CuReSim: a customized read simulator

This is the manual for CuReSim version 1.2
CuReSim (Customized Read Simulator) is a customized tool which generates synthetic New-Generation Sequencing reads, supporting read simulation for major letter-base sequencing platforms. CuReSim is developed in Java and is distributed as an executable jar file. Wrappers to integrate CuReSim in Galaxy are also available.

1 Overview

The main features of CuReSim include :
— genome (fasta) or read (fastq) as input file
— choice between several error distributions
— a particular attention has been paid to special cases for which several introduced errors in the same read can result in less number of errors as expected due to compensatory changes
— generation of diagrams to know exactly the simulated error model (R is required)
— **NEW** in version 1.2 : multi-fasta input is allowed
— **NEW** in version 1.2 : user can define the maximum number of Ns allowed per read

2 Installation and usage

CuReSim does not need installation step but require Java installed on your machine
(see `http://www.oracle.com/us/technologies/java/overview/index.html` for more details on Java). Wrappers to integrate CuReSim in Galaxy are also available in CuReSim_galaxy forlder.
To run CuReSim, use the following command line :

```
java -jar CuReSim.jar [options] -f <input_file> [options]
```

3 Methods

CuReSim runs in four steps.

Input file pre-processing

When the input file is a genome in FASTA format, the first step uniformly draws random genome positions to generate reads without errors from both strands. The read size is normally distributed with a mean and variance which is user-defined. When the input file is a MULTI-FASTA file, the number of reads per chromosome is proportionally computed depending on the chromosome length and CuReSim is then run on each chromosome. When the input file already contains reads in FASTQ format, the pre-processing step consist in checking format and computing some values such as number of bases and mean read length. At the end of this step, a set of reads without errors is available.

Indel generation

In the second step, CuReSim introduces insertions and deletions in reads without errors of step1. The insertion and deletion rates are independent and user-defined. Indels can be uniformly drawn along the reads ($option - ui$). They can also be uppermost introduced in homopolymers. In this case, an iterative algorithm mostly introduces indels in the longer homopolymers.

Substitution generation

Substitution rate is user-defined. Substitutions can be uniformly drawn ($option - us$) or follow an exponential distribution depending on read position, i.e. substitution probability increases at the end of the read.

Correction step

We paid particular attention to special cases for which several introduced errors in the same read can result in less number of errors as expected due to compensatory changes. For example, deletion of an inserted base is forbidden, as well as to substitute an insertion. However, in some particular cases, introducing a succession of n mutations can finally lead to a minimal edit distance different from the number of introduced mutations n. The edit distance between two strings is defined as the minimum number of edit operations needed to obtain one string from the other one, with the basic edit operations being insertion, deletion, or substitution of a single character. Here you have two simple examples of this case :

```
Example 1:
ATGC -> 1 deletion (T) -> AGC -> 1 substitution (G->T) -> ATC => number of operations = 2
minimal edit distance = 1 (deletion of G between ATGC and ATC)
Example 2:
AAATGAA -> 1 deletion (T) -> AAAGAA -> 1 insertion (G) -> AAAGGAA => number of operations = 2
minimal edit distance = 1 (substitution T->G between AAATGAA and AAAGGAA)
```

In theory, the number of introduced mutation during simulation should be equal to the minimal edit distance between the read without error and the read with mutations, with the same weight for each basic operation. We add a step to correct the number of introduced errors with those corresponding to minimal edit distance. In the case of read simulation, it is important to know exactly error model of simulation in order to estimate and compute correct values in further analysis. This step can be time consuming. It can be skipped with *option − skip*.

4 Parameters

Options are given with default values in brackets.

```
usage: java -jar CuReSim.jar [options] -f <input_file> [options]
```

-f	(*String*)	genome fasta file or reads fastq file. It is the only MANDATORY field
-o	(*String*)	name of output fastq file [output.fastq]
-n	(*integer*)	number of reads to generate [50000]
-m	(*integer*)	read mean size [200 bases]
-sd	(*float*)	standard deviation for read size [20.0]
-r	(*integer*)	number of random reads [0]
-d	(*float*)	deletion rate [0.01]
-i	(*float*)	insertion rate [0.005]
-s	(*float*)	substitution rate [0.005]
-ui		when option -ui is added uniform distribution is drawn for indels [homopolymers]
-us		when option -us is added uniform distribution is drawn for substitutions [exponential]
-q	(*character*)	quality encoding character in fastq file [5]
-N	(*integer*)	maximum number of Ns allowed per read [0]
-v		verbose mode generating diagrams. R software is required [false]
-skip		skip the correction step [false]
-h		print this help

5 Output files

Three types of output files are generated : output fastq file, log file and diagrams (optional).

Output fastq file

This file contains reads generated by CuReSim with substitutions and indels. The read name encodes several data on simulation process and is of the form :

```
@<#1>_<#2>_<#3>_<#4>_<#5>_<#6>_<#7>_<#8>

1: genome name
2: original position
3: strand (0=forward;1=reverse)
4: random read (0=non-random;1=random)
5: number of insertions
```

```
6: number of deletions
7: number of substitution
8: read number (unique within a genome)
```

Log file

log.txt file records program activities. It gives input parameters, real simulation values and the final repartition of reads depending on number of errors.

Diagrams (optional)

Diagrams are obtained with $option - v$. R software is required (see `http://www.r-project.org/` for more details on R). You can find all diagrams generated by CuReSim in *diagrams* directory. Figures 1-4 shows example of diagrams generated by CuReSim.

FIGURE 1 –

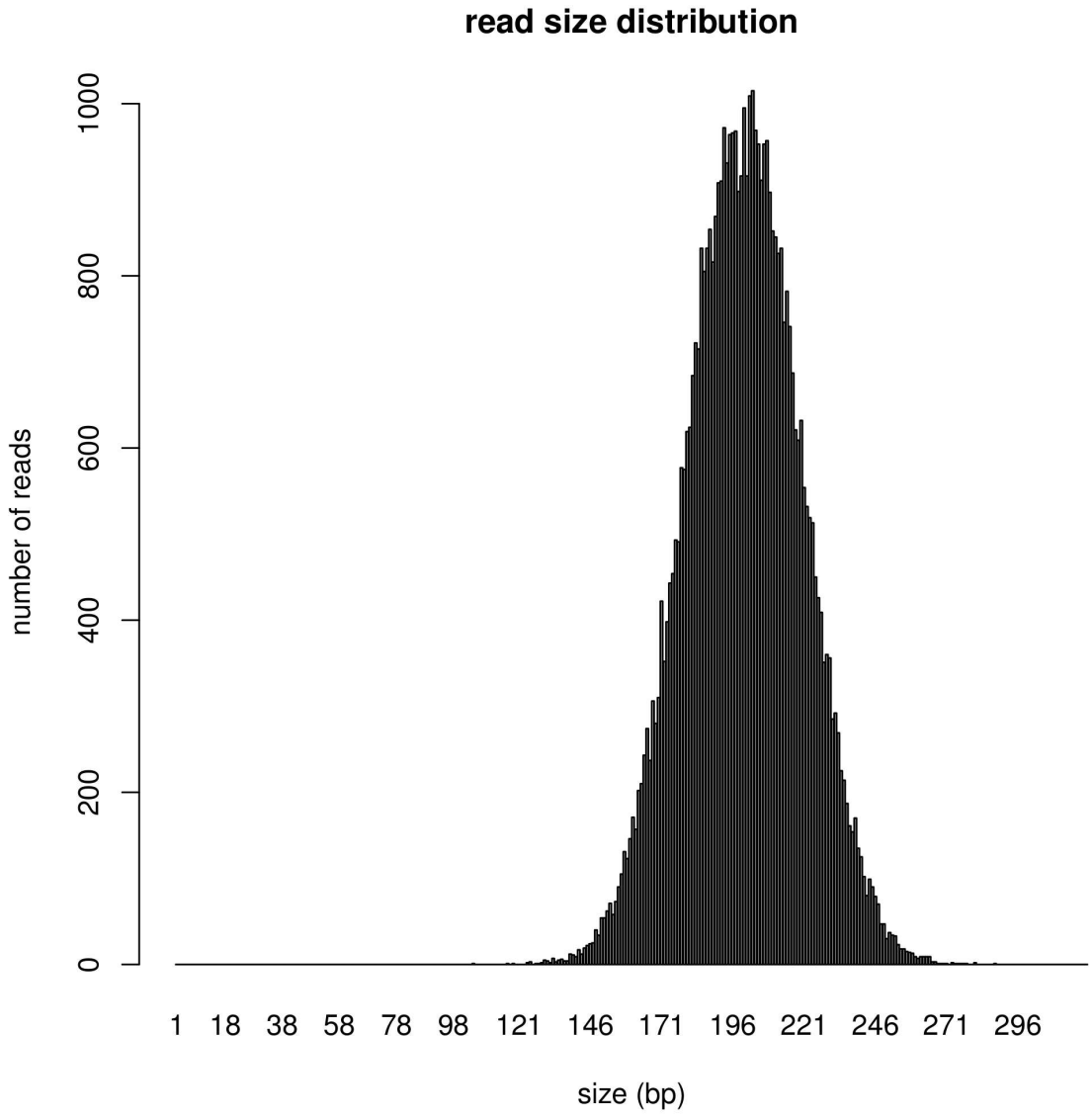


FIGURE 2 –

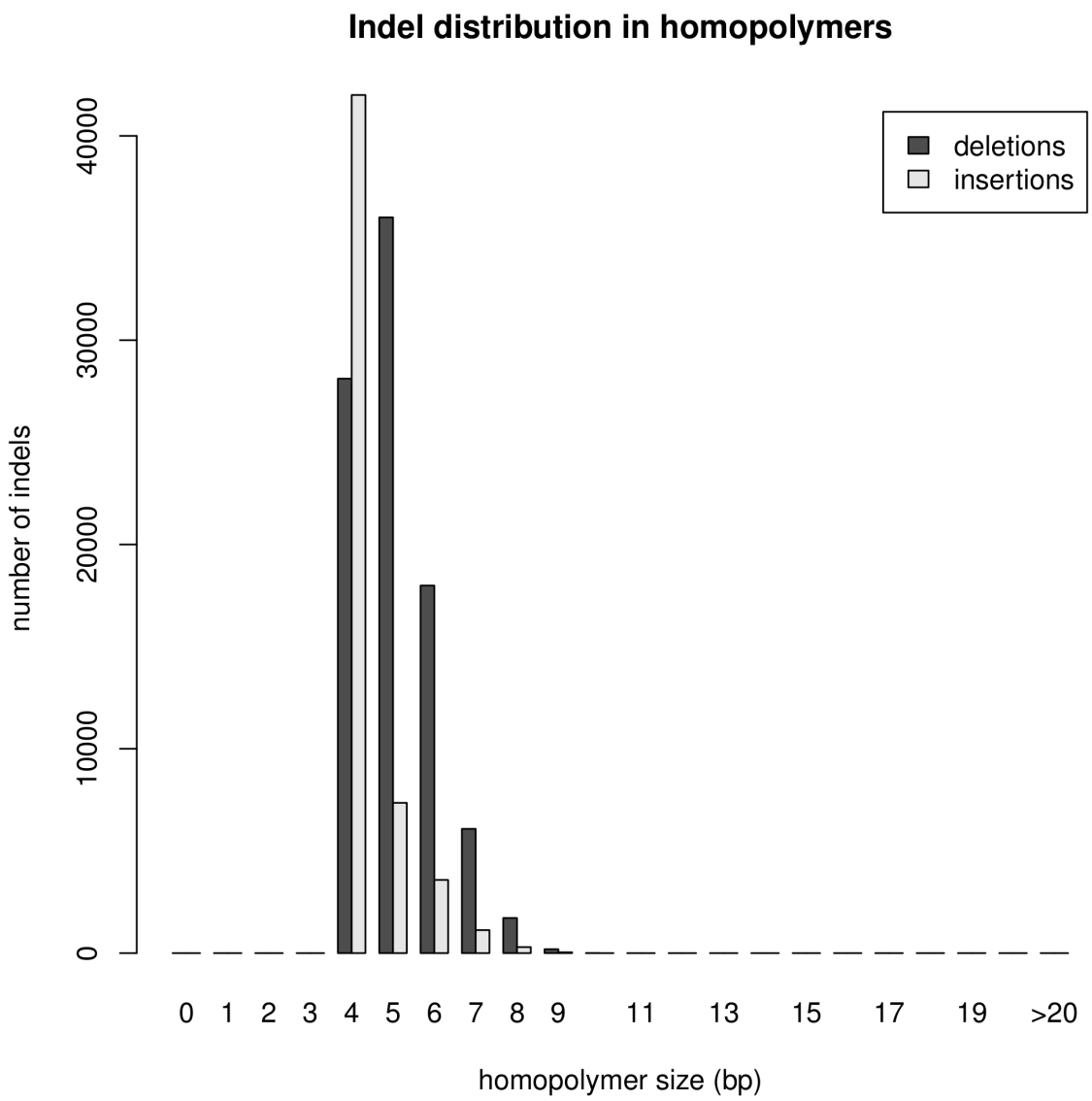

Indel distribution in homopolymers
deletions
insertions
number of indels
0
10000
20000
30000
40000
0 1 2 3 4 5 6 7 8 9 11 13 15 17 19 >20
homopolymer size (bp)

FIGURE 3 –

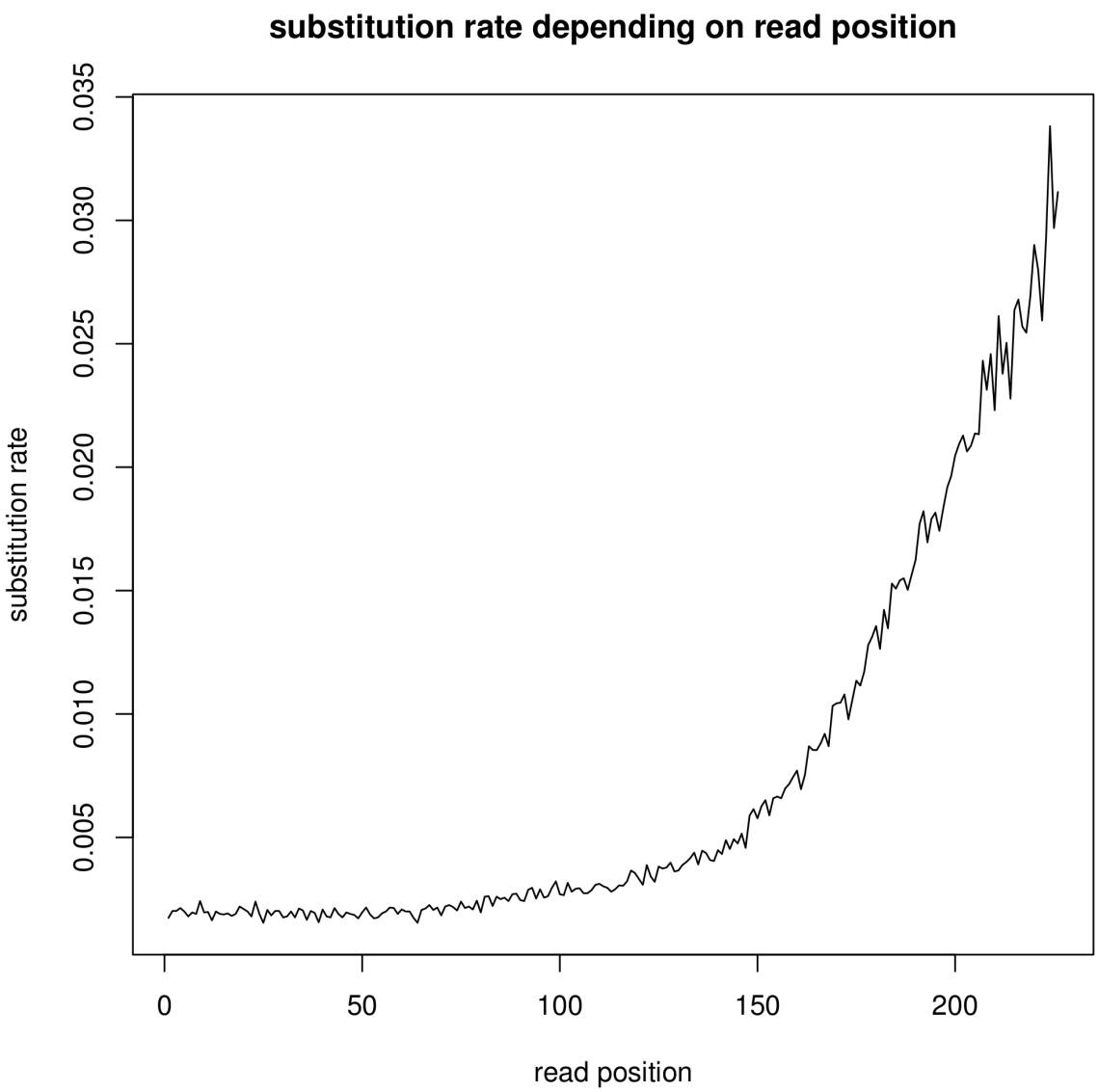
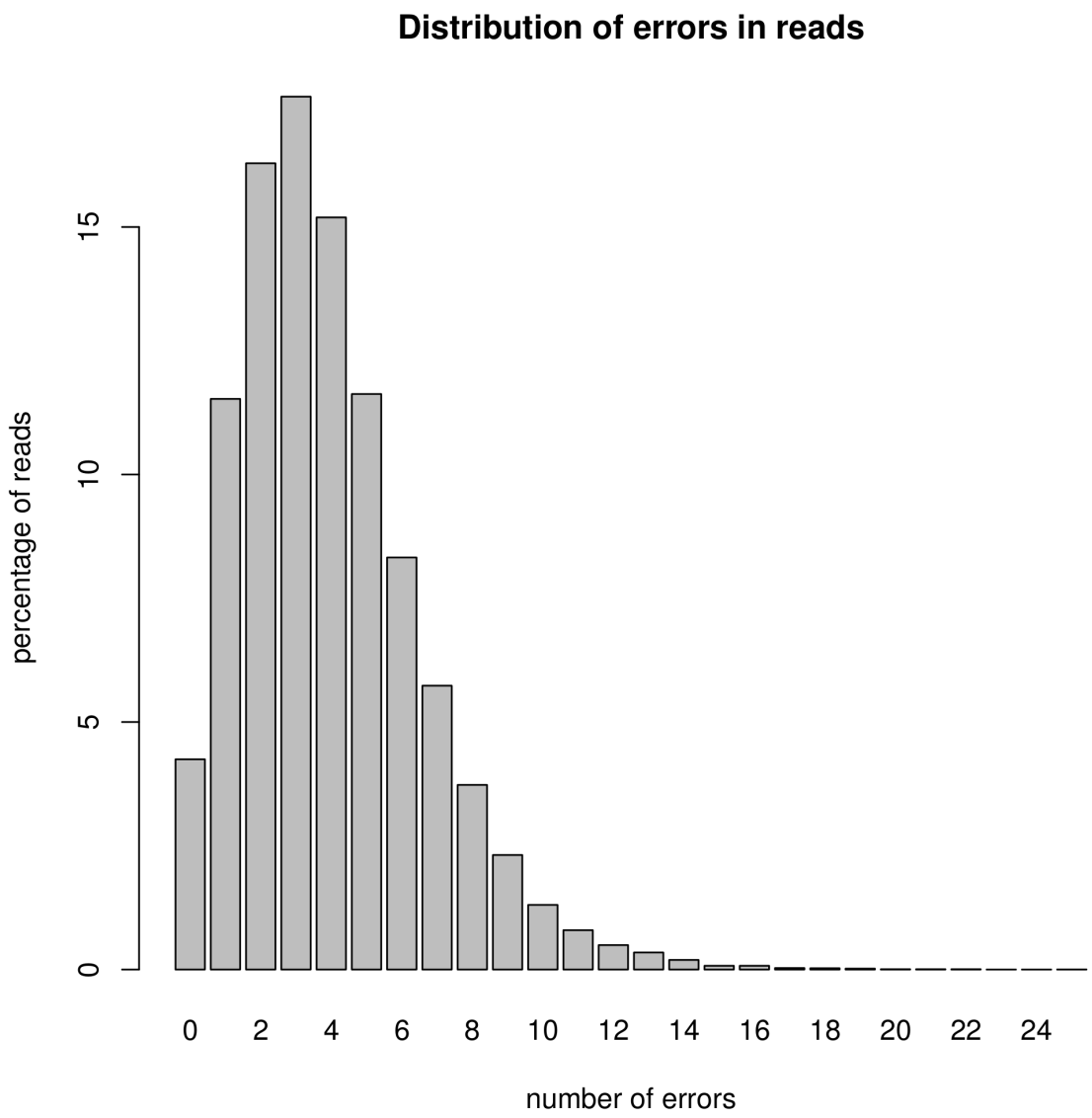

substitution rate depending on read position
substitution rate
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0
50
100
150
200
read position

FIGURE 4 –


Distribution of errors in reads
percentage of reads
0
5
10
15
0
2
4
6
8
10
12
14
16
18
20
22
24
number of errors