MinHash Alignment Process (MHAP) Documentation

Release 2.1

Sergey Koren and Konstantin Berlin

December 24, 2016

Contents

CHAPTER 1

Overview

MHAP (pronounced MAP) is a reference implementation of a probabilistic sequence overlapping algorithm. Designed to efficiently detect all overlaps between noisy long-read sequence data. It efficiently estimates Jaccard similarity by compressing sequences to their representative fingerprints composed on min-mers (minimum k-mer).

MHAP is included within the Canu assembler. Canu can be downloaded here.

Contents:

1.1 Installation

1.1.1 Before your start

MHAP requires a recent version of the JVM (1.8u6+). JDK 1.7 or earlier will not work. If you would like to build the code from source, you need to have the JDK and the Maven build system available.

1.1.2 Prerequisites

- java (1.8u6+)
- maven (3.0+)

If you have not already installed the dependencies using maven, you will need an internet connection to do so during maven installation.

Here is a list of currently supported Operating Systems:

1. Mac OSX (10.7 or newer)
2. Linux 64-bit (tested on CentOS, Fedora, RedHat, OpenSUSE and Ubuntu)
3. Windows (XP or newer)

1.1.3 Installation

Pre-compiled

The pre-compiled version is recommended to users who want to run MHAP, without doing development. To download a pre-compiled tar run:

```
$ wget https://github.com/marbl/MHAP/releases/download/v2.1.1/mhap-2.1.1.jar.gz
```

And if `wget` not available, you can use `curl` instead:

```
$ curl -L https://github.com/marbl/MHAP/releases/download/v2.1.1/mhap-2.1.1.jar.gz
```

Then run

```
$ gunzip mhap-2.1.1.jar.gz
```

Now to run mhap run

Source

To build the code from the release:

```
$ wget https://github.com/marbl/MHAP/archive/v2.1.1.zip
```

If you see a certificate not trusted error, you can add the following option to wget:

```
$ --no-check-certificate
```

And if `wget` not available, you can use `curl` instead:

```
$ curl -L https://github.com/marbl/MHAP/archive/v2.1.1.zip > v2.1.zip
```

You can also browse the https://github.com/marbl/MHAP/tree/v2.1.1 and click on Downloads.

Once downloaded, extract to unpack:

```
$ unzip v2.1.1.zip
```

Change to MASH directory:

```
$ cd MHAP-2.1.1
```

Once inside the directory, run:

```
$ maven install
```

This will compile the program and create a target/mhap-2.1.1.jar file which you can use to run MHAP. The quick-start instructions assume you are in the target directory when running the program. You can also use the target/mhap-2.1.1.jar file to copy MHAP to a different system or directory. If you would like to run the validation utilties you must also download and build the SSW Library. Follow the instructions on the utilities page.

1.2 Quick Start

1.2.1 Running MHAP

Running MHAP provides command-line documenation if you run it without parameters. Assuming you have followed the installation instructions instructions, you can run:

```
$ java -jar mhap-2.1.1.jar
```

MHAP has two main usage modes, the main finds all overlaps between the input sequences. The second only constructs an index which can be subsequently reused.

1.2.2 Finding overlaps

```
$ java -Xmx32g -server -jar mhap-2.1.1.jar -s<fasta/dat from/self file> [-q<fasta/dat to file or dire
```

Both the -s and -q options can accept either FastA sequences or binary dat files (generated as described below). The -q option can accept either a file or a directory, in which case all FastA/dat files in the specified directory will be used. By default, only the sequences specified by -s are indexed and the sequences in -q are streamed against the constructed index. Generally, 32GB of RAM is sufficient to index 40K sequences. If you have more sequences, you can partition your data and run MHAP on the partitions. You can also increase the memory MHAP is allowed to use by changing the Xmx parameter to a larger limit.

The optional -f flag provides a file of repetitive k-mers which should be biased against selected as min-mers. The file is a two-column tab-delimited input specifying the kmer and the fraction of total kmers the k-mer comprises. For example:

```
$ head kmers.ignore
464
GGGGGGGGGGGG        0.0005
```

means the k-mer GGGGGGGGGGG represents 0.05% of the k-mers in the dataset (so if there are 100,000 total k-mers, it occurs 50 times). The first line specifies the total number of k-mer entries in the file.

It is also possible to use the k-mer list as a positive selection as was used in Carvalho et. al.. Specify the k-mer list as above and the flag:

```
--supress-noise 2
```

which will not allow any k-mer not in in the input file to be a minmer. The k-mers above –filter-threshold will be ignored as repeats.

```
--supress-noise 1
```

will downweight any k-mer not in the input file to bias against its selection as a minmer. The k-mers above –filter-threshold will be downeighted as repeats.

1.2.3 Constructing binary index

```
$ java -Xmx32g -server -jar mhap-2.1.jar -p<directory of fasta files> -q <output directory> [-f<kmer
```

In this use case, files in the -p directory will be converted to binary sketch files in the -q directory. Subsequent runs using these files (instead of FastA files) will be faster as the sequences no longer need to be sketched, only loaded into memory.

1.2.4 Output

MHAP outputs overlaps in a format similar to BLASR's M4 format. Example output:

```
[A ID] [B ID] [% error] [# shared min-mers] [0=A fwd, 1=A rc] [A start] [A end] [A length] [0=B fwd,
```

An example of output from a small dataset is below:

```
155 11 0.164156 206 0 69 1693 1704 0 1208 2831 5871
155 15 0.157788 163 0 16 1041 1704 1 67 1088 2935
155 27 0.185483 159 0 455 1678 1704 0 0 1225 1862
```

In this case sequence 155 overlaps 11, 15, and 27. The error percent is computed from the Jaccard estimate using mash distance.

1.2.5 Options

The full list of options is available via command-line help (–help or -h). Below is a list of commonly used options.

Usage 1 (direct execution): java -server -Xmx<memory> -jar <MHAP jar> -s<fasta/dat from/self file> [-q<fasta/dat to file>] [-f<kmer filter list, must be sorted>]

Usage 2 (generate precomputed binaries): java -server -Xmx<memory> -jar <MHAP jar> -p<directory of fasta files> -q <output directory> [-f<kmer filter list, must be sorted>]

–filter-threshold, default = 1.0E-5 [double], the cutoff at which the k-mer in the k-mer filter file is considered repetitive. This value for a specific k-mer is specified in the second column in the filter file. If no filter file is provided, this option is ignored.

–help, default = false Displays the help menu.

–max-shift, default = 0.2 [double], region size to the left and right of the estimated overlap, as derived from the median shift and sequence length, where a k-mer matches are still considered valid. Second stage filter only.

–min-olap-length, default = 116 [int], The minimum length of the read that used for overlapping. Used to filter out short reads from FASTA file.

–min-store-length, default = 0 [int], The minimum length of the read that is stored in the box. Used to filter out short reads from FASTA file.

–no-self, default = false Do not compute the overlaps between sequences inside a box. Should be used when the to and from sequences are coming from different files.

–no-tf, default = false Do not perform the tf weighing, in the tf-idf weighing.

–num-hashes, default = 512 [int], number of min-mers to be used in MinHashing.

–num-min-matches, default = 3 [int], minimum # min-mer that must be shared before computing second stage filter. Any sequences below that value are considered non-overlapping.

–num-threads, default = 8 [int], number of threads to use for computation. Typically set to #cores.

–ordered-kmer-size, default = 12 [int] The size of k-mers used in the ordered second stage filter.

–ordered-sketch-size, default = 1536 [int] The sketch size for second stage filter.

–repeat-idf-scale, default = 3.0 [double] The upper range of the idf (from tf-idf) scale. The full scale will be [1,X], where X is the parameter.

–repeat-weight, default = 0.9 [double] Repeat suppression strength for tf-idf weighing. <0.0 do unweighted MinHash (version 1.0), >=1.0 do only the tf weighing. To perform no idf weighting, do no supply -f option.

–settings, default = 0 Set all unset parameters for the default settings. Same defaults are applied to Nanopore and Pacbio reads. 0) None, 1) Default, 2) Fast, 3) Sensitive.

–store-full-id, default = false Store full IDs as seen in FASTA file, rather than storing just the sequence position in the file. Some FASTA files have long IDS, slowing output of results. This options is ignored when using compressed file format.

–supress-noise, default = 0 [int] 0) Does nothing, 1) completely removes any k-mers not specified in the filter file, 2) supresses k-mers not specified in the filter file, similar to repeats.

–threshold, default = 0.78 [double], the threshold cutoff for the second stage sort-merge filter. This is based on the identity score computed from the Jaccard distance of k-mers (size given by ordered-kmer-size) in the overlapping regions.

–version, default = false Displays the version and build time.

-f, default = “” k-mer filter file used for filtering out highly repetative k-mers. Must be sorted in descending order of frequency (second column).

-h, default = false Displays the help menu.

-k, default = 16 [int], k-mer size used for MinHashing. The k-mer size for second stage filter is seperate, and cannot be modified.

-p, default = “” Usage 2 only. The directory containing FASTA files that should be converted to binary format for storage.

-q, default = “” Usage 1: The FASTA file of reads, or a directory of files, that will be compared to the set of reads in the box (see -s). Usage 2: The output directory for the binary formatted dat files.

-s, default = “” Usage 1 only. The FASTA or binary dat file (see Usage 2) of reads that will be stored in a box, and that all subsequent reads will be compared to.

1.3 Utilities

1.3.1 Using MHAP extras

In addition to the main overlapping algorithm, MHAP indcludes several utilities for validating overlaps and simulating data.

1.3.2 Validating overlaps

Assuming you have a mapping of sequences to a truth (such as a reference genome) in BLASR’s M4 format, you can validate overlaps using MHAP’s EstimateROC utility which will compute PPV/Sensitivity/Specificity:

```
$ java -cp mhap-2.1.1.jar edu.umd.marbl.mhap.main.EstimateROC <reference mapping M4> <overlaps M4/MHA
```

The default minimum overlap length is 2000 and default number of trials is 10000. This will estimate sensitivity/specificity to within 1%. It can be increased at the expense of runtime. Specifying 0 will examine all possible N^2 overlap pairs.

The dynamic programming flag (true/false) will check overlaps not present in the reference mapping by running a Smith-Watermann alignment to identify the overlap specified. This step requires the SSW Library to be separately compiled and installed:

```
$ wget https://github.com/mengyao/Complete-Striped-Smith-Waterman-Library/archive/master.zip
$ unzip master.gip && cd  Complete-Striped-Smith-Waterman-Library-master/src
$ make all
$ cd /full/path/to/mhap/target/lib
$ ln -s /full/path/to/Complete-Striped-Smith-Waterman-Library-master/src/libsswjni.so
```

The verbose flag (true/false) enables logging to report true overlaps missing from the result and false-positives where no alignments could be found matching the required thresholds.

The minimum identity of the overlap (0.7 by default) is the lower bound for the sensitivity of an overlapper to evaluate. It is used to select matches to the reference that could be found by the overlapper. It is also used to threshold the minimum identity found by the Smith-Waterman alignment above.

The load all overlaps flag (true/false) will evaluate the specificity and PPV on all overlaps reported by the overlapper if enabled, not only those for good reads (where both reads were mapped to the reference in the truth set).

1.3.3 Simulating Data

MHAP includes a tool to simulate sequencing data with random error as well as estimate Jaccard similarity for the simulated data.

```
$ java -cp mhap-2.1.1.jar edu.umd.marbl.mhap.main.KmerStatSimulator <# sequences> <sequence length (
```

The error rates must be between 0 and 1 and are additive. Specifying 10% insertion, 2% deletion, and 1% substitution will result in sequences with a 13% error rate. If no reference sequence is given, completely random sequences are generated and errors added. Otherwise, random sequences are drawn from the reference and errors added. Errors are added randomly with no bias.

```
$  java -cp mhap-2.1.1.jar edu.umd.marbl.mhap.main.KmerStatSimulator <# trials> <kmer size> <sequenc
```

This usage will output a distribution of Jaccard similarity between a pair of overlapping sequences with the specified error rate (when using the specified k-mer size) and two random sequences of the same length. If no reference sequence is given, completely random sequences are generated and errors added, otherwise sequences are drawn from the reference. When one-sided error is specified (by typing true for the parameter), only one of the two sequences will have error simulated, matching a mapping of a noisy sequence to a reference. If a set of k-mers for filtering is given, they are excluded when computing Jaccard similarity, both between random and overlapping sequences.

1.4 Contact

1.4.1 Bugs, feature requests, comments:

If you encounter any problems/bugs, please check the known issues pages:

https://github.com/marbl/MHAP/issues

If not, please report the issue either using the contact information below or by submitting a new issue online.

Please include information on your run:

```
1) any output produced by MHAP

3) sample data, if possible, to reproduce the issue
```

Who to contact to report bugs, forward complaints, feature requests:

Konstantin Berlin: kberlin@gmail.com

Sergey Koren: sergek@umd.edu