SweeD 3.0

Pavlos Pavlidis & Nikolaos Alachiotis

Contents

Chapter 1

Introduction

SweeD implements a composite likelihood ratio test which detects complete selective sweeps using Site Frequency Spectrum (SFS) patterns of single-nucleotide polymorphisms (SNPs). It is based on the SweepFinder algorithm impemented by Nielsen et al. [2005].

- SweeD is a command-line C program.
- Supported input file formats: SweepFinder format
- It can process several alignments in a single run.

Chapter 2

The Site Frequency Spectrum (SFS) pattern of selective sweeps

Figure 2.1 shows the generation of SNP patterns that can be used to localize a selective sweep. The figure consists of 6 snapshots that illustrate a population of chromosomes at different points in time. Snapshot 1 is the oldest since it refers to further in the past, whereas snapshot 6 refers to present. Thus, in snapshot 1, neutral mutations were segregating in a population. At some time point (snapshot 2), a beneficial mutation appears (black circle). Since this mutation is beneficial, the frequency of the chromosome that carries it will increase in the population (snapshot 3). However, recombination between the beneficial chromosome and the neutral chromosomes may occur. At snapshot 4, recombination occurs on the left side of the beneficial mutation while on snapshot 5 recombination occurs on the right side. Finally, at snapshot 6 we denote the region where the SFS has been shifted to low- and high-frequency derived variants.

2.1 The selective sweep model as implemented by Nielsen et al. 2005

Nielsen et al. [2005] implemented a simple model of a selective sweep.

Its main features are:

- sampling occurs at the time of selective sweep completion. This means that *no* mutations have occurred more recently than the selective sweep
- all observed SNPs were existing in the population *prior to* the selective sweep
- recombination is responsible for observing SNPs. If there is no recombination then no SNPs would be observable.

If the selective sweep is very recent, then a first approximation to model a selective sweep is to assume that the probability of each ancestral lineage to escape the selective sweep is $P_e = 1 - exp^{-\alpha d}$, where $\alpha = r/sln(2N)$, r is the recombination probability between two adjacent bases, s is the selection coefficient, N the effective population size, and d the distance in base-pairs between the SNP and the sweep location. This approximation means that no coalescent event has occurred between the present and the time point that the selective sweep is complete. Furthermore, it assumes that recombination occurs before (backward in time) any coalescent

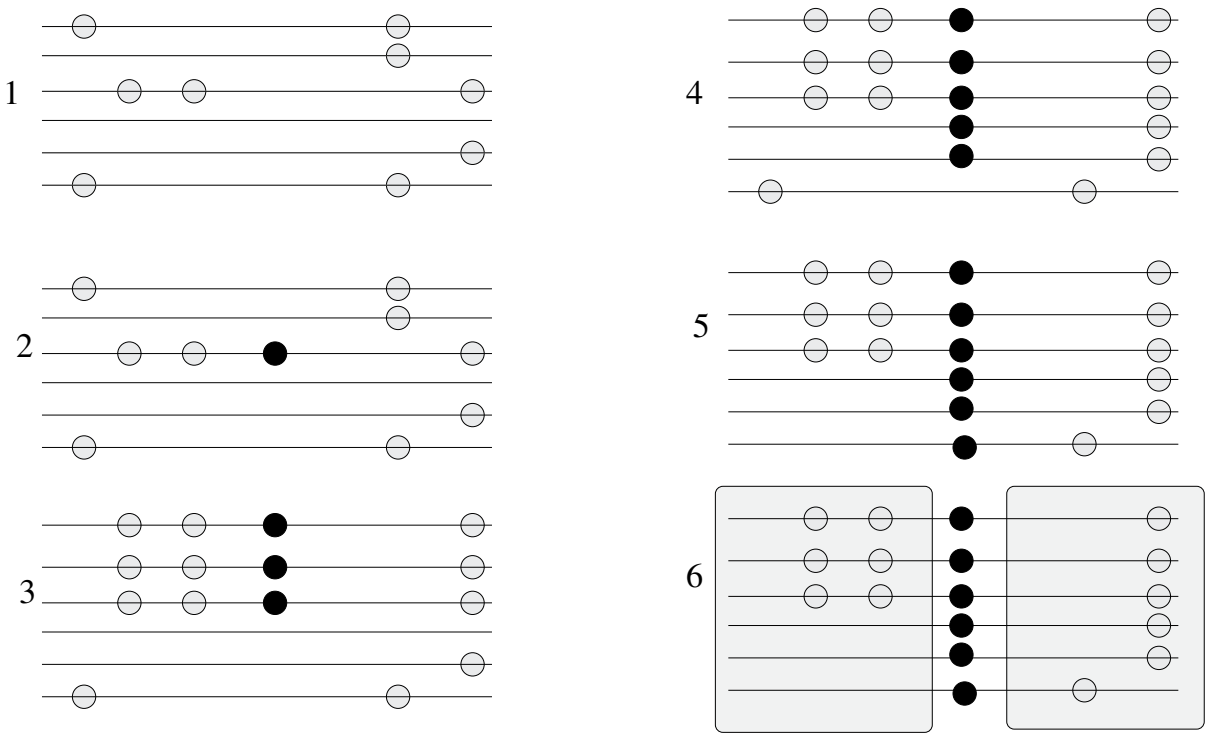

Figure 2.1: SFS patterns generated by a selective sweep. 1. Neutral mutations (light circles) are present in the population. 2. A beneficial mutation (black circle) appears in the population. 3. The frequency of the chromosome that carries the beneficial mutation increases. 4. Due to recombination (between chromosomes 5 and 6) neutral mutations that were previously on a neutral chromosome are located now on a beneficial chromosome. 5. Recombination occurs between chromosomes 5 and 6 and brings other neutral mutations on the beneficial chromosome. 6. The gray square denotes the region where the SFS has been shifted to low- and high-frequency dervied variants.

event during the selective sweep period, i.e. it is assumed that lineages escape the selective sweep independently.

Under these assumptions, the probability that k lineages escape the selective sweep is given by the binomial distribution:

$$P_e(k) = \binom{n}{k} P_e^k (1 - P_e)^{n-k}$$

If k lineages escape the selective sweep, then the ancestral sample prior to the sweep (forward in time) contains $H = min(n, k + 1)$ lineages. Given that the SFS before the selective sweep (forward in time) is given by $p = (p_1, p_2, \ldots, p_{n-1})$, then the probability of observing j lineages with the derived mutation in an ancestral sample of size H is given by:

$$p_{j,H} = \sum_{i=j}^{n-1} p_i \frac{\binom{i}{j}\binom{n-1}{H-j}}{\binom{n}{H}}$$

Finally, if the ancestral sample size is $k + 1$ and j lineages ($j \leq k + 1$) carry the derived mutation, then the probability that the line which initiated the selective sweep is one of the lines that carry the derived mutation is $j/(k + 1)$. This implies that the probability to observe a derived mutation of frequency B at the time point that the selective sweep completes is:

$$p_B = P_e(n)p_b + \sum_{k=0}^{n-1} P_e(k)(p_{B+1-n+k,k+1} \frac{B + 1 - n + k}{k + 1} + p_{B,k+1} \frac{k + 1 - B}{k + 1}).$$

For further information see the section **Test 2: Parametric approach** in Nielsen et al. [2005].

Assuming that the SFS for a sample of size n before the sweep (forward in time) is denoted by $p = (p_1, p_2, \ldots, p_{n-1})$.

Chapter 3

Features

3.1 Command line options

Typing `SweeD -h` or `SweeD -help` the following help message is displayed:

```
 SweeD | SweeD-P | SweeD-C | SweeD-P-C
 -name runName
 -input inputFile
 -grid gridNumber
[-folded]
[-monomorphic]
[-isfs inputSFS]
[-osfs outputSFS]
[-osf outputSF]
[-threads threadNumber]
[-checkpoint checkpointInterval]
[-eN timeT sizeX]
[-G rateG]
[-s sequences]
[-h|-help]
[-v|version]


-name <STRING>          Specifies a name for the run and the output files.

-input <STRING>         Specifies the name of the input alignment file.
                        Supported file formats: SF (Sweep Finder) format.

-grid <INTEGER>         Specifies the number of positions in the alignment
                        where the CLR will be computed.
```

```
-folded                 Considers the SFS folded (the ancestral and derived
                        states can not be distinguished).

-monomorphic            Includes the monomorphic sites in the analysis. The

-isfs <STRING>          Specifies the name of the input SFS file.

-osfs <STRING>          Specifies the name of the output SFS file.

-osf <STRING>           Specifies the name of the output SF file.

-threads <INTEGER>      Specifies the number of threads.

-checkpoint <INTEGER>   Specifies the checkpoint interval in seconds (default: 3600).

-eN <FLOAT> <FLOAT>     Sets population size to sizeX*N0 at time timeT, where N0 is
                        the present-day population size.

-G <FLOAT>              Sets the growth rate of the population size at time 0. The growth
                        rate continues to be exponential until the -eN command specifies a
                        constant population size.

-s <INTEGER>            Specifies the number of sequences when no input file is provided.

-h|-help                Displays this help message.

-v|-version             Displays version information.
```

3.2 Input file formats

3.2.1 The SweepFinder format

SweeD can process dataset in SweepFinder format. This data format consists of four columns, which describes four attributes for each SNP:

- **location**: the location of a SNP
- **x**: the number of sequences carry the derived allele for a SNP
- **n**: the number of *valid* sequences at a SNP
- **folded**: a binary character which denotes if the SNP is unfolded (0) or folded (1).

SweepFinder format assumes a mandatory header line:

location x n folded

SweepFinder format is assumed to describe binary data, i.e. at each SNP position two states must be present. For example, if a SNP is described by the vector A, A, A, A, C, C, C and the ancestral state is A, then the derived state (C) exists in 3 sequences and the ancestral state (A) in 4 sequences. If there are more than 2 states (e.g. A, A, A, T, C, C, C), and the ancestral state is A, then SweepFinder model does not hold strictly. However, in such cases one could assume A as ancestral state, and T,C as derived states. Then, there would be 3 sequences with the ancestral state (A) and 4 sequences with the derived state (T,C).

Column **n** describes the number of *v*alid sequences. This means the number of sequences that carry information which can be assumed as either derived or ancestral. For example, umbiguous states (e.g. N) is assumed to be invalid and thus it should not be counted to the total number of sequences.

Column **folded** describes whether the site frequency spectrum is folded or unfolded for the specific SNP. For example if the SNP is described by the vector A, A, A, A, C, C, C and the ancestral state is A, then the site frequency spectrum for the specific SNP is unfolded because the ancestral state in the SNP vector is unambiguously described. If the ancestral state is unknown, or if the ancestral state does not provide any information to distinguish between ancestral and derived state in the SNP vector then the SNP should be assumed folded. For example, if the SNP vector is A, A, A, A, C, C, C and the (defined by an outgroup) ancestral state is T, then we should consider that the state is folded, because there is no information to distinguish whether A or C are derived or ancestral.

Column **x** describes the number of sequences with the derived state. If the SNP is folded and the SNP is described by the A, A, A, A, C, C, C, then it is equivalent to assume $x = 3$ or $x = 4$.

3.2.2 FASTA format

Example of text file with one FASTA alignment

```
>D_sec
GTTGTTTAAATACCAATCGATTTGCATTCAAGTTTGAGAATTCTAGGATTTTTCAATTTT
>Dmel_A82_1230
GTTGTTTAAA-------------GCATTTAAT-GTTTCAGCCATACGACTCTTCA-----
>Dmel_A84_1230
GTTGATTAGA-------------GCATTTAAT-CTTTCAGCCATACGACTCTTCA-----
>Dmel_A95_1230
GTTGTTTAAA-------------GCATTTAAT-CTTTCAGCCATACGACTCTTCA-----
```

NOTICE:

'>' specifies the name of the sequence. This line is ignored. Thus, it is possible to have multiple words separated by a whitespace as a name of the sequence.

Example of text file with more than one FASTA alignments

SweeD can analyze files that contain more than one FASTA alignments. Alignments should be separated by an empty line starting with //. For

```
>D_sec
GTTGTTTAAATACCAATCGATTTGCATTCAAGTTTGAGAATTCTAGGATTTTTCAATTTT
>Dmel_A82_1230
GTTGTTTAAA-------------GCATTTAAT-GTTTCAGCCATACGACTCTTCA-----
>Dmel_A84_1230
GTTGATTAGA-------------GCATTTAAT-CTTTCAGCCATACGACTCTTCA-----
>Dmel_A95_1230
GTTGTTTAAA-------------GCATTTAAT-CTTTCAGCCATACGACTCTTCA-----
//
>D_seq1
GTTGTTTAAATACCAATCGATTTGCATTCAAGTTTGAGAATTCTAGGATTTTTCAATTTT
>D2
GTTGTTTAAA-------------GCATTTAAT-GTTTCAGCCATACGACTCTTCA-----
>D3
GTTGATTAGA-------------GCATTTAAT-CTTTCAGCCATACGACTCTTCA-----
>D4
GTTGTTTAAA-------------GCATTTAAT-CTTTCAGCCATACGACTCTTCA-----
```

3.2.3 ms-like format

ms-like format matches the output format of the widely used ms software [Hudson, 2002] (henceforth denoted as Hudson's ms). Hudson's ms implements coalescent simulations for various demographic scenarios. The software can be downloaded from `https://webshare.uchicago.edu/users/rhudson1/Public/ms.folder/ms.tar.gz`. Hudson's ms outputs binary data (0 and 1) instead of DNA data (A, C, G, or T). This is because an infinite site model is implemented. Thus, each site in the alignment will contain maximum two states. State 0 corresponds to no mutation while state 1 is used when a mutation has occured. Usually, state 1 is called 'derived' and state 0 is called 'ancestral'.

WARNING

Note that, the SweeD results *depend* on which state is the derived and which is the ancestral when the data is denoted as *unfolded*. This means that if your real data are folded, you should use the flag `-folded` with the ms-like data. In this case SweeD will ignore the distinction between 0 and 1.

Example of ms-like file

```
ms 5 2 -t 3
53303 53650 13864

//
segsites: 6
positions: 0.4478 0.5128 0.5537 0.6123 0.7253 0.7368
 000100
%% 101010
```

```
010001
101000
101010

//
segsites: 4
positions: 0.0747 0.1319 0.4368 0.5681
0000
1100
0000
0010
0011
```

The above example contains two binary alignments. The first one consists of 6 segregating sites while the second alignment of 4. The word 'segsites' defines the number of segregating sites. The word 'positions' denotes the relative positions of the segregating sites ranging from 0.0 to 1.0. In other words, the entire simulated alignment is assumed to be of length 1. Therefore, the SNP positions appear as floating-point numbers between 0.0 and 1.0.

WARNING

Hudson's ms outputs the relative positions using 4 decimal digits. This means that if you simulate many SNPs, then several of them will be exactly at the same alignment position. This might create problems to the calculation of the CLR when your analysis contains hundrends of thousands of SNPs.

To solve this problem, one can replace the line in ms.c that prints the positions of the SNPs:

- Open the ms.c file with your favorite editor. e.g. textpad, emacs, vim, gedit, ...
- Find line: `fprintf(pf,"%6.4lf ",posit[i] );`. This should be around line 191.
- Replace this line with: `fprintf(pf,"%e ",posit[i] );`
- Compile ms.c using the command `./clms`

3.2.4 MaCS-like format

MaCS [Chen et al., 2009] is a Markovian coalescent simulator. It is similar to Hudson's ms but it only generates approximations of the ancestral recombination graph by assuming that coalescent has a Markovian property along the sequence. Thus, MaCS can be many orders of magnitude faster than Hudson's ms when the recombination rate between the ends of the simulated regions is large (for example when whole chromosomes are simulated). On the other hand, MaCS is less accurate than Hudson's ms, especially when recombination values are small.

Example of MaCS-like file

```
COMMAND:        macs 40 22422 -t 0.0001 -r 0.043 -h 5e4 -eN 0.036744 0.00204
-eN 0.0375 8.0 -eN 0.1395 1.5 -s 417790 -i 2
SEED:   417790
SITE:   0           1.197754e-1 10000000000000000000000000000000000000
SITE:   1           0.232366639 00000000000000001000000000000000000000
SITE:   2           0.244833485 01101101000111010111100001011101000010001
SITE:   3            0.30002599 00000000000000000001000000000000000000
SITE:   4           0.878531842 11100000001000010000100000100110011000011
SITE:   5           0.914214234 10000100001000010001100011100100011101111
TOTAL_SAMPLES:  40
TOTAL_SITES:    6
BEGIN_SELECTED_SITES
0       1       2       3       4       5
END_SELECTED_SITES
SITE:   0           0.117200364 11001100000010100000010010010000000000000
SITE:   1           0.295165608 10001000001000010000110000010100000010001
SITE:   2           0.424410309 11101111001111111011110110101110110110011
SITE:   3           0.562284003 00000010000001001000000000000000000000000
SITE:   4           0.704560616 00000000100000000000000000000000000000000
TOTAL_SAMPLES:  40
TOTAL_SITES:    5
BEGIN_SELECTED_SITES
0       1       2       3       4
END_SELECTED_SITES
```

SweeD uses only the data in the lines that begin with the word 'SITE:'. The MaCS-like format is similar to the ms-like format but the SNPs appear transposed. In MaCS files, a line represents a SNP. For example, line:
`SITE: 0  0.117200364 11001100000010100000010010010000000000000`
represents the first SNP (SITE: 0) and the relative position of this SNP is 0.117200364.

File format recognition in SweeD

- FASTA files are recognized from the symbol '>'.
- ms-like files are recognized from the //.
- FASTA files with more than one alignments are recognized from the // as well. The '>' symbol appears before // in FASTA files and therefore no confussion between ms-like and FASTA files can occur. This means that there is no // before the first sequence.
- MaCS-like files are recognized from the word `COMMAND:`. Files with more than one alignments are recognized from: `SITE: 0`.

3.2.5 VCF format

SweeD supports the VCF (Variant Call Format) file format. VCF is a text file format. It contains meta-information lines, a header line, and then data lines each containing information about a position in the genome. An example is provided at `http://www.1000genomes.org/node/101` and is also given below:

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF ALT    QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
21     11 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
21     1237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
21     12347 microsat1 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

NOTICE

Different chromosomes (first column in VCF - #CHROM) are treated as different alignments. In the example above, this means that the first 5 lines refer to chromosome 20 and the last 3 lines refer to chromosome 21. These are considered as different datasets.

The current release of SweeD uses the OmegaPlus [Alachiotis et al., 2012] parser to parse the VCF file format. Thus, it is assumed that sequence information (phased or unphased) is provided. Future versions of SweeD will be based only on the REF, ALT and frequency information of the VCF file.

3.3 Output files

A single run of SweeD outputs 2 files:

- an information file (SweeD_Info.runName), which contains information related to the run of the program (the command line for instance)
- a report file (SweeD_Report.runName), which consists the main output file of the program (the score of the statistic at each position).

runName is the name of the run that is provided by the user via the **-name** argument.

3.3.1 Information file

The information file contains details related to the run of the program such as:

- the command line
- the number of sequences and SNPs of the alignment
- the number of non-polymorphic sites that were discarded
- the run time of *each* alignment
- the total run time of the program (for *all* alignments)

Example:

```
                                    ---------------

                                         SweeD
                                    ---------------


SweeD version 3.0 released by Nikolaos Alachiotis and Pavlos Pavlidis in November 2012.


Command:

       ./SweeD -input ../fastaex.fas -grid 140 -name test


Alignment 1
               Outgroup:             none

               Sequences:            10
               Sites:                150
               Discarded sites:      0

               Processing:           0.05 seconds

               Position:             1.030000e+02
               Likelihood:           3.029496e+00
               Alpha:                3.163686e+01


Total elapsed time 0.05 seconds.
```

3.3.2 Warning file

The warning file contains warnings to the user if more than one SNPs are located at the same alignment position.
Example:

```
// Alignment 1

 SNIPs 0 and 1 correspond to the same alignment position: 0
 SNIPs 2 and 3 correspond to the same alignment position: 9
```

Consecutive SNPs may be associated with the same alignment position when SweeD analyzes data generated with Hudson's ms or the MaCS software. For example, if the relative positions of SNPs i and j are 0.0012 and 0.0014 respectively, and the alignment length is 1000, then both SNPs will correspond to alignment position 1.

If more than one SNPs correspond to the same alignment position,
SweeD results are **NOT** seriously affected.

Note however that, if big datasets (e.g. several thousands of SNPs) that have been generated by Hudson's ms are analyzed, it is possible that a large number of SNPs will correspond to exactly the same alignment position (because of Hudson's ms 4-digit relative position precision). This may cause problems and lead to significant deviations in the results. (See warning in Subsection 3.2.3).

3.3.3 Report file

The report file consists the main output file of SweeD.

Standard report file form

For each alignment, the standard form report file contains:

1. the alignment positions where the SweeD score is calculated,
2. the corresponding likelihood value,
3. and the corresponding α value, which is a function of the selection coefficient, the recombination rate and the effective population size.

The columns are TAB delimited. Results from different alignments are separated by $alignmentIndex$, where $alignmentIndex$ is the index of the alignment in the input text file (e.g. //1, //2, ...)
Example:

```
//1
Position        Likelihood      Alpha
100     0.000000e+00    1.200000e+03
```

```
17450   5.967174e-02    4.425233e+02
34800   0.000000e+00    2.400001e-01
52150   0.000000e+00    1.200000e+03
69500   7.810233e-01    1.865193e+00
```

Chapter 4

How to use SweeD

4.1 Installation

SweeD can run on Linux platforms.

4.1.1 Linux platforms

To compile the source code use one of the provided makefiles as follows:

```
make -f makefile
```

Valid 'makefile' input is:

- Makefile.gcc: Use this makefile to generate the sequential version (SweeD).
- Makefile.PTHREADS.gcc: Use this makefile to generate the parallel pthreads version (SweeD-P).
- Makefile.CHECKPOINTS.gcc: Use this makefile to generate the checkpointable version (SweeD-C).
- Makefile.PTHREADS.CHECKPOINTS.gcc: Use this makefile to generate the checkpointable parallel pthreads version (SweeD-P-C).

To analytically compute the SFS use one of the following makefiles. This requires the MPFR library `www.mpfr.org`.

- Makefile_MPFR.gcc: Use this makefile to generate the sequential version (SweeD).
- Makefile_MPFR.PTHREADS.gcc: Use this makefile to generate the parallel pthreads version (SweeD-P).
- Makefile_MPFR.CHECKPOINTS.gcc: Use this makefile to generate the checkpointable version (SweeD-C).
- Makefile_MPFR.PTHREADS.CHECKPOINTS.gcc: Use this makefile to generate the checkpointable parallel pthreads version (SweeD-P-C).

- To analytically compute the SFS, you need to install the MPFR library. See details below.
- For the checkpointable version, the DMTCP library is required. See details below.

4.1.2 Installation of the MPFR library

In modern Ubuntu systems you can install the MPFR library by typing

```
sudo apt-get install libmpfr-dev
```

Then you can compile the code directly.

If you cannot use the automatic installation process to install the MPFR, but you have to download it manually, then do the following steps:

- download the MPFR library from `http://www.mpfr.org/mpfr-current/#download`.
- uncompress the file by `tar xvfz mpfr-3.1.1.tar.gz`
- follow the instructions of the MPFR library to install it. Typically, this include the steps: `./configure`, `make`, `sudo make install`. If you have no root priviledges you can install the library locally. This can be done by specifying the installation directory when you call the ./configure. e.g., `./configure --prefix=/home/USER/mpfr`, and then `make` and `make install`. The library then will be installed at the directory /home/USER/mpfr/, where USER is your user name.
- assume that the installation of the library is in the folder /home/USER/mpfr. Then inside /home/USER/mpfr there will be a folder called 'include' and a folder called 'lib'.
- to compile SweeD you must specify the 'include' and the 'lib' directories.
- in the makefile, at the end of the line that starts with CFLAGS, add -I/home/USER/mpfr/include
- in the makefile, immediately after the LIBRARIES type -L/home/USER/mfpr/lib. After that, type the -lmpfr -lgmp.

4.1.3 Installation of the DMTCP library

- download the library from the `http://dmtcp.sourceforge.net/`
- uncompress the file by typing `tar xvfz dmtcp-1.2.6.tar.gz`
- follow the instructions (written in the INSTALL file) to install the library locally.
- to use the library with the SweeD adapt the following line to point to your local installation path
  `DMTCPAWARELIB=/lhome/lalachins/Desktop/SWEED/dmtcp-1.2.6/lib/libdmtcpaware.a`

4.2 Execution examples

SweeD is a command-line tool. All supported command-line flags are provided in subsection 3.1. SweeD can be used to either analyze data in order to detect candidate selective sweep locations or to calculate the analytical SFS.

To analyze data the following command line flags are obligatory:

- The **-name** option specifies a suffix for the output files. This is to avoid overwriting files from previous analyses.
- **-grid** specifies the number of positions where the likelihood will be calculated. The first and last positions correspond to the first and last SNP positions respectively. Thus, the number of positions must be equal or greater than 2.
- **-input** specifies the input file. This can be in SweepFinder format (see section Features).

In the following we provide some command-line examples.

4.2.1 Standard

SweepFinder input data

To carry out a typical analysis, three input arguments are required: the input file in SweepFinder format, the number of positions to assess the SweeD at (-grid), a name for the run (-name) Example:

```
./SweeD  -name test -input data.SF -grid 5
```

This command line requires the computation of SweeD at 5 (-grid) positions along the alignment in the data.SF file (-input). The name of this run is 'test' (-name).

Binary data

In addition to the five basic arguments for DNA analyses, when binary data are used the length of the alignment must be also provided (-length). Example (ms file):

```
./SweeD -name test2 -input ms.out -length 10000 -grid 100
```

Example (MaCS file):

```
./SweeD -name test3 -input macs.out -length 10000 -grid 100
```

4.2.2 VCF file

```
./SweeD  -name test -input data.VCF -grid 15
```

4.2.3 FASTA file

```
./SweeD  -name test -input data.FA -grid 15
```

4.2.4 Using additional command line flags

- `-folded` Use this flag to assume that the data are folded. The algorithm will not distinguish between ancestral and derived states. For example, when you use the ms file format, the states '0' and '1' will be equivalent.

- `-monomorphic` Use this flag to include monomorphic sites in the analysis. Typically, in FASTA file formats (or VCF) there are several monomorphic sites. These sites will be excluded by default. If the `-monomorphic` flag is present, the monomorphic sites will be included as well.

- `-isfs` Use this flag to specify the SFS that will be used in the analysis. By default SweeD will compute the SFS from the data itself. Using the `-isfs` can be useful when you analyze simulated data and you intend to use the same SFS for all the simulated datasets. Another relevant application is when the SFS is computed analytically. Then, the SFS can be computed once and then used in several analyses by employing the `-isfs` flag.

- `-osfs` Output the SFS. This flag is required when you calculate the analytical SFS without analyzing data.

- `-threads` Specify the number of threads (when the PTHREAD version is called).

- `-eN` See below

- `-G` See below

- `-s` See below

4.2.5 Calculating the SFS analytically

SweeD can calculate the SFS analytically by implementing the theory developed by [Živković and Stephan, 2011]. The analytical SFS will be either used to analyze data (by providing the datafile with the `-input` flag, or it can be reported to an output file (by using the `-osfs` flag). It is possible to analyze data and report the SFS as well. To calculate the SFS analytically you need to specify the number of sequences and the demographic model. Currently, the following demographic models have been implemented:

- stepwise changes

- exponential growth in the most recent phase and stepwise changes before that

You need to use the `-s` flag *always* when you calculate the SFS analytically.

- To provide stepwise demographical changes use the `-eN` flag. The `-eN` flag is followed by two numbers, the time from the present day (in units of 4N) that the population size changed, and the relative size of the population (compared to the present day population size).

- To provide the exponential growth for *the most recent* phase use the `-G` flag. Note that the `-osfs` command line flag must be used when computing the SFS analytically *if no data are analyzed.*

WARNING
Important, the events must be given in a backward chronological order, i.e., from the most recent event to the most ancient event.

Below, we provide two examples that demonstrate the calculation of the analytical SFS.

```
./SweeD  -name test -s 100 -eN 0.1 0.2 -eN 0.3 5 -eN 1.2 0.8
./SweeD  -name test -s 100 -G 25 -eN 0.13 0.036 -eN 0.23 3.56
```

These scenarios are illustrated in figures 4.1 and 4.2.

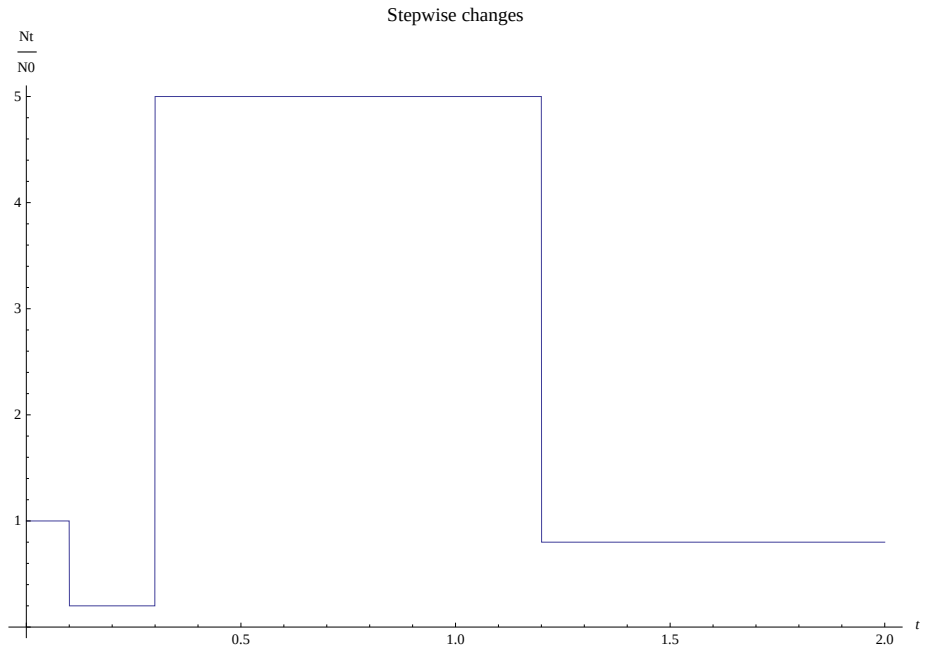

Figure 4.1: A stepwise demographic model related to the command *./SweeD -name test -s 100 -eN 0.1 0.2 -eN 0.3 5 -eN 1.2 0.8*

4.3 Execution details

1. The **-name** option specifies a suffix for the output files. This is to avoid overwriting files from previous analyses.

2. **-grid** specifies the number of positions where the likelihood will be calculated. The first and last positions correspond to the first and last SNP positions respectively. Thus, the number of positions must be equal or greater than 2.

3. **-input** specifies the input file. This can be in SweepFinder format (see section Features).

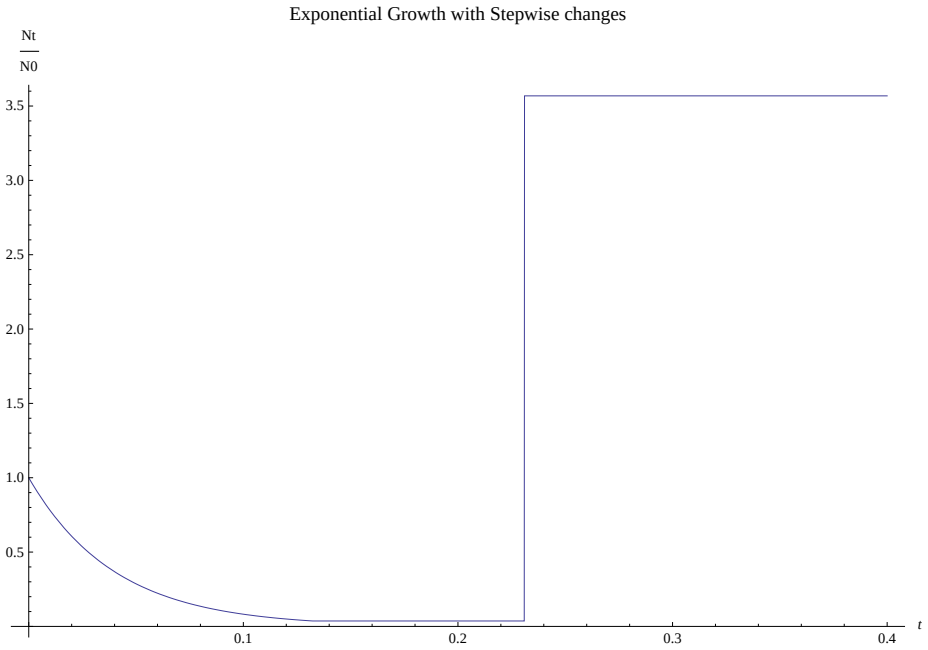

Figure 4.2: A demographic model with an exponential growth phase followed by stepwise demographic changes. It is associated with the command *./SweeD -name test -s 100 -G 25 -eN 0.13 0.036 -eN 0.23 3.56* .

Bibliography

N. Alachiotis, A. Stamatakis, and P. Pavlidis. Omegaplus: a scalable tool for rapid detection of selective sweeps in whole-genome datasets. *Bioinformatics*, 28(17):2274–2275, Sep 2012. doi: 10.1093/bioinformatics/bts419. URL `http://dx.doi.org/10.1093/bioinformatics/bts419`.

Gary K Chen, Paul Marjoram, and Jeffrey D Wall. Fast and flexible simulation of dna sequence data. *Genome Res*, 19(1):136–142, Jan 2009. doi: 10.1101/gr.083634.108. URL `http://dx.doi.org/10.1101/gr.083634.108`.

Richard R Hudson. Generating samples under a wright-fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, Feb 2002.

Rasmus Nielsen, Scott Williamson, Yuseob Kim, Melissa J Hubisz, Andrew G Clark, and Carlos Bustamante. Genomic scans for selective sweeps using snp data. *Genome Res*, 15(11):1566–1575, Nov 2005. doi: 10.1101/gr.4252305. URL `http://dx.doi.org/10.1101/gr.4252305`.

Daniel Živković and Wolfgang Stephan. Analytical results on the neutral non-equilibrium allele frequency spectrum based on diffusion theory. *Theor Popul Biol*, 79(4):184–191, Jun 2011. doi: 10.1016/j.tpb.2011.03.003. URL `http://dx.doi.org/10.1016/j.tpb.2011.03.003`.