DeepSNVMiner 1.0 Manual

Manual Table of Contents:

1) About DeepSNVMiner

DeepSNVMiner is a flexible open-source software package capable of detecting SNVs and small indels in sub-sets of cell populations. DeepSNVMiner expects sequence data containing UID barcodes by default (whose exact length must be defined by the user) although it is possible to run without UID tags and simply utilize a combination of the start and end bases to generate a unique tag. DeepSNVMiner workflow consists of six main blocks: fastq QC and filtering, grouping reads by barcode, alignment, calling variants, reporting, and optional graphing (requires R).

2) Installation

DeepSNVMiner is available from github at the following url https://github.com/mattmattmattmatt/DeepSNVMiner and when installed is roughly 400Mb including test data.

There are two ways to install:

1. If you have git installed, checkout the code:

```
$ git clone https://github.com/mattmattmattmatt/DeepSNVMiner
$ cd DeepSNVMiner
```

2. To download files as a zip go to https://github.com/mattmattmattmatt/DeepSNVMiner and select the ‘Download ZIP’ button. Open the zip file and type

```
$ cd DeepSNVMiner-master
```

To check whether the installation is successful for default case run:

```
$ ./run_deepseq.pl
```

Running with test data:
To test the installation using sample data we first need to configure

```
$ ./configure_deepseq.pl -test
$What is the path to samtools [default=/usr/bin/samtools]?/path1/samtools
$What is the path to bwa [default=/usr/bin/bwa]?/path2/bwa
```

The configuration script should next build the bwa index and exit. Next we need to run the test suite.

```
$ ./test_deepseq.pl
```

Results are in the sample/results directory. The test should complete with the message 'Installation Success!'

Running with your data:
To set the local system to use your data you need to set paths to bwa, samtools, bwa index files, and reference fasta (saves having to pass the same arguments each time)

```
$ ./configure_deepseq.pl
$What is the path to samtools [default=/usr/bin/samtools]?/path1/samtools
$What is the path to bwa [default=/usr/bin/bwa]?/path2/bwa
$What is the path to single reference fasta file?/path3/ref.fa
```

NOTE: The directory containing the reference fasta file must also contain the bwa index files or else it will build the indices. For more information see http://bio-bwa.sourceforge.net/bwa.shtml

Available arguments:
To see basic help information type:

```
$ run_deepseq.pl –h
```

To see more details type:

```
$ run_deepseq.pl -man
```

3) Dependencies

1. **External software dependencies**: DeepSNVMiner is a perl package and has been tested on MAC, CentOS, Redhat, and Ubuntu platforms using perl versions ranging from 5.10 to 5.14.

   In the simplest use-case DeepSNVMiner requires locally installed **bwa**, locally installed **samtools**, and a single **reference fasta** file (including **bwa index files**).

   Requirements:

   a. Samtools is available from http://samtools.sourceforge.net/ or http://www.htslib.org/ and once installed needs to be passed in using the parameter '-samtools /path/to/samtools' or loaded once using ./configure_deepseq.pl

   b. The reference fasta file used for the alignment needs to be passed in using the parameter '-ref_fasta /path/to/ref.fa' or loaded once using ./configure_deepseq.pl. NOTE: IF YOU HAVE PRE-COMPUTED

INDEX FILES (e.g. ref.sa or ref.fa.sa) ALREADY PUT THEM IN THIS DIRECTORY OR ELSE DEEPSEQ WILL BUILD THEM.

   c. BWA is available from http://sourceforge.net/projects/bio-bwa/files/ and once installed needs to be passed in using the parameter '-bwa /path/to/bwa' or loaded once using ./configure_deepseq.pl

2. **Genome Build**: Please note DeepSNVMiner is able to work with data from any genome build (currently tested for GRCh37 and GRCh38).

4) Input requirements

DeepSNVMiner requires three input files and a unique name used for the working directory and for file generation.

1. **-read1_fastq** *fastq_read1_file* -> REQUIRED:
   NOTE: currently only uncompressed files are allowed

2. **-read2_fastq** *fastq_read2_file* -> REQUIRED:
   NOTE: currently only uncompressed files are allowed

3. **-coord_bed** *bed_file* -> REQUIRED:
   Bed file (https://genome.ucsc.edu/FAQ/FAQformat.html#format1) containing all target regions to search for variants

4. **-filename_stub** *name_for_analysis* -> REQUIRED:
   User defined filename for the analysis; used to create the working directory and used when naming files generated during analysis

5. **-working_dir** *working_directory*:
   Path to local working directory for an analysis already underway. Not required for new analyses.

6. **-ref_fasta** *reference_genome_fasta_file***:**
   Single fasta file for the reference genome used for alignment, can be set one time with '-config' flag

7. **-samtools** *samtools_bin***:**
   Path to bwa binary, can be set one time with '-config' flag

8. **-bwa** *bwa_bin*:
   Path to bwa binary, can be set one time with '-config' flag

9. **-uid_len1** *length_of_UID_sequence_from_read_start* (default=10):
   Length of the barcode section from the start of the reads

10. **-uid_len2** l*ength_of_UID_sequence_from_read_end* (default=0):
    Length of the barcode section from the end of the reads. By default the sequence at the end of the reads do not contribute to the barcode. If utlized will be concatenated to the sequence defined by '-uid_len1'

11. **-no_uid** *no_UID_barcode_in_sequence* (default=UID present)
    Used when there is no UID barcode present in the sequence. To approximate a barcode the actual sequence from the genome will be utilized (default=first 10 bases). In this instance the 'barcode' is not removed from the FASTQ file compared to the default behavior where the barcode sequence portion is removed.

12. **-no_adaptor** *no_adaptor_in_sequence* (default=search_for_adaptor):
    Do not search for adaptor sequence in the reads for removal. DeepSNVMiner currently is only capable of detecting adaptors at the start of the reads and samples the first 10000 reads searching for repeating common substrings at the start. As defaults all sequences >= 10bp found in >=5% of all reads are removed. If this filtering is insufficient please use an external tool such as cutadapt (https://code.google.com/p/cutadapt/) prior to running DeepSNVMiner.

13. **-start_command** *resume_analysis_from_analysis_blocks* (default=new analysis):
    Resume an existing analysis from 1 of 6 starting points; check_fastq, pool_reads, bwa, call_variants, report, or graph (requires R).

14. **-graph** *generate_graphs* (default=OFF):
    Generate graphs of all variants detected, all supermutants detected, and the size distribution of read groups. This option requires 'R' and utilizes Rscript to generate a two pdf files for each genomic region in the bed file.

15. **-threads** *threads_for_bwa* (default=1):
    Use multi-threading for bwa. Ignored for all other analysis steps

16. **-min_seqlen** *minimum_remaining_length_required_for_inclusion* (default=no_minimum):
    Minimum length of remaining sequence after removal of barcode and adaptor sequence from raw FASTQ files

17. **-sm_count** *minimum_read_group_size* (default=5):
    Minimum number of reads sharing a common barcode to be considered for variant detection of supermutants.

18. **-sm_portion** *minimum_group_variant_fraction* (default=0.4):
    Minimum fraction of reads sharing a common barcode that need to be variant at the same base to be considered a supermutant.

19. **-min_group** *minimum_number_of_supermutants_required_for_supergroup* (default=1):
    Number of supermutants (>=5 reads with common barcode with 40% containing variant) required to be counted as a supergroup.

20. **-conf_file** *pass in different conf file* (default=deepseq.conf)

5) Running DeepSNVMiner

1. **Cutoffs:** Several important cutoffs are utilized:
   a. Supermutant: To be classified as a supermutant there must be at least 5 reads sharing the same barcode of which 40% or more share the same mutation (currently SNV or small indel). The minimum number of reads can be modifed with '-sm_count' and the portion of reads containing the variant can be modified with '-sm_portion'

   b. Supergroup: To be classified as a supergroup the same variant must be detected by only one supermutant. To increase this modify this cutoff the '-min_group' can be utilized

2. **Example usage:**
   a. To run with default parameters:
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed
      ```
   b. Generate graphs:
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -graph
      ```
   c. Don't search for adaptors:
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -no_adaptor
      ```
   d. Set UID to be first and last 6 bases of reads:
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -uid_len1 6 -uid_len2 6
      ```
   e. Set paths initially:
      ```
      $ configure_deepseq.pl
      ```
   f. Run on data with no uids
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -no_uid
      ```
   g. Restart existing analysis from bwa
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -working_dir test_deep_123456
      -start_command bwa
      ```
   h. Pass in bwa, samtools, and reference fasta (not needed when configure_deeps.pl has been run)
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -samtools /path/samtools -bwa
      /path/bwa -ref_fasta /path/ref.fa
      ```
   i. Define supermutant as needed 20 reads with 75% sharing a variant
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -sm_count 20 -sm_portion 0.75
      ```
   j. Require supergroup to contain 5 supermutants
      ```
      $ run_deepseq.pl -read1_fastq read1 -read2_fastq read2 -filename_stub
      test_deepseq -coord_bed regions.bed -min_group 5
      ```

6) Output format

There are two default output files, one for supermutants (filename_stub.pass_single_supermutants.tsv) and one for supergroups (filename_stub.pass_group_supermutants.tsv).
The file contains a summary of all 'supermutant' snvs or indels detected in the sequence. Output files are designed for importing into excel or libreoffice using tab as a delimiter.

1. Supermutant Output Columns

A. **chr**: chromosome

B. **start**: start_genomic_coordinate

C. **end** : end_genomic_coordinate

D. **variant_base:** variant base for snv, inserion, or deletion
   a. If SNV → will be single non reference base-by-base
   b. If Deletion → will be '-N' where N is the length of the deletion
   c. If Insertion → will be '+[AGTC]' where [ACTG] is the inserted base(s)

E. **barcode**: Barcode sequence

F. **variant read count**: Number of reads in the group containing the variant

G. **group count**: Number of reads in the group sharing the same barcode

H. **percent variant reads**: Percent of reads in the group containing the variant (F/G*100)

2. Supergroup Output Columns

A. **supermutant number:** number of supermutants contained in the supergroup

B. **chr:** chromosome

C. **start**: start_genomic_coordinate

D. **end** : end_genomic_coordinate

E. **variant_base:** variant base for snv, inserion, or deletion
   a. If SNV → will be single non reference base-by-base
   b. If Deletion → will be '-N' where N is the length of the deletion
   c. If insertion → will be '+[AGTC]' where [ACTG] is the inserted base(s)

F. **supermutants:** summary of supermutants in supergroup. Format for each supermutant is 'barcode (variant read count / total reads in group)' with multiple supermutants separated by commas.

7) Contact

Contact: matt.field@anu.edu.au

8) Citation

Citation: Please cite Field et al. BMC Bioinformatics….. when publishing results DeepSNVMiner

9) Acknowledgements

We thank the National Computational Infrastructure (Australia) for access to significant computation resources and technical expertise. This work supported by NHMRC Australia Fellowship 585490, National Institutes of Health [grant number U19 AI100627], and Bioplatforms Australia.