SortMeDNA User Manual

Evguenia Kopylova
jenya.kopylov@gmail.com

July 24 2015, version 1.0

Contents

1 Introduction

Bonsai Bioinformatics Research Group
CRIStAL (UMR CNRS 9189 University of Lille), 59650 Villeneuve d'Ascq , France
Inria Lille Nord Europe, 59655 Villeneuve d'Ascq, France

The Knight Lab, University of California, San Diego, 9500 Gilman Dr, CA, La Jolla, USA

SortMeDNA is a genomic and metagenomic read mapper designed for reads produced by second- and third-generation sequencing technologies. It has been tested with Illumina, 454, Ion Torrent and PacBio data with read lengths ranging from 100 to 10,000 nucleotides. SortMeDNA takes as input a file of reads (fasta or fastq format) and a reference sequence(s) file, and outputs alignments in SAM and BLAST-like formats. Additionally, SortMeDNA can be used as a filter to output only a FASTA(Q) file of matching and non-matching reads.

Availability: `http://bioinfo.lifl.fr/sortmedna/`

For questions & help, please contact:

1. Evguenia Kopylova jenya.kopylov@gmail.com
2. Laurent Noe laurent.noe@lifl.fr
3. Mikael Salson mikael.salson@lifl.fr
4. Rob Knight robknight@ucsd.edu
5. Helene Touzet helene.touzet@lifl.fr

2 Installation

2.1 Install from tarball release

1. Download `sortmedna-1.0.tar.gz` from `https://github.com/ekopylova/sortmedna/releases`
2. Extract the source code package into a directory of your choice, enter `sortmedna-1.0` directory and type,

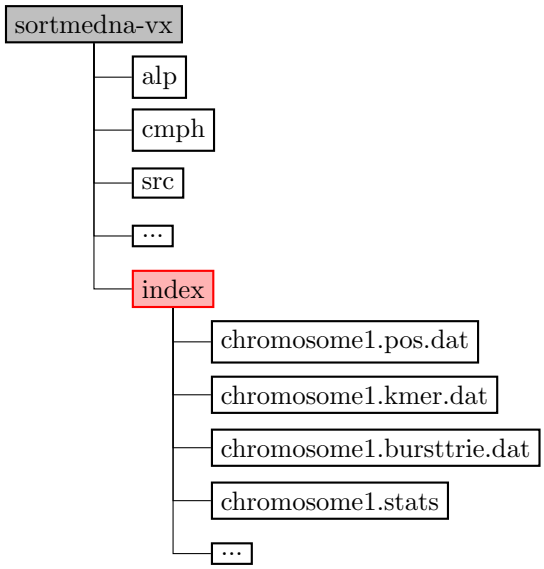
   ```
   > bash ./build.sh
   ```

3. At this point, two executables `indexdb_dna` and `sortmedna` will be located in the `sortmedna-1.0` directory. If the user would like to install the executables into their default installation directory (`/usr/local/bin` for Linux or `/opt/local/bin` for Mac) then type,

   ```
   > make install (with root permissions)
   ```

4. To begin using SortMeDNA, type '`indexdb_dna -h`' or '`sortmedna -h`'. Databases must first be indexed using `indexdb_dna`.

Figure 1: `/sortmedna-vx` directory tree. The index for the reference sequence `chromosome1.fasta` (located anywhere in the system) is placed into the `/index` subdirectory of `/sortmedna-vx`.



2.2 Install development version from git

1. Clone the sortmedna directory to your local system

```
> git clone https://github.com/ekopylova/sortmedna.git
```

2. Build sortmedna

```
> cd sortmedna
> bash ./build.sh
```

2.3 Uninstall

If the user installed SortMeDNA using the command '`make install`', then they can use the command '`make uninstall`' to uninstall SortMeDNA (with root permissions).

3 How to run SortMeDNA

3.1 Index the reference database: command `indexdb_dna`

3.1.1 `indexdb_dna` man page

The executable `indexdb_dna` indexes a reference database.

To see the man page for `indexdb_dna`,

```
>> ./indexdb_dna -h

  Program:     SortMeDNA version 1.0-dev
  Copyright:   2013-2015 Bonsai Bioinformatics Research Group
               LIFL, University Lille 1, CNRS UMR 8022, INRIA Nord-Europe
               2015 Knight Lab, Department of Pediatrics, UCSD, La Jolla
  Disclaimer:  SortMeDNA comes with ABSOLUTELY NO WARRANTY; without even the
               implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
               See the GNU Lesser General Public License for more details.
  Contact:     Evguenia Kopylova, jenya.kopylov@gmail.com
               Laurent Noe, laurent.noe@lifl.fr
               Mikael Salson, mikael.salson@lifl.fr
               Rob Knight, robknight@ucsd.edu
               Helene Touzet, helene.touzet@lifl.fr


  usage:   ./indexdb_dna --ref db.fasta,db.idx [OPTIONS]:

  ------------------------------------------------------------------------------------------------------------
  | parameter        value           description                                                    default |
  ------------------------------------------------------------------------------------------------------------
     --ref           STRING,STRING   FASTA reference file, index file                               mandatory
                                      (ex. --ref /path/to/file1.fasta,/path/to/index1)
                                       If passing multiple reference sequence files, separate
                                       them by ':',
                                      (ex. --ref /path/to/file1.fasta,/path/to/index1:
                                      /path/to/file2.fasta,path/to/index2)
   [OPTIONS]:
     --fast          BOOL            suggested option for aligning ~99% related species             off
     --sensitive     BOOL            suggested option for aligning ~75-98% related species          on
     --tmpdir        STRING          directory where to write temporary files
     -m              INT             the amount of memory (in Mbytes) for building the index        3072
     -L              INT             seed length                                                    18
     --max_pos       INT             maximum number of positions to store for each unique L-mer     250
                                      (setting --max_pos 0 will store all positions)
     -v              BOOL            verbose
     -h              BOOL            help
```

The estimated amount of memory required for building an index will be below 100*(length of reference sequence). Although the index can be fragmented into chunks of size specified by option `-m INT`, at this point only separate sequences can be split and not full sequences themselves. An error will be issued if some sequence cannot fit into the specified memory.

3.1.2 Example 1: index a reference sequence database

In this example we use the default memory limit for the index, being 3GB, and we build an index using the `--sensitive` option suitable for metagenomics and population diversity studies. Otherwise, the option `--fast` can be chosen for genome resequencing projects.

```
 >> ./indexdb_dna --ref arabidopsis_thaliana_genome.fasta --sensitive -v

  Collecting sequence distribution statistics ..  done  [2.787837 sec]

  start index part # 0:
    (1/3) building burst tries .. done  [38.326844 sec]
    (2/3) building CMPH hash .. done  [153.426536 sec]
    (3/3) building position lookup tables .. done [43.084145 sec]
```

```
  total number of sequences in this part = 1
    writing kmer data to ~/index/arabidopsis_thaliana_genome.kmer_0.dat
    writing burst tries to ~/index/arabidopsis_thaliana_genome.bursttrie_0.dat
    writing position lookup table to ~/index/arabidopsis_thaliana_genome.pos_0.dat

start index part # 1:
  (1/3) building burst tries .. done  [23.876025 sec]
  (2/3) building CMPH hash .. done  [55.050192 sec]
  (3/3) building position lookup tables .. done [26.181276 sec]
  total number of sequences in this part = 1
    writing kmer data to ~/index/arabidopsis_thaliana_genome.kmer_1.dat
    writing burst tries to ~/index/arabidopsis_thaliana_genome.bursttrie_1.dat
    writing position lookup table to ~/index/arabidopsis_thaliana_genome.pos_1.dat

start index part # 2:
  (1/3) building burst tries .. done  [28.457168 sec]
  (2/3) building CMPH hash .. done  [69.546082 sec]
  (3/3) building position lookup tables .. done [36.065969 sec]
  total number of sequences in this part = 1
    writing kmer data to ~/index/arabidopsis_thaliana_genome.kmer_2.dat
    writing burst tries to ~/index/arabidopsis_thaliana_genome.bursttrie_2.dat
    writing position lookup table to ~/index/arabidopsis_thaliana_genome.pos_2.dat

start index part # 3:
  (1/3) building burst tries .. done  [22.938778 sec]
  (2/3) building CMPH hash .. done  [75.675569 sec]
  (3/3) building position lookup tables .. done [24.874294 sec]
  total number of sequences in this part = 1
    writing kmer data to ~/index/arabidopsis_thaliana_genome.kmer_3.dat
    writing burst tries to ~/index/arabidopsis_thaliana_genome.bursttrie_3.dat
    writing position lookup table to ~/index/arabidopsis_thaliana_genome.pos_3.dat

start index part # 4:
  (1/3) building burst tries .. done  [35.091087 sec]
  (2/3) building CMPH hash .. done  [77.267817 sec]
  (3/3) building position lookup tables .. done [39.220080 sec]
  total number of sequences in this part = 3
    writing kmer data to ~/index/arabidopsis_thaliana_genome.kmer_4.dat
    writing burst tries to ~/index/arabidopsis_thaliana_genome.bursttrie_4.dat
    writing position lookup table to ~/index/arabidopsis_thaliana_genome.pos_4.dat
    writing SAM header and nucleotide distribution statistics to
    ~/index/arabidopsis_thaliana_genome.stats
  done.
```

The indexed databases (ex. `arabidopsis_thaliana_genome.bursttrie_0.dat`) will be stored in the directory '`/some/path/to/sortmedna/index`'.

3.2 Map reads against the indexed reference database: command `sortmedna`

3.2.1 `sortmedna` man page

The executable **sortmedna** maps HTS reads against an indexed reference database.

To see the man page for **sortmedna**,

```
>> ./sortmedna -h
```

```
  Program:     SortMeDNA version 1.0-dev
  Copyright:   2013-2015 Bonsai Bioinformatics Research Group:
               LIFL, University Lille 1, CNRS UMR 8022, INRIA Nord-Europe
               2015 Knight Lab, Department of Pediatrics, UCSD, La Jolla
  Disclaimer:  SortMeDNA comes with ABSOLUTELY NO WARRANTY; without even the
               implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
               See the GNU Lesser General Public License for more details.
  Contact:     Evguenia Kopylova, jenya.kopylov@gmail.com
               Laurent Noe, laurent.noe@lifl.fr
               Mikael Salson, mikael.salson@lifl.fr
               Rob Knight, robknight@ucsd.edu
               Helene Touzet, helene.touzet@lifl.fr


  usage:   ./sortmedna --ref db.fasta,db.idx --reads file.fa --aligned base_name_output [OPTIONS]:

  ------------------------------------------------------------------------------------------------------------
  | parameter          value           description                                                  default |
  ------------------------------------------------------------------------------------------------------------
     --ref             STRING,STRING   FASTA reference file, index file                             mandatory
                                         (ex. --ref /path/to/file1.fasta,/path/to/index1)
                                         If passing multiple reference files, separate
                                         them using the delimiter ':',
                                         (ex. --ref /path/to/file1.fasta,/path/to/index1:
                                         /path/to/file2.fasta,path/to/index2)
     --reads           STRING          FASTA/FASTQ reads file                                       mandatory
     --aligned         STRING          aligned reads filepath + base file name                      mandatory
                                         (appropriate extension will be added)

   [COMMON OPTIONS]:
     --other           STRING          rejected reads filepath + base file name
                                         (appropriate extension will be added)
     --fastx           BOOL            output FASTA/FASTQ file                                      off
                                         (for aligned and/or rejected reads)
     --sam             BOOL            output SAM alignment                                         off
                                         (for aligned reads only)
     --SQ              BOOL            add SQ tags to the SAM file                                  off
     --blast           INT             output alignments in various Blast-like formats
                                        0 - pairwise
                                        1 - tabular (Blast -m 8 format)
                                        2 - tabular + column for CIGAR
                                        3 - tabular + columns for CIGAR and query coverage
     --log             BOOL            output overall statistics                                    off
     --num_alignments  INT             report first INT alignments per read reaching E-value        -1
                                        (--num_alignments 0 signifies all alignments will be output)
       or (default)
     --best            INT             report INT best alignments per read reaching E-value         1
                                         by searching --min_lis INT candidate alignments
                                        (--best 0 signifies all candidate alignments will be searched)
     --min_lis         INT             search all alignments having the first INT longest LIS       2
                                         LIS stands for Longest Increasing Subsequence, it is
```

```
                                  computed using seeds' positions to expand hits into
                                  longer matches prior to Smith-Waterman alignment.
 --print_all_reads BOOL           output null alignment strings for non-aligned reads          off
                                  to SAM and/or BLAST tabular files
 --paired_in       BOOL           both paired-end reads go in --aligned fasta/q file           off
                                  (interleaved reads only, see Section 4.2.4 of User Manual)
 --paired_out      BOOL           both paired-end reads go in --other fasta/q file             off
                                  (interleaved reads only, see Section 4.2.4 of User Manual)
 --match           INT            SW score (positive integer) for a match                      2
 --mismatch        INT            SW penalty (negative integer) for a mismatch                 -3
 --gap_open        INT            SW penalty (positive integer) for introducing a gap          5
 --gap_ext         INT            SW penalty (positive integer) for extending a gap            2
 -N                INT            SW penalty for ambiguous letters (N's)                       scored as --mismatch
 -F                BOOL           search only the forward strand                               off
 -R                BOOL           search only the reverse-complementary strand                 off
 -a                INT            number of threads to use                                     1
 -e                DOUBLE         E-value threshold                                            1
 -m                INT            INT Mbytes for loading the reads into memory                 1024
                                   (maximum -m INT is 4096)
 -v                BOOL           verbose                                                      off


[ADVANCED OPTIONS] (see SortMeDNA user manual for more details):
 --passes          INT,INT,INT    three intervals at which to place the seed on the read       L,L/2,3
                                  (L is the seed length set in ./indexdb_dna)
 --edges           INT            number (or percent if INT followed by % sign) of             4
                                  nucleotides to add to each edge of the read
                                  prior to SW local alignment
 --num_seeds       INT            number of seeds matched before searching                     2
                                  for candidate LIS
 --full_search     BOOL           search for all 0-error and 1-error seed                      off
                                  matches in the index rather than stopping
                                  after finding a 0-error match (<1% gain in
                                  sensitivity with up four-fold decrease in speed)
 --pid             BOOL           add pid to output file names                                 off


[HELP]:
  -h               BOOL           help
  --version        BOOL           SortMeDNA version number
```

The command `sortmedna` takes as input a reference sequence file (in fasta format) and a set of HTS letter reads (in fasta or fastq format). The indexed reference sequences created by `indexdb_dna` are loaded into `sortmedna` independently.

In the following set of examples we will show how to use SortMeDNA as a filter and as an aligner. We will use the TAIR10 *A. thaliana* genome available here `ftp://ftp.arabidopsis.org/home/tair/Sequences/whole_chromosomes/` and 100,000 Illumina reads from the study SRR519675 (includes 5-8 *A. thaliana* plants).

3.2.2 Example 2: aligning reads using the SAM format

The reference index was built using the `--sensitive` option. The options highlighted in blue are the default options, they do not need to provided in the command line.

```
>> ./sortmedna --ref arabidopsis_thaliana_genome.fa --reads ./100000_SRR519675.fasta --match 2
              --mismatch -3 --gap_open 5 --gap_ext 2 -e 1 -a 1 -v --aligned aligned_reads --sam
```

```
Program:     SortMeDNA version 1.0-dev
Copyright:   2013-2015 Bonsai Bioinformatics Research Group:
             LIFL, University Lille 1, CNRS UMR 8022, INRIA Nord-Europe
             2015 Knight Lab, Department of Pediatrics, UCSD, La Jolla
Disclaimer:  SortMeDNA comes with ABSOLUTELY NO WARRANTY; without even the
             implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
             See the GNU Lesser General Public License for more details.
Contact:     Evguenia Kopylova, jenya.kopylov@gmail.com
             Laurent Noe, laurent.noe@lifl.fr
             Mikael Salson, mikael.salson@lifl.fr
             Rob Knight, robknight@ucsd.edu
             Helene Touzet, helene.touzet@lifl.fr


size of reads file: 16201782 bytes
partial section(s) to be executed: 1 of size 16201782 bytes

Computing Gumbel parameters ... done [0.17 sec]
seed length = 18
number of seeds = 2
pass 1 = 18, pass 2 = 9, pass 3 = 3
edges = 4 (as integer)

Begin partial reads section # 1:
Time to mmap reads and set up pointers             [0.03 sec]
  Begin analysis of: arabidopsis_thaliana_genome.fa
  Time to load the index part 1/5                  [35.56 sec]
  Begin index search ...  done                     [66.95 sec]
  Time to load the index part 2/5                  [26.42 sec]
  Begin index search ...  done                     [72.14 sec]
  Time to load the index part 3/5                  [31.96 sec]
  Begin index search ...  done                     [60.35 sec]
  Time to load the index part 4/5                  [26.42 sec]
  Begin index search ...  done                     [94.32 sec]
  Time to load the index part 5/5                  [37.43 sec]
  Begin index search ...  done                     [107.69 sec]
  Total number of reads mapped (in this file section):  98712
  Time to output reads to file                     [5.18 sec]
```

The first 10 lines of the resulting SAM alignment file contain the following input,

3.2.3 Example 3: aligning reads using a BLAST-like format

The reference index was built using the `--sensitive` option. The options highlighted in blue are the default options, they do not need to provided in the command line.

```
>> ./sortmedna --ref arabidopsis_thaliana_genome.fa --reads ./100000_SRR519675.fasta --match 2
              --mismatch -3 --gap_open 5 --gap_ext 2 -e 1 -a 1 -v --aligned aligned_reads --blast

  Program:     SortMeDNA version 1.0-dev
```

```
>> head -n 10 aligned_reads.sam

1. @HD VN:1.0 SO:unsorted
2. @SQ SN:Chr1 LN:30427671
3. @SQ SN:Chr2 LN:19698289
4. @SQ SN:Chr3 LN:23459830
5. @SQ SN:Chr4 LN:18585056
6. @SQ SN:Chr5 LN:26975502
7. @SQ SN:chloroplast LN:154478
8. @SQ SN:mitochondria LN:366924
9. @PG ID:sortmedna VN:1.0 CL:./sortmedna --ref arabidopsis_thaliana_genome.fa \
          --reads ./100000_SRR519675.fasta --match 2 --mismatch -3 \
          --gap_open 5 --gap_ext 2 -e 1 -a 1 -v \
          --aligned aligned_reads --sam
10. SRR519675.23 16 Chr1 3408525 255 100M * 0 0 \
GGAATGAAAAGAGCATCCGCCTTTGGTGTACTTGAGCTCTCTTCGTCATCAGTGAAAAATGGAACCTACAAAGAACATTTTAAGATTTAAACATATAGAA \
* AS:i:200 NM:i:0



  Copyright:   2013-2015 Bonsai Bioinformatics Research Group:
               LIFL, University Lille 1, CNRS UMR 8022, INRIA Nord-Europe
               2015 Knight Lab, Department of Pediatrics, UCSD, La Jolla
  Disclaimer:  SortMeDNA comes with ABSOLUTELY NO WARRANTY; without even the
               implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
               See the GNU Lesser General Public License for more details.
  Contact:     Evguenia Kopylova, jenya.kopylov@gmail.com
               Laurent Noe, laurent.noe@lifl.fr
               Mikael Salson, mikael.salson@lifl.fr
               Rob Knight, robknight@ucsd.edu
               Helene Touzet, helene.touzet@lifl.fr


  size of reads file: 16201782 bytes
  partial section(s) to be executed: 1 of size 16201782 bytes

  Computing Gumbel parameters ... done [0.15 sec]
  seed length = 18
  number of seeds = 2
  pass 1 = 18, pass 2 = 9, pass 3 = 3
  edges = 4 (as integer)

  Begin partial reads section # 1:
  Time to mmap reads and set up pointers             [0.03 sec]
    Begin analysis of: arabidopsis_thaliana_genome.fa
    Time to load the index part 1/5                  [36.71 sec]
    Begin index search ...  done                     [66.18 sec]
    Time to load the index part 2/5                  [26.16 sec]
    Begin index search ...  done                     [72.16 sec]
    Time to load the index part 3/5                  [31.19 sec]
    Begin index search ...  done                     [59.13 sec]
    Time to load the index part 4/5                  [25.79 sec]
    Begin index search ...  done                     [92.82 sec]
    Time to load the index part 5/5                  [35.30 sec]
    Begin index search ...  done                     [106.72 sec]
```

```
Total number of reads mapped (in this file section):  98712
Time to output reads to file                      [5.85 sec]
```

The first alignment (corresponding to the SAM output in the previous example) in the resulting BLAST-like alignment file is,

```
Sequence ID: Chr1 CHROMOSOME dumped from ADB: Jun/20/09 14:53; last updated: 2009-02-02
Query ID: SRR519675.23 HWI-ST378_0095:2:1101:2087:2123 length=100
Score: 165 bits (200) Expect: 1.09e-35 strand: -

Target:  3408525    GGAATGAAAAGAGCATCCGCCTTTGGTGTACTTGAGCTCTCTTCGTCATCAGTGAAAAAT    3408584
                    ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Query:         1    GGAATGAAAAGAGCATCCGCCTTTGGTGTACTTGAGCTCTCTTCGTCATCAGTGAAAAAT    60

Target:  3408585    GGAACCTACAAAGAACATTTTAAGATTTAAACATATAGAA    3408624
                    ||||||||||||||||||||||||||||||||||||||||
Query:        61    GGAACCTACAAAGAACATTTTAAGATTTAAACATATAGAA    100
```

3.2.4 Example 4: only filter the reads into FASTA/Q file, no alignment output

The reference index was built using the `--sensitive` option. The options highlighted in blue are the default options, they do not need to provided in the command line. When we only specify the `--fastx` option, the program runs much quicker than with either `--sam` or `--blast` options, since we stop searching for further alignments after the first match.

```
>> ./sortmedna --ref arabidopsis_thaliana_genome.fa --reads ./100000_SRR519675.fasta --match 2
               --mismatch -3 --gap_open 5 --gap_ext 2 -e 1 -a 1 -v --aligned aligned_reads --fastx

  WARNING: option '--other' has been left blank, no output file for rejected reads will be created.

  Program:     SortMeDNA version 1.0-dev
  Copyright:   2013-2015 Bonsai Bioinformatics Research Group:
               LIFL, University Lille 1, CNRS UMR 8022, INRIA Nord-Europe
               2015 Knight Lab, Department of Pediatrics, UCSD, La Jolla
  Disclaimer:  SortMeDNA comes with ABSOLUTELY NO WARRANTY; without even the
               implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
               See the GNU Lesser General Public License for more details.
  Contact:     Evguenia Kopylova, jenya.kopylov@gmail.com
               Laurent Noe, laurent.noe@lifl.fr
               Mikael Salson, mikael.salson@lifl.fr
               Rob Knight, robknight@ucsd.edu
               Helene Touzet, helene.touzet@lifl.fr


  size of reads file: 16201782 bytes
  partial section(s) to be executed: 1 of size 16201782 bytes

  Computing Gumbel parameters ... done [0.16 sec]
  seed length = 18
  number of seeds = 2
  pass 1 = 18, pass 2 = 9, pass 3 = 3
  edges = 4 (as integer)

  Begin partial reads section # 1:
   Time to mmap reads and set up pointers             [0.03 sec]
     Begin analysis of: arabidopsis_thaliana_genome.fa
     Time to load the index part 1/5                  [36.30 sec]
     Begin index search ...  done                     [40.66 sec]
     Time to load the index part 2/5                  [26.35 sec]
     Begin index search ...  done                     [35.73 sec]
     Time to load the index part 3/5                  [33.23 sec]
     Begin index search ...  done                     [37.68 sec]
     Time to load the index part 4/5                  [24.91 sec]
     Begin index search ...  done                     [35.37 sec]
     Time to load the index part 5/5                  [37.14 sec]
     Begin index search ...  done                     [38.75 sec]
     Total number of reads mapped (in this file section):  98712
     Time to output reads to file                     [0.44 sec]
```

3.2.5 Filtering paired-ended reads

When writing aligned and non-aligned reads to FASTA/Q files, sometimes the situation arises where one of the paired-end reads aligns and the other one doesn't. Since SortMeDNA looks at each read individually, by default the reads will be split into two separate files. That is, the read that aligned will go into the `--aligned` FASTA/Q file and the pair that didn't align will go into the `--other` FASTA/Q file.

This situation would result in the splitting of some paired reads in the output files and not optimal for users who require paired order of the reads for downstream analyses.

For users who wish to keep the order of their paired-ended reads, two options are available. If one read aligns and the other one not then,

(1) `--paired-in` will put both reads into the file specified by `--aligned`

(2) `--paired-out` will put both reads into the file specified by `--other`

The first option, `--paired-in` is optimal for users that want all reads in the `--other` file to be non-rRNA. However, there are small chances that reads which do not align will also be put into the `--aligned` file.

The second option, `--paired-out` is optimal for users that want only aligned reads in the `--aligned` file. However, there are small chances that reads which are aligned will also be put into the `--other` file.

If neither of these two options is added to the `sortmedna` command, then aligned and non-aligned reads will be properly output to the `--aligned` and `--other` files, possibly breaking the order for a set of paired reads between two output files.

It's important to note that regardless of the options used, the `--log` file will always report the true number of reads aligned (not the number of reads in the `--aligned` file).

4 SortMeDNA advanced options

`--num_seeds INT`

The threshold number of seeds required to match in the primary seed-search filter before moving on to the secondary seed-cluster filter. More specifically, the threshold number of seeds required before searching for a longest increasing subsequence (LIS) of the seeds' positions between the read and the closest matching reference sequence. By default, this is set to 2 seeds.

`--passes INT,INT,INT`

In the primary seed-search filter, SortMeDNA moves a seed of length L (parameter of `indexdb_dna`) across the read using three passes. If at the end of each pass a threshold number of seeds (defined by `--num_seeds`) did not match to the reference database, SortMeDNA attempts to find more seeds by decreasing the interval at which the seed is placed along the read by using another pass. In default mode, these intervals are set to $L, L/2, 3$ for Pass 1, 2 and 3, respectively. Usually, if the read is highly similar to the reference database, a threshold number of seeds will be reached in the first pass.

`--edges INT(%)`

The number (or percentage if followed by %) of nucleotides to add to each edge of the alignment region on the reference sequence before performing Smith-Waterman alignment. By default, this is set to 4 nucleotides.

`--full_search FLAG`

During the index traversal, if a seed match is found with 0-errors, SortMeDNA will stop searching for further 1-error matches. This heuristic is based upon the assumption that 0-error matches are more prelevant than 1-error matches. By turning it off using the `--full_search` flag, the sensitivity may increase (often by less than 1%) but with up to four-fold decrease in speed.