

# SV-Simulation Documentation

## Index

### Introduction

### Prerequisites and installation

1. System
2. ROOT package
3. SV-Simulation programs

### Executing the simulation

Program 'sim'

Alternative program

Moving on to *PEMer* workflow

### Input and output files

1. Input format
  - 1.1. SV file
  - 1.2. Read length file
  - 1.3. Chromosome data
2. Output format
  - 2.1. 'reads\_out1.fa'
  - 2.2. 'reads\_out\_pure.fa'
  - 2.3. 'ref\_genome.fa'
  - 2.4. 'sv\_genome.fa'
  - 2.5. 'svs\_out.txt'
  - 2.6. 'GenomeFile.root'
  - 2.7. Standard out stream

Additional output files by 'SimSV'

  - 2.8. 'GenomeFile.log'
  - 2.8. Graphs in jpg format

### Reference

## Introduction

*SV-Simulation* is a software developed to parameterize *PEMer* workflow and estimate its efficiency in reconstructing structural variants (SVs).<sup>[1]</sup> It randomly creates SVs into a reference genome sequence and generates paired-end reads with linker sequences as described in ref [1]. It can also introduce sequence errors into the reads based on a 454 error model.<sup>[1]</sup> The output file of simulated reads is in a compatible format with the input file for the *PEMer* workflow. Therefore, it is used to benchmark the *PEMer* and examine its performance as described in ref [1]. Furthermore, *SV-simulation* can also be readily applied to evaluate other SV constructing algorithms.

## Prerequisites and installation

### 1. Unix or Mac system

### 2. ROOT package

Free ROOT package can be obtained from <http://root.cern.ch/>

### 3. *SV-simulation* programs

The *SV-simulation* programs can be obtained from the *PEMer* package at <http://sv.gersteinlab.org/pemer>.

To install the simulation programs, type the following:

```
cd PEMer_Package/SV_simulation/sv_sim
make
```

If the compilation fails the most likely reason is that you need to change compiler flags. Please type in the following to get the compiler flags specific to your platform.

```
root-config --cflags
```

Please insert the output to the line defining ROOTFLAGS variable in the file 'Makefile'. Type in 'make clean' and then 'make'.

An example of a make file for MAC OS X 10.4 (Leopard) is given in file 'Makefile.mac'. You can run it by typing 'make -f Makefile.mac'

## Executing the simulation

After installation (See 'Prerequisites and installation'), the simulation can be carried out as follows. The user should first change directory before executing the simulation using the command line:

```
cd PEMer_Package/SV_simulation/
```

## Program 'sim'

### Description

This is the core program written in C++ that carries out the simulations. It outputs text files and generates figures in the ROOT browser (See 'Input and output files').

### Synopsis

```
./sim -svd SVFile -rd ReadLenthFile or value [Options] GenomeFile
```

#### SVFile

Input file with information of SV numbers and lengths. (See 'SV file' in 'Input and output files')

#### ReadLenthFile or value

Input file with information of read length distribution. (See 'Read length file' in 'Input and output files').

Alternatively, this argument can be a value, e.g. 36 for Solexa platform reads.

#### GenomeFile

Input file with genomic information. (See 'Chromosome data' in 'Input and output files')

### [Options]

#### -diploid

If this option is specified, the simulation is carried out for diploid genome. SVs generated are heterozygous. If this option is left out, the simulation is carried out for haploid genome.

#### -logmean Num

'Num' specifies the value of the mean of PE-spans in log space. Default is 7.8.

#### -logsd Num

'Num' specifies the value of standard deviation of PE-spans in log space. Default is 0.29.

#### -coverage Num

'Num' specifies the value of genomic coverage the user wants to simulate. Default is 7.

-454 OR -solexa

Specifying either one of them will generate simulated reads from 454 platform of Solexa platform. Default is 454.

An example of running the simulation on the sample dataset included in the package is as follows. The optional arguments are default in this example.

```
./sim -svd Sample_data_SV_simulation/my_svs.txt -rd
Sample_data_SV_simulation/lengths_of_reads_with_44bp_linker.txt
Sample_data_SV_simulation/chr21.fa
```

## Alternative program

### Description

Instead of executing the program 'sim', the user can choose to execute the program 'SimSV'. 'SimSV' has additional features that might not be necessary for all users. 'SimSV' is written in shell. Default shell is C shell. For other shells, please modify the first line of the 'SimSV' accordingly. 'SimSV' has two major differences from 'sim'.

a) All the output files go to a folder in the working directory created by 'SimSV' program. The folder is named after 'GenomeFile' (See below). For example, the 'GenomeFile' is called 'chr21.fa', the folder will be named as 'chr21.dir'.

b) 'SimSV' also generates figures in .jpg format.

The user should specify the same arguments described for 'sim' by modifying lines 15-20 in the 'SimSV' program.

The user should provide three files at line 35 of the 'SimSV' program. They are the dataset the user wants to compare with the simulated dataset, e.g. experimental data. The three files contain information about distributions of paired-end span, length of reads and length of ends respectively. They are all in the same format as 'Read length file' in 'Input and output files' section. The user should provide the same three files for lines 2-4 of the script 'make\_distr.cpp' and provide the .root file in the output directory for line1.

### Synopsis

```
./SimSV GenomeFile
```

#### GenomeFile

Input file with genomic information. (See 'Chromosome data' in 'Input and output files')

An example of running the simulation on the sample dataset included in the package is as follows. The options are default in this example.

```
./SimSV Sample_data_SV_simulation/chr21.fa
```

## Moving on to *PEMer* workflow

For 454 simulated reads, the output file 'reads\_out1.fa' and 'reads\_out\_pure.fa' are in a format compatible with the *PEMer* workflow input. (See also 'Output format') An example of running the output file 'reads\_out1.fa' through the first step of *PEMer* workflow using default options is as follows. (See also *PEMer\_workflow\_documentation.doc*)

```
./PairedEndPipelineSQ.py reads_out1.fa
```

To use Solexa simulated reads on *PEMer*, the output files 'reads\_out1.fa' and 'reads\_out2.fa' can be readily reformatted to become input files for the 2<sup>nd</sup> step in the bundle analysis - 'runMegablast.py'. (See *PEMer\_workflow\_documentation.doc* for more detail). Alternatively, they can be mapped with other algorithms such as MAQ, the output from which can be further analyzed by *PEMer*.

## Input and output files

### 1. Input format

#### 1.1. SV file

The program needs an input file which contains the information about the numbers and lengths of deletions, insertions and inversions generated in the genome. Each line of the file is formatted as 'SV NUM LEN', where 'SV' is 'deletion', 'insertion' or 'inversion', 'NUM' is the number of certain events of specific length, 'LEN' is the length of an event. An example of the file is as follows.

```
deletion 1 1000
deletion 1 5000
insertion 2 10000
inversion 10 500
```

#### 1.2. Read length file

The program needs an input file which contains the information about the distribution of the read length. The file should be made up with integers which indicate read lengths, each integer occupying one line. The simulated paired-end reads will have the same length distribution as this input file. An example of the first few lines of the read length file is as follows.

```
245
245
246
246
247
247
```

247  
247  
248

### 1.3. Chromosome data

The program needs an input file which is in FASTA format and contains the genomic sequence of a given chromosome that user wants to do simulation on. This genome should be the reference genome. After simulation, there will be deletions, insertions or inversions in the novel genome in reference to the reference genome. The definition line should be formatted as '>CHR:1-LAST', where 'CHR' is the chromosome number, 'LAST' is the position of the last nucleotide on the chromosome. An example is as follows:

```
>chr21:1-47883217
AATAATTTAACCTTTTAAATTTTAAAAATATGATTTTCCTGAACATTTAA
TAAGCTACCAAGAAAAAAGTATTTAAAAATAGAATTAATTCAAAATATT
TTCCAGGAGAATAGTATCTCGATAATATCAGGGTTGGAAAGACCATACTA
TATTTAGTTGCACAATAGAGGATTAGAAACATCTTTCTCTGTGTGCTCAC
```

## 2. Output format

The simulation produces the following output files in the working directory.

### 2.1. 'reads\_out1.fa' (If simulation is chose to be on 454 platform)

The file produces paired-end reads in FASTA format, which is the format of the input file for the *PEMer* workflow. <sup>[4]</sup> Errors are simulated for the reads. The definition line is formatted as '>NUM\_End1StartCoordinate\_End1EndCoordinate\_End2StartCoordinate\_End2EndCoordinate length=VAL uaccno=ID', where 'NUM' is the counting number of each read, 'End1StartCoordinate' is the start coordinate of the first end of the read pair, 'End1EndCoordinate' is the end coordinate of the first end of the read pair, 'End2StartCoordinate' is the start coordinate of the second end of the read pair, 'End2EndCoordinate' is the end coordinate of the second end of the read pair, 'VAL' is the length of each read, 'ID' is the identifier of each read. An example of the first few lines is as follows.

```
>1_20953785_20953830_20952122_20952275 length=244 uaccno=E1
TGCTAGCCAGACAGAATAAAAGTCTAGATATATGTCTAAAGGGACAGTTGGA
ACCGAAAGGGTTTGAATTCAAACCTTTTCGGTTCCAACGTGCATCACCTGAGGT
CAGGAGTTCCAGACCAGCCTGACCAATATGTTGAAACCTGTCTCTCCTAAATA
TAAAAAGTTAGCTGGGTATGGTGGCACATGCCTGTAAATCCCGCTACTGGGGG
CAGGGGTATCACTTGAAACCCGGGAGGCGGAGA
>2_17102842_17103084_17100618_17100619 length=289 uaccno=E2
CCAAAATAGGCAACACAACATTCATATGGACAAAAGTTGACACGTGACCCCT
CCACCAGTACTTTTTCAACACAAAACGTTAAGTTCAAAGTAGGTCCTATATCTAA
ATGTACACACCAAAATCATAAAGGCTTATACAAGAAAATCATTAGATAATATCTC
CAAGCTCTTTAAGTCTATTAGATAGAAAGAAAATCTGTGGACCATGAAAACAACT
AATAAATATGATTCCCATCAAAAAATTTTGTGGAACCGAAAGGGTTTGTAATTC
AAACCTTTTCGG
TTCCGAACCT
>3_21627270_21627377_21625054_21625173 length=272 uaccno=E3
```

```

CTTGGAGTGATTTTTTAGACGTTGACATTCATACGACATGGATTGTGAATAGC
AATAGGAGCATTCTACCTGACTTTATTGGGAGGGAGTAATAAGGTATTTATTGG
GTTGGAACCGAAAGGGTTTGAATCAAACCCTTTCGGTTCCAACAAGTGAACATA
AGTTACGTCACTAGTCTTCAAAGTTTACTAATATTTTCGTAATATCTAAAT
GTTTTTAATCCCATCAGAGTATTTCTCATAATTTTAGTTTTTCTTCTCTGAGAGTTT

```

Alternatively, the corresponding output files here are 'reads\_out1.fa' and 'reads\_out2.fa', given that simulation is chose to be on Solexa platform. Errors are simulated for the reads. Since simulated paired-end solexa reads do not contain linker, the corresponding two end sequences are stored in these two files respectively. An example is as follows:

'reads\_out1.fa':

```

>1_1_41847813_41847848 length=36 uaccno=E1
TAAGGTCTTCAGTTTCCCCTAATTTCTTTGAGCTCT
>2_1_30956479_30956514 length=36 uaccno=E2
TCCTACAGTTCCTAGCCCTGAGTGCTTAATCATAGT
>3_1_45555146_45555181 length=36 uaccno=E3
AGTCAGCTCCCTGGCCCTTGGACTTAGGAAGCCTCA

```

'reads\_out2.fa':

```

>1_2_41850465_41850500 length=36 uaccno=E1
ATCCAGACCCAAAACGCCCTCAGTTCCCTTACCTGA
>2_2_30959663_30959698 length=36 uaccno=E2
TTCCAGCTTCACACAATAACAATTCAGGCATGTTGCA
>3_2_45557627_45557662 length=36 uaccno=E3
TCAACACCCACACCGGAGCGGCCCATTTATTACAAG

```

## 2.2. 'reads\_out\_pure.fa'

For 454 reads, same as 'reads\_out1.fa' except that sequencing errors are not simulated for the reads in this file.

For Solexa reads, sequences from two ends without sequencing errors are contained in this single file. An example from this output is as follows:

```

>1_1_41847813_41847848 length=36 uaccno=E1
TAAGGTCTTCAGTTTCCCCTAATTTCTTTGAGCTCT
>1_2_41850465_41850500 length=36 uaccno=E1
ATCCAGACCCAAAACGCCCTCAGTTCCCTTACCTGA
>2_1_30956479_30956514 length=36 uaccno=E2
TCCTACAGTTCCTAGCCCTGAGTGCTTAATCATAGT
>2_2_30959663_30959698 length=36 uaccno=E2
TTCCAGCTTCACACAATAACAATGCAGGCATGTTGCA
>3_1_45555146_45555181 length=36 uaccno=E3
AGTCAGCTCCCTGGCCCTTGGACTTAGGAAGCCTCA
>3_2_45557627_45557662 length=36 uaccno=E3
TCAACACCCACACCGGAGCGGCCCATTTATTACAAG

```

## 2.3. 'ref\_genome.fa'

The reference genome the simulation is carried out against. It should be the same as input file 'Chromosome data'.

#### 2.4. 'sv\_genome.fa'

The novel genome the simulation has generated. The simulation created deletions, insertions and/or inversions in the novel genome in reference to the reference genome. It is in the same format as the 'ref\_genome.fa'.

#### 2.5. 'svs\_out.txt'

The file contains information about the start and end coordinates of the SVs created by the simulation. Each line of the file is formatted as 'SV START END', where 'SV' is 'deletion', 'insertion' or 'inversion', 'START' and 'END' are the start and end coordinates in the reference genome respectively. An example of the first few lines is as follows.

```
deletion 46931701 46932700
deletion 13264802 13269801
insertion 44465715 44465715
insertion 22766759 22766759
inversion 44947859 44948358
inversion 34940662 34941161
```

#### 2.6. 'GenomeFile.root'

The file is named after the 'GenomeFile' file. For example, if the 'GenomeFile' file is called 'chr21.fa', the above output file is named as 'chr21.root'. The file can be opened by ROOT so that a few output graphs can be visualized. Type in the following command lines.

```
root
new TBrowser
```

An interactive window will be available. Go to the menu bar 'File ->Open' and browse the 'GenomeFile.root' file. Then go to 'ROOT Files' in the left column. You will find the 'GenomeFile.root' file which contains the graphs that help visualizing the simulation results.

'h\_end\_len': Length distribution of ends.

'h\_frg\_len': Length distribution of PE-spans.

'h\_frg\_start': Start positions of PE-spans in the reference genome.

'h\_frg\_end': End positions of PE-spans in the reference genome.

'h\_read\_len': Length distribution of reads.

'h\_sv\_len': Length distribution of structural variants (SVs).

'h\_sv\_start': Start positions of SVs in the reference genome.

'h\_sv\_end': End positions of SVs in the reference genome.

'h\_sig\_null': Signal distribution for homopolymers of size 0 by Roche 454 sequencing technology.

'h\_sig\_\*': Signal distribution for homopolymers of size \* by Roche 454 sequencing technology. \* denotes 1-9.



## 2.7. Standard out stream

Text printed to the standard out stream contains information about the options set by the user in the 'sim' program. An example of the file is as follows. In line 6 of the example, it gives 'WARNING' because there exist 'N's which denote any nucleotides in the reference genome and the 'N's are skipped. The program chunks genomes into 1000bp pieces in order to shorten the time of execution. Therefore, in lines 9 and 15 of the example, it gives the number of pieces each genome is chunked into. The lines 17 to 27 indicate the cutoffs for homopolymer signals for Roche 454 sequencing technology. For instance, if a homopolymer gives a signal between 2.46378 and 3.47849, it will be read out as a homopolymer of length 3. The discrepancy between the numbers in line 28 and line 29 is due to gaps denoted as 'N' in the reference genome. If an end pair spans a region containing 'N', the read is not printed. Lines 32-42 indicate the number of errors introduced by the simulation. Of these, each line is formatted as 'LEN:ALL INS DEL RATIO', where 'LEN' is the actual length of homopolymer in the genome, 'ALL' is the total number of nucleotides in sequences of corresponding length of homopolymer, 'INS' is the number of nucleotides that are errors inserted into corresponding length of homopolymer, 'DEL' is the number of nucleotides that are errors deleted from corresponding length of homopolymer and 'RATIO' is calculated as  $(\text{'INS'} + \text{'DEL'}) / \text{'ALL'}$ .

```

1 Intend to generate 7x coverage with DNA fragments of length 2440.
2
3 Parameters of lognormal distribution for DNA fragments are:
4 logmean = 7.8
5 logsd    = 0.29
6 WARNING: Sequence is shorter than coordinates provided.
7 Reference genome is chr21:1-46944324
8 It is of 46944324 nucleotides in length.
9 Chunked in 46945 pieces.
10
11 Using TRandom3 random generator.
12
13 SV genome is chr21:1-43844324
14 It is of 43844324 nucleotides in length.
15 Chunked in 43845 pieces.
16
17 Cut offs for 0 are: 0-0.559619
18 Cut offs for 1 are: 0.559619-1.42831
19 Cut offs for 2 are: 1.42831-2.46378
20 Cut offs for 3 are: 2.46378-3.47849
21 Cut offs for 4 are: 3.47849-4.48654
22 Cut offs for 5 are: 4.48654-5.49161
23 .....
24 Cut offs for 96 are: 95.5132-96.5132
25 Cut offs for 97 are: 96.5132-97.5132
26 Cut offs for 98 are: 97.5132-98.5132
27 Cut offs for 99 are: 98.5132-100
28 Generated 120654 fragments for monoploid genome.
29 Printed reads for 85437 fragments.
```

```

30   Average fragment length is 2543.72 nucleotides.
31
32   0: 0 133624 0 inf
33   1: 10253067 60036 49560 0.0106891
34   2: 6790212 91176 29450 0.0177647
35   3: 4068321 70814 46618 0.028865
36   4: 1140956 21636 17715 0.0344895
37   5: 483085 9514 8505 0.0372999
38   6: 162510 3132 3029 0.0379115
39   7: 71981 1404 1370 0.0385379
40   8: 29184 553 534 0.0372464
41   9: 18792 326 326 0.0346956
42  10: 13480 256 235 0.0364243
43   Total: 23108384 393634 158545 0.0238952

```

### Additional output files by 'SimSV'

If the user chooses to run 'SimSV' program instead of 'sim', there are several output files in addition to the outputs by 'sim'. (See 'Executing the Simulation' for details how to run 'SimSV')

#### 2.8. 'GenomeFile.log'

The file is named after the 'GenomeFile' file. For example, if the 'GenomeFile' file is called 'chr21.fa', the above output file is named as 'chr21.log'. The file saves what is printed to the standard out stream by 'sim' program. (See '7) Standard out stream' in 'Output format')

#### 2.9. Graphs in jpg format.

'ends\_len.jpg': Length distribution of ends. Simulation result is in green, the user specified dataset is in blue. (See 'Alternative program' in 'Executing the simulation')

'frgs\_len.jpg': Length distribution of PE-spans. Simulation result is in blue, the user specified dataset is in green. (See 'Alternative program' in 'Executing the simulation')

'frgs\_start.jpg': Start position of PE-spans in the reference genome.

'frgs\_end.jpg': End position of PE-spans in the reference genome.

'reads\_len.jpg': Length distribution of reads. Simulation result is in blue, the user specified dataset is in green. (See 'Alternative program' in 'Executing the simulation')

'h\_sv\_len': Length distribution of structural variants (SVs).

'h\_sv\_start': Start positions of SVs in the reference genome.

'h\_sv\_end': End positions of SVs in the reference genome.

'error\_model.jpg': Signal distributions for homopolymers by Roche 454 sequencing technology used in the error model of the simulation.

## Reference

1. Korbel J, Abyzov A, Mu XJ, Carriero N, Cayting P, Zhang Z, Snyder M, Gerstein M: **PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data.** *Genome Biology* 2009, **10**:R23.