Human Sequence Removal
National Center for Biotechnology Information

Author: Stephen Sherry, PhD (point of contact)

Version: 1.0
Effective Date:

1 Abstract

Sequencing for the Human Microbiome Project (HMP) is being done using Illumina and 454 next-generation sequencing platforms that generate short reads in large volumes. As the sequences submitted contain a small percentage of reads for the human from whom the sample was collected, it is ethically necessary to provide the full set of sequences under controlled access, and to provide only the sequences that have been screened for human “contamination” publicly. Best Match Tagger (BMTagger) is an efficient tool that discriminates between human reads and microbial reads without doing an alignment of all reads to the human genome.

2 Introduction

This SOP describes Best Match Tagger (BMTagger) tool run at NCBI. Given FASTA, FASTQ files or SRA accession of microbiome dataset, this tool produces list of reads that are most probably human contaminants and should not be disclosed to public.

3 Requirements

3.1 Program requirements

BMTagger requires the following programs to be available in the path:

- makeblastdb and blastn: these can be downloaded from blast distribution: ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/LATEST/
- extract_fullseq
- bmfilter
- srprism
- bmtool: needed to prepare the bitmask file for the reference genome.

3.3 Memory/Disk space requirements

Programs run from this script (bmfilter, srprism) require about 8.5Gb memory and three times as much harddisk space for index data.
Disk space needed for temporary files depends on input, and is typically the same size as that of the input for metagenomic datasets.

4 Procedure

The following steps are to be carried out once per reference genome:

4.1 Make index for bmfilter

```
  bmtool -d <reference.fa> -o <reference.bitmask> -A 0 -w 18
```

Where reference.fa is fasta file for the screening database. For HMP, this can be the human genome.
Output is a binary file generated in reference.bitmask
To make a compressed index, add flag "-z" to the above command line.

4.2 Make index for srprism

```
  srprism mkindex -i <reference.fa> -o <reference.srprism> -M 7168
```

This generates files with the prefix reference.srprism

4.3 Make blastdb for blast

```
  makeblastdb -in <reference.fa> -dbtype nucl
```

This generates database files for blastn.

4.4 Run BMTagger

The commands depend on the data source for reads.

- For single reads in fasta format, the command is:

```
    bmtagger.sh -b reference.bitmask -x reference.srprism \
                -T tmp -q0 -1<file.fa> -o<file.out>
```

- For paired reads in fasta format, the command is:

```
    bmtagger.sh -b reference.bitmask -x reference.srprism \
                -T tmp -q0 -1<mate1.fa> -2<mate2.fa> -o<file.out>
```

- For single reads in fastq format, the command is:

```
    bmtagger.sh -b reference.bitmask -x reference.srprism \
                -T tmp -q1 -1<file.fq> -o<file.out>
```

- For paired reads in fastq format, the command is:

```
    bmtagger.sh -b reference.bitmask -x reference.srprism \
       -T tmp -q1 -1<mate1.fq> -2<mate2.fq> -o<file.out>
```

- For reads read directly from SRA, the command is:

```
    bmtagger.sh -b reference.bitmask -x reference.srprism \
       -T tmp -A <run> -o <outdir>
```

- In all above scenarios, -b, -x, and -T specify the index for bmfilter, the index for srprism, and the directory to use for temporary files, respectively.
- If no temporary directory is specified, current working directory is used.

- Flag -q of 0 and 1 specify fasta and fastq input files, respectively.
- Output specified by -o is a file name if input is fasta or fastq, and it is a directory if the input is a run. The output for, say run SRR059480, when -o is myresults will be a file myresults/SRR059480.blacklist that contains the SRA indexes of reads found to be human rather than the full id. *Output files with inputs as fasta or fastq contain the ids of reads found to be human.*

Environment

```
    PATH
        is used to find programs called from script
    TMPDIR
        if set is used to initialize temporary directory, otherwise
        /tmp is used
    SRPRISM
        if set specifies name and optianally path to srprism
    BMFILTER
        if set specifies name and optionally path to bmfilter
    EXTRACT_FA
        if set specifies name and optionally path to extract_fullseq
    BLASTN
        if set specifies name and optionally path to blastn
```

Config files

At start time bmtagger.sh looks for file bmtagger.conf and if it is present imports it. Also every time option -C is used, bmtagger.sh tries to parse it, and ends with error if file is not found.

Config file is regular shell script which may set any variables specified in "Environment" section, plus any of following:

```
bmfiles
    if set specifies bmfilter bitmap file
blastdb
    if set specifies blastdb for blastn
srindex
    if set specifies srprism index file
```

5 Implementation

BMTagger is available at ftp://ftp.ncbi.nlm.nih.gov/pub/agarwala/bmtagger

6 Discussion

BMTagger attempts to first discriminate between human reads and microbial reads by comparing the 18mers found in the query with those in the human genome. If it fails to make a determination, then we use an alignment procedure that guarantees to find all matches with up to two errors, if such an alignment exists. We find that our heuristic is at least an order of magnitude faster than using megablast for 454, or BWA for Illumina reads (data not shown). The reads can be presented to the tagger as fasta or fastq files, or it can also retrieve reads directly from SRA.

The BMTagger script carries out the following steps:

1. Generate a random string that will be used to name temporary files created in the rest of the process.
2. Generate *bmfilter* command using the input parameters specified. User can specify reads as fasta or fastq files, or specify the SRA run acces- sion as the source of reads. For SRA run accessions, the process uses SRA toolkit to access the reads.
3. Using *bmfilter*, classify reads as foreign or human, and for reads that cannot be classified, generate subsequences of length 32 bp with a distance of 4 bases between the previous and the next, except the last one if sequence length is not a multiple of 32. Low-complexity subsequences are not generated.
4. Run *srprism* on subsequences for the first mate looking for an ali- gnment with at most one error. If the reads are paired, then remove reads found as human using first mate from the subsequences for the second mate and run *srprism* on the remaining subsequences. Running *srprism* requires making an index for the genome first.
5. Combine outputs from *bmfilter* and *srprism*, and remove temporary files.

7 Related Documents & References

Rotmistrovsky, K and Agarwala, R. (2011) BMTagger: Best Match Tagger for removing human reads from metagenomics datasets, Bioinformatics, Unpublished.

Li, H and Durbin, R. (2009) Fast and accurate short read alignment with Burrows- Wheeler transform, Bioinformatics, 25(14), 1754–1760.

SRA Toolkit:
http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=show&f=software&m=software&s=software

8 Revision History

Version	Author/Reviewer	Date	Change Made
1.0		9/22/2011	Establish SOP