

  The `-t` option is particularly useful with very large samples of `epos` results, where the number of data points close to the zero line can be so large as to bog down the `epos2plot` run. In any case, the plot of these values is Figure 1, which we already looked at in the Introduction. It illustrates the excellent fit between the predicted and the expected population size.

3.2 `plotSum`

Often several samples are taken from a given population. For example, Figure 4A shows the medians of eight bootstrapted samples drawn from the Chq population of *Daphnia pulex*. Figure 4B summarizes these curves as mean±SEM.

References

J. Kelleher, A. Etheridge, and G. McVean. Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS Comput. Biol.*, 12:1–2, 2016.

X. Liu and Y.-X. Fu. Exploring population size changes using SNP frequency spectra. *Nature Genetics*, 47: 555–562, 2015.

M. Lynch, B. Haubold, P. Pfaffelhuber, and T. Maruki. Inference of historical population-size changes with allele-frequency data. *In prep.*, 2019.

J. Wakeley. *Coalescent Theory: An Introduction*. Roberts & Company, Colorado, 2009.