Procedure Outline

Firstly, I use MSMS to simulate several selective sweeps through one well-mixed and one spatially segregated population. The sweeps use the following parameters. Note that for each parameter, the unscaled number denotes the original value of the parameter in question–for example, the true mutation rate–and the scaled number denotes the value of the parameter after factoring in MSMS's required constants.

Recombination Sites = 5000
Sample Size = 100, 2 samples per deme
Deme Count = 50 (only in segregated simulation. The segregated simulations occur in 1-D space, i.e a line.)
N_e = 100,000 (well-mixed) and 2,000 (per deme, spatially segregated)
$\theta = 4 * N_e *$ mutation rate across the whole genome ($250000 \cdot 10^{-6}$ unscaled, 2000 scaled)
$\rho = 4 * N_e *$ recombination rate across the whole genome ($2.379 \cdot 10^{-7}$ unscaled, 1903.2 scaled)
$M = 4 * N_e *$ migration rate (.15 unscaled, 1200 scaled)
$SAa = 4 * N_e *$ heterozygous selection coefficient (.01 unscaled, 1000 scaled)
$SAA = 4 * N_e *$ homozygous selection coefficient (.02 unscaled, 2000 scaled)
SF (Number of generations run after sweep completion) = 0

This simulation (in the well-mixed case) can be replicated via the following MSMS command:

```
msms 100 5 -N 100000 -t 2000 -r 1903.2 5000 -Sp 0.8 -SAA 2000 -SaA 1000 -SF 0
```

The following example of the output produced by the MSMS simulation is taken from the MSMS user manual:

```
segsites: 2
positions: 0.50061 0.70488
10
00
00
00
01
```

This output shows that there are two points along a genome at which mutations are present in a sample size of five individuals. Specifically, across the sample group, there are mutations at the points 0.50061 and 0.70488. Each row of 0s and 1s represents an individual in the sample group. A 0 represents the presence of the ancestral allele, whereas a 1 represents a mutant allele. MSMS presents these markers in the same order that it displays the segregating sites in. For example, in this simulation, an individual with the sequence 01 would have an ancestral allele at position 0.50061 and a derived allele at position 0.70488.

After collecting this output, I use a Python script to convert it into input recognizable by SweepFinder2. SweepFinder accepts a site frequency spectrum as input, in the form of a table with columns representing the position of each mutation, the number of times it appears, the sample size used to gather this information, and the site's polarization. The conversion script functions by creating a map of all the segregating sites present in a particular MSMS simulation and their associated mutation counts. It initializes all mutation counts to 0 and moves, individual by individual, through the output file. Whenever the algorithm encounters a 1, it increments the mutation count associated with the segregating site it is currently at by 1 and moves on to the next site in the output. If it encounters a 0, it moves onto the next site without making any changes to the map. After all the individuals in a simulation have been processed, the code iterates through the map and converts each (position, count) tuple into a row in the SweepFinder table. In each row, the position and mutation count are obtained from the map. Note that all positions are multiplied by 1,000,000, to convert them to integer bases. The sample size is set to the used value of 100, and the polarization is set to true, since all positions are known to contain derived alleles.

Finally, I execute SweepFinder using the file produced by the conversion algorithm using 100 test sites, which are positions at which SweepFinder guesses the sweep originated from. From the SweepFinder manual, ”test sites are equally spaced across the genomic region spanned by the positions in [the input file].” Using 100 sites causes the output to show an upwards spike in confidence as SweepFinder approaches the location of the sweep and helps better visualize the results. The following command replicates this effect, assuming the input file is named well-mixed.txt:

```
./SweepFinder2 -s 100 well-mixed.txt result-well-mixed.txt
```

Current Results

Running the converted MSMS output through SweepFinder produces two parameters. The likelihood ratio (LR) represents the probability that the observed count of mutant alleles were produced by selective pressure divided by the probability that they were produced by chance in a neutral model. Therefore, a higher LR value translate to a higher confidence in the presence of a selective force at any given point on a genome. SweepFinder is more confident in the presence of a non-neutral model when presented with MSMS input generated using a well-mixed population as opposed to a spatially segregated one. Well-mixed populations produce LR values on the order of 10^2, whereas populations existing in a 1-D or 2-D spatial structure produce LR values on the order of 10^0 and 10^1 respectively. SweepFinder estimates an alpha parameter that represents, in a population where all the parameters are known prior, the quantity $\alpha = r \ln(2N)/s$. Note that in this equation, r is the per-generation recombination rate between two bases, N is the effective population size, and s is the selection coefficient. The theoretical value of α is about 39 in the 1-D population and roughly 58 in the well-mixed population. Bizarrely, SweepFinder's α estimates in the 1-D population are all on the order of 10^1, but, in the well-mixed population, the estimates are on the order of 10^0, a less reasonable approximation than in the spatially segregated case. Additionally, despite the α parameter seemingly not varying with respect to the mutation rate, decreasing the mutation rate causes SweepFinder to produce higher alpha values. For instance, with a mutation rate of 10^{-7} across the entire genome, SweepFinder estimates α values on the order of 10^1 and 10^2 close to the sweep. If the mutation rate is decreased to 10^{-8}, this effect intensifies and all α estimates are on the order of 10^3.