README for Stairway Plot v2.1

Installation

Stairway plot 2 was written in Java. It shall be runnable under any operation system with Java 1.7 or higher installed. No compilation needed. Simply unzip the zip file to a folder, such as "stairway_plot_v2.1" and follow the instruction below to run the program.

Content of the distribution

The distribution includes a folder called stairway_plot_es, which contains the required Java programs and libraries, two example "blueprint" files and a README file (this file).

Blueprint file

A "blueprint" file serves as a configuration file and input file. This distribution includes two example blueprint files, one for unfolded SFS (two-epoch.blueprint) and one for folded SFS (two-epoch_fold.blueprint). The easiest way to create a blueprint file is modifying an example blueprint file and saving it as a new file. The content and format of a blueprint file is:

1. Anything after a "#" is considered as comments and will not be used by the program.
2. **popid**: this is the id of the population. It shall be a word without white spaces
3. **nseq**: number of sequences or haploids with which you created your SFS. For *n* diploid individuals, nseq=2*n*.
4. **L**: length of sequence or more specifically the total number of observed nucleic sites (after filtering), including polymorphic and monomorphic. The number of polymorphic sites will be further separated by mutation size and described in **SFS**. That is, $L = \sum_{i=0}^{nseq} \xi_i$ or $L = \sum_{i=0}^{nseq/2} \eta_i$ (see definitions of ξ_i and η_i in **SFS** below).
5. **whether_folded**: whether the SFS is folded. Its value shall be either true or false.
6. **SFS**: snp frequency spectrum, i.e. number of singleton, number of doubleton, etc. (separated by white space). For unfolded SFS, it shall have nseq-1 numbers ($\xi_1, \xi_2, \dots, \xi_{nseq-1}$). For folded SFS, it shall have nseq/2 numbers ($\eta_1, \eta_2, \dots, \eta_{nseq/2}$). These can be floating point numbers, such as estimated SFS by program ANGSD.
7. **smallest_size_of_SFS_bin_used_for_estimation**: the smallest size of SFS bin to be used for estimation. By default, (for example, if commenting out this line, as in the example blueprint files) this number is 1. To exclude singletons from estimation, set this number to 2.
8. **largest_size_of_SFS_bin_used_for_estimation**: the largest size of SFS bin to be used for estimation. By default, (for example, if commenting out this line, as in the example blueprint files) this number is nseq-1 for unfolded SFS or nseq/2 for folded SFS. To exclude singletons from estimation, set this number to nseq-2 for unfolded SFS.
9. **pct_training**: percentage of sites used for training. By default, this number is 0.67, i.e. 2/3.

10. **nrand**: number of random break points for each try (separated by white space). We suggest to use 4 numbers, roughly (nseq-2)/4, (nseq-2)/2, (nseq-2)*3/4, nseq-2. The range of the numbers is between 1 and nseq-2. The smaller the number, the more likely the estimation may underfit the training data. The larger the number, the more likely the estimation may overfit the training data. Stairway plot will pick the best number (among the numbers specified here) to control overfitting based on the remaining 1-pct_training sites used as testing data.
11. **project_dir**: project directory. This folder will be created to store intermediate files and final results.
12. **stairway_plot_dir**: this is the folder containing the stairway plot programs and required libraries. By default, this is stairway_plot_es.
13. **ninput**: number of input files to be created for estimation. Similar to bootstrap, ninput subsamples of pct_training of the sites will be created. Estimations based on ninput input files will be used to produce the estimation (median) and 95% pseudo-CI. By default, this number is 200.
14. **random_seed**: an integer used as the random seed for the program. If this row is commented out, the program will randomly pick one seed for you.
15. **mu**: assumed mutation rate per site per generation. This is used to convert scaled results to those in Ne and year.
16. **year_per_generation**: assumed generation time (in years). This is used to convert scaled results to those in Ne and year.
17. **plot_title**: title shown in the plot.
18. **xrange**: Time (scaled in 1k year) range for plotting; format: xmin,xmax; use "0,0" for default (i.e. automatically determined by results).
19. **yrange**: Ne (scaled in 1k individual) range; format: xmin,xmax; "0,0" for default (i.e. automatically determined by results).
20. **xspacing**: use this to set X axis spacing for plotting. The larger the number the sparser the X ticks.
21. **yspaceing**: use this to set Y axis spacing for plotting. The larger the number the sparser the Y ticks.
22. **fontsize**: font size for plotting.

How to run

The simplest way to run stairway plot is use Stairbuilder.class to create a batch file and then run the batch file. First, create a blueprint file and copy it to the folder which contains the stairway_plot_es (say the folder name is stairway_plot_v2.1). Then, in the commandline environment, change directory to stairway_plot_v2.1 and run this command:

```
java -cp stairway_plot_es Stairbuilder blueprint_file
```

blueprint_file is the name of the blueprint file.

Example:

```
java -cp stairway_plot_es Stairbuilder two-epoch.blueprint
```

This step will create a batch file. If the operation system is DOS/Windows, the batch file will be named *blueprint_file*.bat, otherwise, it will be named *blueprint_file*.sh. Then run the batch file:

Under DOS/Windows, simply run the command

```
blueprint_file.bat
```

Under Unix-like system, run the command

```
bash blueprint_file.sh
```

Then wait the batch file to finish. In the meantime intermediate files including an additional batch file called *blueprint_file*.plot.bat will be created.

The result files are contained in the project_dir, including a summary of ninput runs for each number of nrand, which can be used to produce a plot using your own preferred plotting program, and two plots, one in png format and one in pdf format. The best number of nrand (this number can be found in the file *blueprint_file*.plot.bat) will be used to produce the "final" summary and plots. In the plot, the red line is the estimation (median), the thick grey lines define the 75% pseudo-CI and the light grey lines define the 95% pseudo-CI.

Paralleled running and advanced usage

For projects with nseq>200, you may consider speeding up the process by running the most time consuming estimation step in parallel. Open the batch file (either *blueprint_file*.bat or *blueprint_file*.sh) you can find that between the comment "Step 1: create .addTheta files" and the comment "Step 2: determine number of break points" are a serial of commands which uses Stairway_unfold_training_testing7.class or Stairway_fold_training_testing7.class to create an estimation (a .addTheta file) for an input file from the project_dir/input. All those commands are independent to each other and can be easily run in parallel, for example, on a computer cluster. Of course you may need to modify the classpath (the parameter after the "-cp") and the path to the input file (the first parameter after Stairway_unfold_training_testing7 or Stairway_fold_training_testing7) according to your own computing environment. After all the .addTheta files have been created, copy them to the project_dir/input folder and continue running the commands after the comment "Step 2: determine number of break points".

The batch file *blueprint_file*.plot.bat uses the estimation results produced by *blueprint_file*.bat. After all plots are created, you can change the plotting parameters in the corresponding blueprint file and rerun the commands after the comment "Step 2: create summaries and plots" to recreate summary and plot files.

The following is the description of the Java programs contained in this distribution. The programs are provided AS IS without charge and without warranty.

Stairbuilder.class

This program takes a blueprint file as input file and creates the batch file *blueprint_file*.bat or *blueprint_file*.sh.

Usage: `java -cp path_to_stairway_plot_es path_to_the_blueprint_file`

path_to_stairway_plot_es is the path to the folder stairway_plot_es.
path_to_the_blueprint_file is the path to the blueprint file.

Example:
```
java -cp stairway_plot_es two-epoch.blueprint
```

Stairpainter.class

This program takes a blueprint file as input file and creates the batch file *blueprint_file*.plot.bat or *blueprint_file*.plot.sh.

Usage: `java -cp path_to_stairway_plot_es path_to_the_blueprint_file`

path_to_stairway_plot_es is the path to the folder stairway_plot_es.
path_to_the_blueprint_file is the path to the blueprint file.

Stairway_unfold_training_testing7.class

This Java program takes an input file presenting the unfolded SFS of a sample and output a serial of estimations of θ assuming a multi-epoch model. It needs SwarmOps java library version 1.0 (or later) from the Hvass Laboratories, which can be downloaded from http://www.hvass-labs.org/projects/swarmops/java/. It also needs SFS_log_likelihood_problem_no_dim_penalty_unfold.class to be located within stairway_plot_es. It output a file with a name *input_file*.addTheta, which appends the θ estimations to the input file, where *input_file* is name of the input file.

Usage:

In DOS/Windows:

```
     java -cp path_to_stairway_plot_es;path_to_swarmops.jar
Stairway_unfold_training_testing7 input_file nrand pct_training
[optional_random_seed]
```

In Linux/Unix:

```
      java -cp path_to_stairway_plot_es:path_to_swarmops.jar
Stairway_unfold_training_testing7 input_file nrand pct_training
[optional_random_seed]
```

Example:

In DOS/Windows:

```
      java -cp stairway_plot_es\;stairway_plot_es\swarmops.jar
Stairway_unfold_training_testing7 two-epoch\input\two-epoch-200 28
0.67
```

In Linux/Unix:

```
      java -cp stairway_plot_es/;stairway_plot_es/swarmops.jar
Stairway_unfold_training_testing7 two-epoch/input/two-epoch-200 28
0.67
```

input_file is the name (including path, if not located in the current folder) of the input file.
nrand specifies the number of random breaks, as specified in the blueprint file.
pct_training is the percentage of sites used for training, as specified in the blueprint file.
[optional_random_seed] is an optional integer used as the seed for randomization.

Input file format:

Columns are separated by TABs.

First row: the first 5 columns are mandatory

1st col: popid as specified in the blueprint file

2nd col: nseq as specified in the blueprint file

3rd col: L as specified in the blueprint file. This can be a floating point number.

4th col: smallest_size_of_SFS_bin_used_for_estimation as specified in the blueprint file.

5th col: largest_size_of_SFS_bin_used_for_estimation as specified in the blueprint file.

Second row: SFS (unfolded) as specified in the blueprint file.

Output file format:

Columns are separated by TABs.

First row: the same as the first row of the input file (see above).

Second row: the break points used in this estimation.

6th row: the same as the second row of the input file (see above).

Beginning from the 7th row are the records of intermediate results:

2nd col: the number of groups of θ estimated (ngroup)

4th col: -log Likelihood of the testing data

6th col: -log Likelihood of the training data

beginning from the 7th col: groups of θ estimated

beginning from the ngroup+7_th col: the corresponding value of θ per site estimated for each group

Beginning from the row started with "final model:" are the records for the final results: The number after the "final model:" is the -log Likelihood of the testing data and the -log Likelihood of the training data. Following that is one row presenting the groups of θ estimated, and one row presenting the corresponding value of θ (over the whole length L) estimated for each group.

Stairway_fold_training_testing7.class

This Java program takes an input file presenting the folded SFS of a sample and output a serial of estimations of θ assuming a multi-epoch model. It needs SwarmOps java library version 1.0 (or later) from the Hvass Laboratories, which can be downloaded from http://www.hvass-labs.org/projects/swarmops/java/. It also needs SFS_log_likelihood_problem_no_dim_penalty_fold.class to be located within stairway_plot_es. It outputs a file with a name *input_file*.addTheta, which appends the θ estimations to the input file, where *input_file* is name of the input file.

Usage:

In DOS/Windows:

```
      java -cp path_to_stairway_plot_es;path_to_swarmops.jar
Stairway_fold_training_testing7 input_file nrand pct_training
[optional_random_seed]
```

In Linux/Unix:

```
      java -cp path_to_stairway_plot_es:path_to_swarmops.jar
Stairway_fold_training_testing7 input_file nrand pct_training
[optional_random_seed]
```

Example:

In DOS/Windows:

```
     java -cp stairway_plot_es\;stairway_plot_es\swarmops.jar
Stairway_fold_training_testing7 two-epoch _fold\input\two-epoch _fold-
200 28 0.67
```

In Linux/Unix:

```
     java -cp stairway_plot_es/;stairway_plot_es/swarmops.jar
Stairway_fold_training_testing7 two-epoch _fold/input/two-epoch _fold-
200 28 0.67
```

`input_file` is the name (including path, if not located in the current folder) of the input file.
nrand specifies the number of random breaks, as specified in the blueprint file.
pct_training is the percentage of sites used for training, as specified in the blueprint file.
[optional_random_seed] is an optional integer used as the seed for randomization.

Input file format:

Columns are separated by TABs.

First row: the first 5 columns are mandatory

1st col: popid as specified in the blueprint file

2nd col: nseq as specified in the blueprint file

3rd col: L as specified in the blueprint file. This can be a floating point number.

4th col: smallest_size_of_SFS_bin_used_for_estimation as specified in the blueprint file.

5th col: largest_size_of_SFS_bin_used_for_estimation as specified in the blueprint file.

Second row: SFS (folded) as specified in the blueprint file.

Output file format:

Columns are separated by TABs.

First row: the same as the first row of the input file (see above).

Second row: the break points used in this estimation.

6th row: the same as the second row of the input file (see above).

Beginning from the 7th row are the records of intermediate results:

2nd col: the number of groups of θ estimated (ngroup)

4th col: -log Likelihood of the testing data

6th col: -log Likelihood of the training data

beginning from the 7th col: groups of θ estimated

beginning from the ngroup+7_th col: the corresponding value of θ per site estimated for each group

Beginning from the row started with "final model:" are the records for the final results: The number after the "final model:" is the -log Likelihood of the testing data and the -log Likelihood of the training data. Following that is one row presenting the groups of θ estimated, and one row presenting the corresponding value of θ (over the whole length L) estimated for each group.

Stairway_output_summary_plot2.class

This Java program takes a blueprint file and the folder containing the θ estimations. It summarizes the estimations and plots the summary.

Usage:

In DOS/Windows:

```
    java -cp path_to_stairway_plot_es;path_to_ gral-core-
0.11.jar;path_to_ VectorGraphics2D-0.9.3.jar
Stairway_output_summary_plot2 path_to_the_blueprint_file
path_to_the_estimations
```

In Linux/Unix:

```
    java -cp path_to_stairway_plot_es:path_to_ gral-core-
0.11.jar:path_to_ VectorGraphics2D-0.9.3.jar
Stairway_output_summary_plot2 path_to_the_blueprint_file
path_to_the_estimations
```

Example:

In DOS/Windows:

```
    java -Xmx1g -cp stairway_plot_es\;stairway_plot_es\gral-core-
0.11.jar;stairway_plot_es\VectorGraphics2D-0.9.3.jar
Stairway_output_summary_plot2 two-epoch_fold.blueprint rand7
```

In Linux/Unix:

```
    java -Xmx1g -cp stairway_plot_es/:stairway_plot_es/gral-core-
0.11.jar:stairway_plot_es/VectorGraphics2D-0.9.3.jar
Stairway_output_summary_plot2 two-epoch_fold.blueprint rand7
```

`path_to_stairway_plot_es` is the path to the folder stairway_plot_es.
`path_to_gral-core-0.11.jar` is the path to the file gral-core-0.11.jar.
`path_to_VectorGraphics2D-0.9.3.jar` is the path to the file VectorGraphics2D-0.9.3.jar.
`path_to_the_blueprint_file` is the path to the blueprint file.
`path_to_the_estimations` is the path to the folder containing the .addTheta files. If this is not specified, the default value is “final”.

Output summary file format:

Columns are separated by TABs.

1st row: title row

Beginning from the 2nd row are the estimated measures of time and population sizes. Every two rows represent a "step" of the stairway plot.

> 1st col: time measured in the expected number of mutation(s) per site
>
> 2nd col: number of estimates used
>
> 3rd col: the median of the population size measured in θ per site
>
> 4th col: the 2.5 percentile estimation of the population size measured in θ per site
>
> 5th col: the 97.5 percentile estimation of the population size measured in θ per site
>
> 6th col: time measured in years
>
> 7th col: the median of the population size measured in individuals
>
> 8th col: the 2.5 percentile estimation of the population size measured in individuals
>
> 9th col: the 97.5 percentile estimation of the population size measured in individuals
>
> 10th col: the 12.5 percentile estimation of the population size measured in individuals
>
> 11th col: the 87.5 percentile estimation of the population size measured in individuals

Test run

This an example of test run on a PC with a Windows system

1. Unzip all files in stairway_plot_v2.1.zip to a folder, e.g. stairway_plot_v2.1.

2. Enter command line environment, change folder to stairway_plot_v2.1.

3. Run `java -cp stairway_plot_es Stairbuilder two-epoch.blueprint`.

4. Run `two-epoch.blueprint.bat`.

Contact

Xiaoming Liu, Ph.D.

Associate Professor,
USF Genomics,
College of Public Health,
University of South Florida
3720 Spectrum Boulevard, Suite 304,
Tampa, FL 33612-9415
Phone: 813-974-9865
Fax: 813-974-0992
Email: xiaomingliu@usf.edu
Lab page: liulab.science

Citation

A manuscript describing the stairway plot v2 and its applications is in preparation. At the meantime, please cite:

Liu X and Fu YX. (2015) Exploring population size changes using SNP frequency spectra. Nature Genetics. 47(5):555-559.