VarBen Manual

Ziyang Li (lzy@bioinfo.org)

February 3, 2019

1. Introduction

VarBen is a variant simulator to generate reference datasets for the testing and optimization of somatic variant calling pipelines.

VarBen can add a comprehensive set of variants, including SNVs, Indels, complex deletion-insertion variants, and large SVs, at any variant allele frequencies (VAFs) into any alignment reads stored in BAM format, including targeted and WGS data. As for SV editing, we can use VarBen to add variations like large deletion, inversion, duplication, translocation (include balanced translocation, insertional translocation, and whole-arm translocation) in aligned BAM files. In addition, VarBen can also simulate variants for the Ion Torrent platform by handling the flow signals in Ion Torrent sequencing reads.

2. Known bugs and limitations

VarBen is under rapid development driven by suggesting and bug reports from the mutation calling community.

1. Currently, we are working on testing a new version of mutation editor for CNV simulation.
2. SV simulation for Ion torrent data is still under development.

3. Software dependencies

The following softwares are required for VarBen:

1) pysam (python package >=0.9.4)

   python2 -m pip install pysam

2) numpy

```
python2 -m pip install numpy
```

3) samtools (http://samtools.sourceforge.net/)

```
git clone https://github.com/samtools/htslib.git
make -C htslib
git clone https://github.com/samtools/samtools.git
make -C samtools
cp samtools/samtools $HOME/bin
git clone https://github.com/samtools/bcftools.git
make -C bcftools
cp bcftools/bcftools $HOME/bin
```

4) bwa (http://bio-bwa.sourceforge.net/)

```
git clone https://github.com/lh3/bwa.git
make -C bwa
cp bwa/bwa $HOME/bin
```

4. Variant simulation by VarBen

VarBen provide a script called MutEditor (muteditor.py) to add SNV and Indel to BAM files, another script called SVEditor (sveditor.py) to add CNV and SV to BAM files. VarBen requires three mandatory inputs: 1) a BAM file containing aligned sequencing reads; 2) a mutation list file containing user-supplied variants (specified by the genome coordinates, variant types, nonreference DNA sequence, and input VAF); 3) an indexed reference genome (samtools faidx and aligner index). The reference genome should be consisted with the reference genome used to align the reads in the BAM file.

Function 1. Spike SNV & Indel into aligned BAM file (muteditor.py)

```
usage: muteditor.py [-h] -m MUTFILE -b BAMFILE -r REFFASTA -o OUTDIR
                    --alignerIndex ALIGNERINDEX [-p PROCESS]
                    [--seqer SEQER] [-g] [--aligner ALIGNER]
                    [--haplosize HAPLOSIZE] [--mindepth MINDEPTH]
                    [--minmutreads MINMUTREADS] [--snpfrac SNPFRAC]
                    [--minmapq MINMAPQ] [--multmapfilter]
                    [--diffcover DIFFCOVER] [--floworder FLOWORDER]
                    [--libkey LIBKEY] [--barcode BARCODE] [--tag]
```

Parameters description

-m/--mutfile MUTFILE

The input mutation list for muteditor.py is a six column format including Chromosome, Start, End, Allele Frequency, Type, and Alternative Sequence. The genomic coordinates are 1-based. There are four types of snv/indel included in the software: snv (single nucleotide variant), ins(insertion), del(deletion), Sub (Complex variant). The file format is shown as below.

#Chrom	Start	End	Allele Frequency	Type	Alternative Sequence
chr7	55259515	55259515	0.6	snv	G
chr7	55249010	55249011	0.25	ins	GTT
chr7	55242465	55242473	0.3	del	.
chr17	37880997	37880997	0.5	Sub	TTAT

The variant types of snv, ins, del, and Sub represent single nucleotide variant, small insertion, small deletion and complex variant, respectively.

(1) For SNV, the first line of the above mutation list will guide VarBen to add a single nucleotide variant of “T>G” at chromosome 7 position 55259515 with 60% VAF. This variant is usually described as NM_005228.5(EGFR):c.2573T>G (p.Leu858Arg).

(2) For small insertion, the second line will guide VarBen to add an insertion of “GTT” between the 55249010 and 55249011 position of chromosome 7 with 25% VAF. This variant is usually described as NM_005228.5(EGFR):c.2308_2309insGTT (p.Asp770delinsGlyTyr).

(3) For small deletion, the third line will introduce a 9 bp deletion (GGAATTAAG) from the 55242465 to 55242473 position of chromosome 7. This variant is usually described as NM_005228.5(EGFR):c.2235_2243del (p.Leu747_Glu749del).

(4) For complex deletion-insertion variant, the fourth line will guide VarBen to delete “G” and then insert “TTAT” at the 37880997 position of chromosome 17 with 50% VAF. This variant is usually described as NM_004448.3(ERBB2):c.2326delinsTTAT (p.Gly776delinsLeuCys).

How to determine the start and end position?

- For single nucleotide variant, the start and end site should be the same position in the genome.
- For short sequence insertion, the alternative sequence is inserted between the start and end site.
- For short sequence deletion, the sequence from the start to the end (columns two and three) is deleted from the sequencing reads.
- For Complex mutation, we want to using a short **sequence A** to replace a **sequence B**, we should put the **sequence B**'s start and end position in our mutation file.

-b/--bamfile BAMFILE

The input BAM files should be aligned to reference genome, sorted in coordinate order and indexed with samtools index. By default, the software considers the BAM file consists by entirely paired-end reads, if a user needs to spike mutation in a BAM file which consists by single-end reads, they need using the -single option.

-r, --reffasta REFFASTA

The reference genome in FASTA format and indexed with samtools (samtools faidx). The target BAM file should be generated by the same reference file used in this option, especially the chromosome names and lengths in the reference FASTA must be the same as in the BAM header.

--alignerIndex ALIGNERINDEX

The indexed reference genome in the FASTA format of aligner. For example, if the aligner is BWA, then BWA index should be provided. This FASTA file is called by the external aligner.

-o, --outdir OUTDIR

A output directory name for edited bam file and other information.

-p, --process PROCESS

Parallel mode: process number (default = 1)

--seqer SEQER

Define the sequence platform for reads generating: illumina, life, BGI. Default is illumina, life stand for the Ion Torrent platform.

-g, --single

To declare that the input bam is single-ended (default is False)

--aligner ALIGNER

Choose an aligner to remap edited reads to reference genome (default is bwa, others include novoalign and tmap). Currently, BWA mem, TMAP, and NovoAlign are supported, although it should be straightforward enough to add further aligners through modification of the remap paired function.

--haplosize HAPLOSIZE

The size of haplotype block to consider when adding more than 1 proximal mutation. (default = 0)

For example, if two SNVs are spiked in 5bp apart and -haplosize is 5 or greater, the two SNVs will be on the same haplotype (i.e. share the same reads for reads covering both positions).

--mindepth MINDEPTH

The minimum depth of reads position which could be edited to simulate mutation. (default = 30)

For instance, if reads depth of one position is 30× (only 30 reads covered this position) and --mindepth is 30 or greater, VarBen will drop this site automatically due to the read depth is not enough to add mutation.

--minmutreads MINMUTREADS

The minimum number of reads to be edited in one position (default = 5). VarBen will calculate the number of mutated reads by the allele frequency and the total number of reads in the position. If the mutation reads number is less than 5 and --minmutreads is 5 or greater, VarBen will drop this site automatically.

--snpfrac SNPFRAC

To avoid spike any mutatoin on top of existing heterozygous alleles, the heterozygous allele fraction set to 0.1 (default = 1)

--minmapq MINMAPQ

Reads mapping quality less than MINMAPQ will not be considered to edit (default 20).

--multmapfilter

Any multi-mapped reads will not be considered to edit (default is True).

--diffcover DIFFCOVER

The coverage difference allowed between the input BAM and output BAM (default 0.9).

--floworder FLOWORDER

If the sequence platform is Ion torrent (--seqer is life), sequencing flower order should be provided.

--libkey LIBKEY

If the sequence platform is Ion torrent (--seqer is life), the library key sequence should be provided.

--barcode BARCODE

If the sequence platform is Ion torrent (--seqer is life), the library barcode sequence should be provided.

--tag

Add tag to edited reads (default False).

Function 2. Spike SVs into aligned BAM file (sveditor.py)

```
usage: sveditor.py [-h] -m SVFILE -b BAMFILE -r REFFASTA -l
                     READLENGTH -o OUTDIR --alignerIndex
                     ALIGNERINDEX [-p PROCESS] [--seqer SEQER] [-g]
                     [--aligner ALIGNER] [--mindepth MINDEPTH]
                     [--minmutreads MINMUTREADS]
                     [--minmapq MINMAPQ] [--multmapfilter]
                     [--floworder FLOWORDER] [--libkey LIBKEY]
                     [--barcode BARCODE] [--tag]
```

Parameters description

-m/--mutfile MUTFILE

The input mutation list format for **sveditor.py** depends on the SV types. There are six types of SVs currently supported for VarBen, including inv (large inversion), del (large deletion), dup (tandem duplication), trans_balance (balanced translocation), trans_unbalance (insertional translocation), and trans_chrom (whole arm translocation). VarBen will simulate exact breakpoint for each SV type. The file format is shown as below.

```
## Mutation list format for deletion & inversion
#Chr   Start       End         Type             AF    
chrX   12994966    12996009    del              0.6   
chr10  32311365    43609712    inv              0.5   
## Mutation list format for tandem duplication
#Chr   Start       End         Type             AF    dup_num
chr1   16076907    16086182    dup              0.6   3
chr1   23665443    23711586    dup              0.5   4
## Mutation list format for translocations
#CHR1  start       End         Type             AF    CHR2     start       End
chr2   29754284    29754947    trans_balance    0.5   chr2     42522695    42523089
chr10  43608984    43609308    trans_unbalance  0.5   chr6     117640981   117640982
chr1   154142762   154142762   trans_chrom      0.5   chr6     117642093   117642093
```

VarBen can simulate different types of SV with exact breakpoint. For tandem duplications, the duplicate times of the sequence should be specified. e.g. dup 0.6 3 specifies the region is duplicated three times with 60% frequency. On the basis of different classification methods, various translocation types are exits. Here, three types of translocations are supported by VarBen, including balanced translocation, insertional translocation, and whole arm translocation (Figure 1).

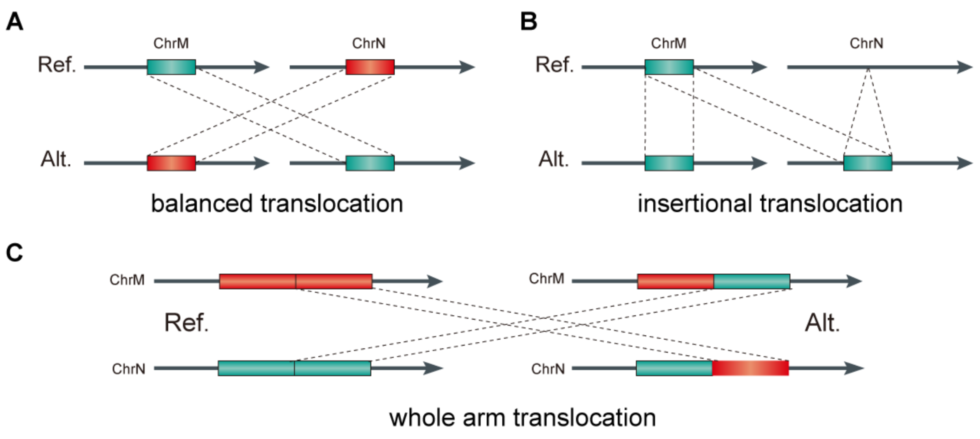

Figure 1. Translocation types supported by VarBen. (A) balanced translocation. (B) insertional translocation. (C) whole arm translocation.

Quick Start

Example of SNV & Indel simulating for Illumina platform

```
python2 /opt/VarBen/bin/muteditor.py -m mutFile.tsv \
-b Illumina_normal.bam -r ./reference/hg19/ucsc.hg19.fasta \
-p 4 --aligner bwa --alignerIndex ./reference/hg19/ucsc.hg19.fasta \
--seqer illumina --haplosize 10 --mindepth 100 --minmutreads 5 \
--snpfrac 0.1 -o Illumina_mut_out
```

Example of SNV & Indel simulating for Ion torrent platform

```
python2 /opt/VarBen/bin/muteditor.py -m mutFile.tsv \
-b IonXpress_001_realigned.bam -r ./reference/tmap/hg19/hg19.fasta \
-p 4 --aligner tmap --alignerIndex ./reference/tmap/hg19/hg19.fasta \
--seqer life -g --haplosize 10 --mindepth 100 --minmutreads 5 \
--snpfrac 0.1--libkey TCAG --barcode CTAAGGTAACGAT -o Ion_mut_out\
```

Example of SV simulating for Ion torrent platform

```
python2 /opt/VarBen/bin/sveditor.py -m svFile.tsv \
-b Illumina_normal.bam -r ./reference/hg19/ucsc.hg19.fasta \
-p 4 --aligner bwa --alignerIndex ./reference/hg19/ucsc.hg19.fasta \

--seqer illumina --mindepth 100 --minmutreads 5 -l 150 \

-o Illumina_sv_out
```