epos: Estimating Population Size from the Site Frequency Spectrum

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

June 13, 2018

1 Introduction

Epos estimates historical population sizes based on a site frequency spectrum. The program implements theory developed by Michael Lynch (Arizona State University) and Peter Pfaffelhuber (Freiburg University), which will be described in a forthcoming paper.

2 Prerequisites

Epos depends on two libraries, the Gnu Scientific Library (lgsl), and the Basic Linear Algebra Subprograms (lblas).

3 Getting Started

epos was written in C on a computer running Linux and should work on any standard UNIX system. However, please contact me at haubold@evolbio.mpg.de if you have any problems with the program.

- Change into the epos directory

  ```
  cd epos
  ```

  and list its contents

  ```
  ls
  ```

- Generate epos

  ```
  make
  ```

- List its options

  ```
  ./epos -h
  ```

- Test program

  ```
  ./epos data/kap144i.dat
  ```

4 Listing

The following listing documents the driver program for `epos`.

```
/***** epos.c ************************************
 * Description:
 * Author: Bernhard Haubold, haubold@evolbio.mpg.de
 * Date: Mon Oct 23 10:10:47 2017
 ***********************************************/
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <gsl/gsl_errno.h>
#include "interface.h"
#include "eprintf.h"
#include "sfs.h"
#include "popSizes.h"
#include "util.h"
#include "xval.h"

void scanFile(FILE *fp, Args *args, char *fileName){
  Sfs *sfs, *bSfs;
  PopSizes *ps;
  int i;
  gsl_rng *rand;

  ps = NULL;
  sfs = getSfs(fp, args);
  printSfsStats(sfs);
  rand =  ini_gsl_rng(args);
  xvalM(sfs, args, rand);
  ps = getPopSizes(sfs, args);
  if(negPopSizes(ps)){
    args->L = 1;
    xvalM(sfs, args, rand);
    ps = getPopSizes(sfs, args);
    if(negPopSizes(ps))
      fprintf(stderr, "WARNING:␣Negative␣population␣sizes\n");
  }
  printTimes(ps, sfs);
  bSfs = NULL;
  for(i=0; i<args->b; i++){
    printf("Entered␣bootstrap\n");
    freePopSizes(ps);
    freeSfs(bSfs);
    bSfs = bootstrapSfs(sfs, rand, args);
    printSfsStats(bSfs);
    ps = getPopSizes(bSfs, args);
    printf("#InputFile:\tbootstrapped_%s\n", fileName);
    printTimes(ps, bSfs);
  }
  freeSfs(sfs);
  freeSfs(bSfs);
  freePopSizes(ps);
  free_gsl_rng(rand, args);
```

```
}

int main(int argc, char *argv[]){
  int i;
  char *version;
  Args *args;
  FILE *fp;

  version = "0.56";
  setprogname2("epos");
  args = getArgs(argc, argv);
  if(args->v)
    printSplash(version);
  if(args->h || args->e)
    printUsage(version);

  if(args->t)
    omp_set_num_threads(args->t);

  if(args->numInputFiles == 0){
    fp = stdin;
    scanFile(fp, args, "stdin");
  }else{
    for(i=0;i<args->numInputFiles;i++){
      printf("#InputFile:\t%s\n", args->inputFiles[i]);
      fp = efopen(args->inputFiles[i],"r");
      scanFile(fp, args, args->inputFiles[i]);
      fclose(fp);
    }
  }
  free(args);
  free(progname());

  return 0;
}
```

5 Change Log

- Version 0.1 (Oct. 25, 2017)
  - First running version based on the GSL.
- Version 0.2 (Oct 26, 2017)
  - Used `LAPACKE_dgesv` in `getPopSizes`; makes no difference compared to the previous version.
- Version 0.3 (Oct 26, 2017)
  - Used `LAPACKE_dgesvx` in `getPopSizes`; makes no difference compared to previous version.
- Version 0.4 (Oct 27, 2017)
  - Allow construction of SFS based on `-t` switch.
  - Allow switching between the GSL algorithm (`-g`) and the default LAPACK algorithm.
- Version 0.5 (November 2, 2017)

  - Allow the straight usage of the trapezoid matrix for solving the system (`-T`).
  - Print out coefficient matrix (`-p`).
- Version 0.6 (November 10, 2017)
  - Use `long double` in `getPopSizesTri2`; did not help.
- Version 0.7 (November 10, 2017)
  - Implement Peter Pfaffelhuber's formula; working only partially.
- Version 0.8 (November 10, 2017)
  - Peter's equation (6) is working.
- Version 0.9 (November 10, 2017)
  - Peter's equation in arbitrary precision in MPFR library. Numerical stability achieved with 329 bits per number.
- Version 0.10 (November 30, 2017)
  - Implemented equation (3) from Peter's memo dated Nov. 13. Not working.
- Version 0.11 (December 1, 2017)
  - Implemented revised equation (3) from Peter's memo dated Nov. 30. Not working.
- Version 0.12 (December 15, 2017)
  - Implemented Estimator 2.1 from Peter's memo dated Dec. 2, 2017. Code is running, but the results look odd.
- Version 0.13 (December 16, 2017)
  - Fixed the implementation of Estimator 2.1; results OK now.
- Version 0.14 (December 18, 2017)
  - Implemented optimization strategy for folded SFS; working.
  - Implemented optimization strategy for unfolded/even SFS; working.
  - Implemented optimization strategy for folded/odd SFS; not working yet.
- Version 0.15 (December 18, 2017)
  - Fixed error in search for optimal number of steps.
- Version 0.16 (December 19, 2017)
  - Fixed error in left-hand side of folded/odd equation; working.
  - Changed search strategy.
  - Changed default value of `-d` from 10^{-6} to 10^{-3}.
  - Simplified user interface.
  - Included set of test cases (`test.sh`).
- Version 0.17 (December 20, 2017)
  - Fixed searching routine.
  - Changed default value of `-d` from 10^{-3} to 10^{-2}.
  - Included addition of the λ-factor; seems to make no difference.

- Version 0.18 (December 20, 2017)
  - Fixed the λ-factor; computation is now much stabilized.
- Version 0.19 (January 11, 2018)
  - Included the λ-factor in the computation of Ψ. Computations now applicable to real data.
  - Changed the default-value of λ from 10^{-7} to 10^{-5}.
- Version 0.20 (January 11, 2018)
  - Consider zero-class mutations in computation, if present.
  - Changed default λ to 2×10^{-5} to get all data sets to run.
- Version 0.21 (January 13, 2018)
  - Reverted output of levels, going from the present into the past.
  - Default output is now as a function of times instead as a function of levels.
  - Included "step-wise" option for plotting times and levels.
  - Fixed time computation.
  - Included error message for negative population sizes.
  - Removed memory leaks and other subtle bugs using `valgrind`.
- Version 0.22 (January 16, 2018)
  - Fixed important bug in function foldedEpsi, where variable `b` was computed as a function of `sfs->f[n/2]` instead of, now `sfs->f[n/2-1]`.
- Version 0.23 (January 16, 2018)
  - Search for optimal λ.
  - Allow arbitrary level as first entry in level list.
  - Output λ, Ψ, and the levels added to make it easier to follow the program.
  - Reduced program to folded/even case.
- Version 0.23 (January 17, 2018)
  - Catch GSL-exceptions.
  - Changed output format
- Version 0.24 (January 18, 2018)
  - Not quite sure what changed.
- Version 0.25 (January 18, 2018)
  - Fixed bug in computation of the λ-term in `foldedEpsi`.
- Version 0.26 (January 19, 2018)
  - Set $\lambda = 0$ and add levels until negative population sizes appear. This is fast and appears to be effective.
- Version 0.27 (January 24, 2018)
  - Output number of polymorphic and monomorphic sites surveyed.
- Version 0.28 (???)
- Version 0.29 (January 28, 2018)

  - Fixed missing resetting of times during iteration over files.
  - Removed superfluous option for step-wise output (`-s`).
  - Output name of input file.
  - Removed inclusion of `mpfr.h` from `epos.c`.
  - Removed search for the initial level to add; by definition this must be 2, i. e. one population size for the entire coalescent.

- Version 0.30 (January 31, 2018)
  - Fixed computation of the mutation rate. Previously I multiplied the per site mutation rate with the number of monomorphic positions. Now it is mutated by the number of all positions.
  - Added bootstrapping.

- Version 0.31 (February 1, 2018)
  - Computation of mutation rate was correct in previous version, after all, so reverted to that.

- Version 0.32 (February 7, 2018)
  - Introduced the `-m` switch.

- Version 0.33 (February 7, 2018)
  - Fixed δ computation in function `delta` in `util.c`. This reduces the computation for $m = 1$ to Watterson's estimator, as expected.

- Version 0.34 (February 9, 2018)
  - Reintroduced unfolded spectrum (`-u`) and compared to the equations in Peter's memo of December 19, 2017.
  - Reintroduced λ and set it by default to 10^{-7}.
  - Reintroduced δ and set it by default to 0.0.
  - Fixed sample size computation at the end of `sfs.c`.

- Version 0.35 (February 9, 2018)
  - Introduced estimation of $\lambda = 1./\mu/$`args->f`.

- Version 0.36 (February 14, 2018)
  - Changed setting of λ to $\lambda = \mu \times$ `args->f`. By default `args->f` $= 1$.

- Version 0.37 (February 16, 2018)
  - Fixed sample size computation for folded/even in function `getSfs` in `sfs.c`.

- Version 0.38 (February 16, 2018)
  - Included working `-m` switch.

- Version 0.39 (February 19, 2018)
  - Fixed passing of λ in iterated runs.
  - Fixed numerical underflow when multiplying with λ in `foldedEpsi` in `foldedE.c`.
  - Fixed numerical underflow when multiplying with λ in `psi` in `unfolded.c`
  - Included check for positive Ψ in both cases.
  - Expanded verbose output.

- Version 0.40 (February 21, 2018)
  - Fixed error in `foldedEpsi` in `foldedE.c`
  - Removed `if(m > 1)` from `getCoeffMat` in `foldedE.c`.
  - Switched `n/2` in `foldedEpsi` to `n/2.`.
  - Ensured that `sfs->u` is always set to a value in `getSfs`.
  - Replace `u = args->u` by `u = sfs->u` in `getCoeffMat` in `unfolded.c`.
  - Re-implemented `foldedEpsi` in `foldedE.c`
- Version 0.41 (February 22, 2018)
  - Changed `4.*u*(n/2.)` to `4.*u/(n/2.)` in `foldedEpsi` in `foldedE.c`. This was a bug in the computation of Ψ for the folded/even case.
- Version 0.42 (February 22, 2018)
  - Removed line `prevMinPsi = DBL_MAX` in `foldedE` in `foldedE.c`.
- Version 0.43 (February 22, 2018)
  - Added diagnostic output in case negative population sizes are found.
- Version 0.44 (February 22, 2018)
  - Ψ now also reported if only one level is included.
- Version 0.45 (February 23, 2018)
  - Included the `-n` option to allow negative population sizes.
- Version 0.46 (February 23, 2018)
  - Fixed `if(change > args->d)`-phrase in `unfolded` and `foldedE`. This lacked re-computation of the population sizes with the best new level added, and assignment of Ψ.
- Version 0.47 (February 26, 2018)
  - Reorganized code to remove duplication. The searching for best population sizes is now done in only one place, `getPopSizes` in `popSizes.c`.
  - Added printing of intermediate population sizes if the `-V` option is used.
- Version 0.48 (March 2, 2018)
  - Multi-threaded version.
- Version 0.49 (March 6, 2018)
  - Reverted to single-threaded behavior by removing `-t` from the options list and setting it to 1 in the background. This avoids the occasional race-conditions observed with the multi-threaded version.
- Version 0.50 (March 14, 2018)
  - Find number of levels through cross-validation (`-c`).
- Version 0.51 (March 14, 2018)
  - Find lambda through cross-validation (`-L`).
- Version 0.52 (March 17, 2018)
  - Include reporting of negative population sizes in verbose output (`-V`).

  - Changed `prevMinPsi = prevMinPsi;` in `getPopSizes` to `prevMinPsi = currMinPsi;`.

- Version 0.53 (April 7, 2018)
  - Always allow negative population sizes.
  - Cross-validation by default.
  - $\lambda = 0$ by default.
  - If negative population sizes are found, the program searches for optimal λ by going through $\lambda = 0..\mu$. This is slow and would need to be optimized in future versions.
  - Added Scripts for extracting quantiles from `epos` output.
- Version 0.54 (April 11, 2018)
  - Fixed array out-of-bounds error in `shuffleArr` in `sfs.c`.
- Version 0.55 (April 12, 2018)
  - "Unfolded" mode not working; so I removed that option for now.
- Version 0.56 (May 31, 2018)
  - Fixed Error in documentation.
- June 13, 2018
  - Posted `epos` on `github`.