Updated September 10, 2017

**MISCL (Mapping Incomplete Sweeps using Composite Likelihood)**

Ha My T. Vy and Yuseob Kim

Ewha Womans University

**INTRODUCTION**

MISCL is a computer program package developed for generating statistics, namely composite likelihood ratio (CLR) and standardized CLR, to detect signatures of incomplete selective sweeps from population genomic data (Vy and Kim 2015). It is assumed that a user is familiar with compiling and running programs under Linux or similar environments.

**A. CALCULATING COMPOSITE LIKELIHOOD RATIO**

1. Input format

In MISCL package, there are four C++ code files for processing four different types of input formats:

**1.1 ms-output format**

This is the output format of simulation programs ms (Hudson 2002) and msms (Ewing and Hermisson 2010) which generate samples under various population genetic models.

Example of ms-output format:

ms 5 2 -t 4 **-r 10 1000**

54262 1525 48309

//

segsites: 7

positions: 0.0098 0.0661 0.3254 0.5398 0.5401 0.5761 0.9994

0111000

0010111

0001001

1000001

0100001

//

segsites: 9

positions: 0.0179 0.1013 0.1954 0.2275 0.2431 0.3083 0.4161 0.5368 0.9078

111011001

000100100

111011001

000100111

111011001

This format represents the table of single nucleotide polymorphism (SNP) in a sample, with “1” denoting the derived allele and “0” the ancestral allele at a site. Locations of SNP sites in a sample, scaled between 0 and 1 by dividing the nucleotide position by the length of sequence, are given after “positions:”. The first line of the file repeats the command line running this simulation, with options specifying parameter values. Our program that calculates CLRs for samples reads this line to extract the recombination rate and sequence length (-r option of msms). Thus, to run the program for this input format, *the first line of the file must include –r option followed by recombination rate and sequence length (bold characters in the example).*

**1.2 ms-output with missing values format**

Example of ms-like format:

./ms 5 2 -t 4 -r 10 1000

34215 34973 9746

//

segsites: 13

positions: 0.0005 0.0494 0.1265 0.2399 0.3147 0.3439 0.4749 0.6235 0.6956 0.7280 0.7314 0.8128 0.9357

0000001010000

1100100NNNN11

0011010001100

10NN100100111

00110100N1100

//

segsites: 15

positions: 0.0903 0.0972 0.1153 0.1466 0.1706 0.3410 0.3594 0.4884 0.5275 0.6187 0.8199 0.9224 0.9326 0.9338 0.9963

110101100110000

00001110111NN01

0000111001NNNNN

111N00010001010

000N11101110101

This option is designed to process data files of ms-output format which contain missing bases (“N”) in addition to ancestral (“0”) and derived (“1”) alleles at polymorphic sites. It can be used when the actual sequence data is transformed into ms-output format. The advantage of using this over the FASTA format is the smaller size of input file and faster processing, since all monomorphic sites are omitted. Note that this format does not display missing bases at monomorphic sites in the data, which does not seriously impact the value of CLR calculated if missing bases are not very frequent (<10% of all bases). However, if there are too many missing bases in the data (for example, a block of sequence filled with gaps), CLR should be calculated skipping (i.e. without multiplying probabilities for) those sites with reduced information. This format does not provide information regarding which sites should be skipped. FASTA format below, however, allows to detect such monomorphic sites to be excluded in the calculation and is thus recommended for data with a large proportion of missing bases.

**1.3 FASTA format, without ancestral/derived state inference**

While the above formats (ms-output and ms-output with missing values) may contain multiple samples (typically representing multiple replicates from simulation), FASTA format here contains one set of sampled sequences (intended for actual sequence data).

Example of FASTA file format:

>firstMel\_2R

TCATATTGCTACGAAATTGGCCAAAACTCCCCAAATATGTAAATTCGTTTCTCCGATC

>secondMel\_2R

TCATATTGCTACGAAATTGGCCAAAACCCCCCAAATATGTTAATTCGTTTCTCCGATC

>thirdMel\_2R

CTAATTATACGCAAAAAATGCTGAAAGCCAGACCTAACCTTACTAGAGAGTATTAATC

The lines starting with “>” contain information about the following DNA sequences such as name and location. These lines are ignored by the program. Therefore, users can write the file *with aligned DNA sequences only, for example*:

TCATATTGCTACGAAATTGGCCAAAACTCCCCAAATATGTAAATTCGTTTCTCCGATC

TCATATTGCTACGAAATTGGCCAAAACCCCCCAAATATGTTAATTCGTTTCTCCGATC

CTAATTATACGCAAAAAATGCTGAAAGCCAGACCTAACCTTACTAGAGAGTATTAATC

For this format of input data, the program uses folded allele frequency spectrum to calculate composite likelihood (see Appendix). Taking each polymorphic site as the core SNP, composite likelihood ratio is calculated twice, first considering the minor allele at the site as the selected derived allele and then the major allele (assuming there are not more than two alleles at a single polymorphic site). The choice of ancestral/derived allele that yields a larger CLR is recorded and reported in the output file.

**1.4 FASTA format, ancestral/derived state available**

This option is used for analyzing a sample of DNA sequences when outgroup sequence is available to allow the inference of ancestral vs. derived alleles. The outgroup sequence can be provided in a separated file of FASTA format or in the same FASTA file together with sequences to be polarized. In the latter case, the last sequence in the file should be the outgroup sequence. In both cases, outgroup sequence should be aligned to match sequences in the sample. The program reads only the first sequence in the separate outgroup data file, even if there are more sequences. Bases on the outgroup sequence are then interpreted as ancestral states. If there are SNP sites in the sample where no information about ancestral allele is provided (outgroup sequence at the corresponding sites should be indicated by “N” instead of A, T, C, or G), the program calculates composite likelihood for those sites without ancestral/derived information as described above. As mentioned above in 1.3, the program for this option can also process data files with only aligned DNA sequences (no “>” lines).

2. Compiling and running

These are four C++ codes to calculate composite likelihood ratio for the four input formats mentioned above:

* MISCL\_msms.cpp for ms-output format
* MISCL\_ms\_missingdata.cpp for ms-like format
* MISCL\_fastaFile\_folded.cpp for FASTA format, without ancestral/derived state inference
* MISCL\_fastaFile\_unfold.cpp for FASTA format, ancestral/derived state available

All of them share the header file MISCL.h and can be run on Linux platform.

After choosing an input format, a user should compile the appropriate program:

g++ MISCL\_x.cpp -o execute\_file

where MISCL\_x.cpp is one of the four C++ files listed above. “execute\_file” is the name a user gives to the executable file.

Depending on which input file format was chosen, the command line to run the executable file requires different switches:

* For ms-output format and ms-output with missing values format, users must specify names of input and output files, for example:

./execute\_file input output -a max\_alpha

*The -a switch here is optional*. “max\_alpha” is the maximum value of alpha (α = 2*Ns*, *s* is selection coefficient and *N* is population size) to be explored in the maximization of CLR. If –a switch is omitted, the program uses the default value of α = 20,000.

* For FASTA format, without ancestral/derived state inference:

Similarly, names of input and output files must be given. In addition, *recombination rate per site per generation should be given through -r switch:*

./execute\_file input output -r rho -a max\_alpha

where rho = 4*N*ρ; ρ is the probability of recombination per generation between two adjacent nucleotide sites. -a switch here is again optional.

* For FASTA format, ancestral/derived state available:

Command line for this option *requires name of another input file*, which contains the outgroup DNA sequence (“input2” below):

./execute\_file input1 input2 output -r rho -a max\_alpha

3. Output format

An output file is a text file (.txt) which consists of six columns:

* The first column:

If the input file is ms or ms-like format, the first column of the output file (with header “REP”) shows the ordered index of replicate samples under analysis.

If the input file is FASTA format, the first column (with header “T” (type)) has one of two char values: “P” or “F”. “F” indicates that the ancestral allele at this site is unknown and thus folded site frequency was used to calculate CLR. “P” indicate that the ancestral allele at that site is known.

* The second column with header “Position” shows locations of SNPs on the genome.
* The third column with header “nd” records the count of derived allele at the focal site.
* The fourth column with header “na” records the count of ancestral allele at the focal site.
* The fifth column with header “alpha” shows the value of alpha (2Ns) which maximized the CLR assuming the SNP site is the putative site.
* The sixth column with header “L” shows the corresponding maximum CLR.

Note that since there are missing base calls, sum of nd and na can be smaller than sample size.

Examples:

Output file for ms or ms-like input format:

REP Position nd na alpha L

0 1300649 10 10 20000 1325.156250

0 1300713 8 12 20000 1873.203125

0 1302764 8 12 20000 2053.703125

0 1302806 7 13 20000 2338.812500

0 1303399 12 8 20000 1022.468750

0 1303906 7 13 20000 2335.500000

0 1305238 7 13 20000 2354.226562

0 1309108 7 13 20000 2287.648438

0 1309129 7 13 20000 2287.281250

Output file for FASTA input format:

T Position nd na alpha L

P 7087 9 11 1800 30.664773

P 12274 7 15 9800 299.186097

P 12459 7 15 7200 300.008314

P 18929 7 15 7200 300.237426

P 22140 7 15 7200 293.535063

F 27180 7 15 200 24.024651

F 64365 8 14 7200 433.435950

P 86218 8 14 200 -14.127104

**B. STANDARDIZATION**

1. Method

Since the distributions of CLR do not follow any known distribution, using conventional standardization method will bias the estimation of selective locus. The solution for this problem is transforming CLR into a new statistic:

****

where ****is the maximum CLR for a given focal site *x*and ****and ****are the mode and the 1-*ε* quantile of the distribution of****obtained from polymorphic sites whose derived allele frequency is *f* in simulated neutral data sets ( Vy and Kim 2015). Since the mean and mode of the distribution of **** are very close to each other, for calculation convenience without affecting the accuracy of CLR test, the program for standardization was written to use mean instead of mode for****.

2. Generating the table of means and quantiles of CLR distributions for different beneficial allele frequencies.

For standardization, the table of means and quantiles of CLR distributions should be obtained from CLRs calculated from data simulated under the null hypothesis (standard neutral model). Moreover, to generate smooth CLR distributions, the neutral data should be large enough to harbor more than around (3 x 105 x sample size) SNPs, to have at least 105 SNPs for each derived allele frequency class.

Users first have to take neutral data as the input file to run the code calculating CLRs (described in part A). The output file from that process (output1) is then taken as input file to run the C++ Code that generates table of means and quantiles of CLR distributions. Command lines to run the code are:

* Compile:

g++ distributionfeature.cpp -o executable\_name2

* Run the executable file:

./executable\_name2 output1 output2 -n nSample -p percentile

nSample here is the sample size and percentile is a value of *ε*. -p switch is optional. If it is omitted, the default value of *ε* = 0.002 is used.

After running the executable file, a table of means and quantiles of CLR distributions is recorded in a new output file (with name output2, given in the command). This new output file can be used as one of the input files to run the final step of standardization.

3. Standardization

Command lines:

g++ standardize.cpp -o standardize

./standardize input1 input2 output

input1 is the file which contains composite likelihood ratios to be standardized and input2 contains the table of means and quantiles of null CLR distributions (obtained at step 2).

Notice: input1 must be in the output format mention in part A. input1 and input2 should be generated for the same sample size to ensure the accuracy of the test.

**C. EXECUTION EXAMPLES**

Below are two examples to demonstrate how the program works. All input and output files are included in the package.

1. Generating CLRs from data of ms-like format:

The text file “ms\_like\_input.txt” in the package is a representative file of ms-like input data format. CLRs of SNPs of this data can be generated by using associated cpp file: “MISCL\_ms\_missingdata.cpp”.

By typing:

g++ MISCL\_ms\_missingdata.cpp -o mslike

./mslike ms\_like\_input.txt output.txt

user should be able to see a new text file made under the name of “output\_of\_ms\_like\_input.txt” with the following contents:

REP Position nd na alpha L

0 258 6 4 3200 35.994694

0 458 4 6 1800 5.639511

0 505 7 3 1800 -0.982491

0 647 4 6 1800 -23.543991

0 655 3 7 200 19.450256

0 752 6 4 200 -44.102478

0 779 6 3 200 -17.780548

0 787 7 3 200 -57.665039

0 808 3 7 800 -10.917984

0 844 3 7 200 7.445938

0 866 4 6 200 -43.224854

0 906 6 4 200 -58.651917

0 974 6 4 200 -95.488922

0 990 4 6 200 -9.836639

Meaning of each column is explained in A, part 3 (output format).

2. Generating CLRs from data of FASTA format:

“FASTA\_input.txt” and “outgoup\_seq.txt” are examples of FASTA input format and an aligned out group sequence.

By typing:

g++ MISCL\_fastaFile\_unfold.cpp -o unfold

./unfold FASTA\_input\_file.fasta outgroup\_seq.fasta output\_of\_FASTAfile\_unfold.txt -r 0.01

a new text file “output\_of\_FASTAfile\_unfold.txt” will be made as the following:

T Position nd na alpha L

P 258 6 4 200 -104.259147

F 404 6 4 200 -56.895948

P 458 4 6 200 -77.394750

P 505 7 3 200 -166.064304

P 647 4 6 200 -63.495995

P 655 3 7 200 18.100554

F 738 6 4 200 -33.510226

P 752 6 4 200 -36.134378

P 779 6 3 200 -14.507043

P 787 7 3 200 -62.862468

P 808 3 7 200 -16.743090

P 844 3 7 200 0.134269

P 866 4 6 200 -50.508689

P 906 6 4 200 -78.924230

P 974 6 4 200 -114.636718

P 990 4 6 200 -6.985607