

sfs: Compute Site Frequency Spectra

Bernhard Haubold

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

June 13, 2018

1 Introduction

For a sample of size n let $f_i(n)$ be the number of sites where i individuals carry a mutation. The vector

$$f_1(n), f_2(n), \dots, f_{n-1}(n)$$

is called the *site frequency spectrum*. The program `sfs` takes multiple haplotype samples simulated with `ms` as input and prints the average site frequency spectrum.

2 Getting Started

`sfs` 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 program's directory

```
cd sfs
```

and list its contents

```
ls
```

- Generate `sfs`

```
make
```

- List its options

```
./sfs -h
```

- Test the program on a data set consisting of 2 simulated haplotype samples:

```
./sfs data/msOdd.dat
```

- Compare to folded SFS

```
./sfs -f data/msOdd.dat
```

- Repeat for an even-numbered sample size

```
./sfs data/msEven.dat
```

```
./sfs -f data/msEven.dat
```

- Apply `sfs` to 10^4 newly simulated haplotypes

```
ms 10 10000 -t 10 | ./sfs
```

where `ms` is the coalescent simulator by Hudson (2002).

3 Listing

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

```
1  /***** sfs.c *****/
   * Description:
   * Author: Bernhard Haubold, haubold@evolbio.mpg.de
   * Date: Wed Sep 20 07:50:43 2017
   *****/
6  #include <stdio.h>
   #include <stdlib.h>
   #include "interface.h"
   #include "eprintf.h"
   #include "sample.h"
11  #include "spectrum.h"

   void scanFile(FILE *fp, Args *args){
       Sample *sa;
       Spectrum *sp;
16      int i, n;

       sa = initializeSample(fp, args);
       sp = newSpectrum(sa->nsam);
       n = 0;
21      while((sa = getSample(0)) != NULL){
           sp = computeSpectrum(sa, sp);
           n++;
       }
       if(!args->r){
26           for(i=0; i<sp->n; i++){
               sp->spectrum[i] /= (double)n;
           }
       if(args->f)
           foldSpectrum(sp);
31      printSpectrum(sp);
       freeSample();
   }

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

41      version = "0.7";
       setprogname2("sfs");
       args = getArgs(argc, argv);
       if(args->v)
46           printSplash(version);
       if(args->h || args->e)
           printUsage(version);
       if(args->t){
           sp = getArtificialSpectrum(args);
51           if(args->f)
```

```

        foldSpectrum(sp);
        printSpectrum(sp);
        return 0;
    }
56  if(args->numInputFiles == 0){
        fp = stdin;
        scanFile(fp, args);
    }else{
        for(i=0;i<args->numInputFiles;i++){
61         fp = fopen(args->inputFiles[i], "r");
            scanFile(fp, args);
            fclose(fp);
        }
    }
66  free(args);
    free(progname());
    return 0;
}

```

4 Change Log

- Version 0.1 (September 25, 2017)
 - First working version.
- Version 0.2 (October 23, 2017)
 - Polished interface.
- Version 0.3 (November 17, 2017)
 - Implemented folding of SFS ($-F$).
- Version 0.4 (November 29, 2017)
 - Enable analytic computation of SFS ($-T$ to specify θ and $-n$ to specify sample size).
- Version 0.5 (December 1, 2017)
 - Fixed error in folding.
- Version 0.6 (December 18, 2017)
 - Cleaned up interface.
- Version 0.7
 - Implemented $-r$ for printing raw counts.
- June 13, 2018
 - Posted on [github](#).

References

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