VarSimLab Documentation

Release 1.0

Abdelrahman Hosny

Feb 14, 2019

Contents

Advances in next generation sequencing have led to the creation of many tools for detecting variations in tumor sequences. To evaluate the performance of these tools, researchers need articifial tumor sequences with known variant positions. Currently, generating such simulated datasets is possible, it requires significant labor and technical know-how. The benchmarking process is especially difficult for tools that call variants from tumor whole exome sequences, an increasingly popular technique for tumor characterization

VarSimLab is a tool designed to dramatically simplify varaint-caller benchmarking. As of the current version (**nabavilab/varsimlab:0.2**), the simulator generates realistic artificial short reads, which harbor structural and copy number variations. VarSimLab can also simulates tumor heterozygosity (Ploidy and Subclones), and can generate whole genome, or whole exome datasets.

The output of VarSimLab is a directory containing tumor and normal reads, a log file containing simulation information, and benchmarking files, with the locations and positions of all variants generated in the run.

The output of VarSimLab ideal for benchmarking variant calling software, and showed high variant rediscovery rates for SNPs INDELS and CNVs when tested using VarScan variant calling software.

CHAPTER 1

Simulator User Guide

1.1 1. Installing Dependencies

Varsimlab can be called from the command line using any python 3 version

installing ART Varsimlab uses art_illumina to generate short reads with realistic sequencing errors. The documentation is available Here

installing SINC VarSimLab uses SInC simulator to generate biologically realistic tumor genomic variations. The source files and instructions on compiling are available Here

installing Bedtools (optional) If you'd like to use VarSimlabs exome sequencing capabilities, Varsimlab uses Bedtools is required. bedtools documentation is available Here

installing BWA (optional) If you'd like to use VarSimLab to automatically align the generated reads back t o your sequence of interest, generating SAM files, download and compile BWA. Download is available Here and Here is the documentation for the alignment algorithm is available Here

1.2 2. Prepare The Reference Genome

There are two ways to easily run VarSimLab for a reference genome

1. Download a pre-prepared folder to use from http://nabavilab.uconn.edu/datasets/varsimlab/ .
2. Prepare your own reference (super easy).

To prepare a reference genome, follow these steps:

1. Copy the reference genome in FASTA format (.FASTA | .FA) to the folder. You can also download genome from UCSC Genome Browser https://genome.ucsc.edu/cgi-bin/hgGateway
2. Copy target regions file in BED format (.BED) to the folder. You can also download the target files from UCSC Table Browser https://genome.ucsc.edu/cgi-bin/hgTables. You can also see more information about preparing the bed file at our github repository https://github.com/NabaviLab/VarSimLab

That's it! Your reference is ready to generate reads from.

1.3 3. Running VarSimLab

Here are the available arguments VarSimLab accepts at the command line required positional arguments:

filename name of output file genome genome to be processessed

VarSimLab also requires one of the following arguments:

-use_genome	generate tumor and normal for entire provided sequence. used for whole genome sequence simualtion
-bed	generate tumor and normal based on bed file containing exonic regions. used for whole exome sequence simulation

read generation parameters: arguments to adjust read generation

-c C	read depth of coverage
-s	use single end reads (default paired)
-l L	read length. default 100 bp
-m M	maximum distance for two bed ranges to be merged into one range. If zero, merges only those ranges that directly overlap with each other

error parameters: arguments to adjust tumor error generation

-cnv	percent of total input to be incorporated into a CNV. Values from 0 to 100. 4 would signify 4 percent of input should be included in CNVs
-cnv_min_size	minimum size of CNVs
-cnv_max_size	CNV_max_size
-snp	percent of total input to be turned into SNPs. Values from 0 to 100. A value of 5 indicates 5 percent of genome should be turned into SNPs
-indel	percent of total input to be included in INDELS. values from 0 to 100, a value of 1 indicates 1 percent of the genome should be included in indels

1.4 4. Understanding Simulator Results

There are two folders inside the *output_prefix* folder.

- **Normal:** it will contain *.FASTQ* file for reads that represent the control (or normal) sample. There will be two *.FASTQ* files if paired end reads were generated, and one if single end reads were generated.
- **Tumor:** it will contain *.FASTQ* file for reads that represent the tumor sample. There will be two *.FASTQ* files per allele for paired end sequencing, or one per allele for unpaired. In addition, it will contain the benchmark data that tells you where SNPs, Indels and CNVs for each allele in each subclone generated.

1.5 5. Understanding Benchmarking files

If a bed file was supplied, two sets of positions are calculated, one relative to the genome, the other relative to the exome. The genome position is likely to be much greater than the exome position, since the exome is much smaller than the genome, and exons are usually surrounded by large noncoding stretches.

CHAPTER 2

Simulator Methods

The full source code of the project can be found on GitHub at https://github.com/NabaviLab/VarSimLab

2.1 1. Simulating SNPs, Indels and CNVs

This step uses SInC There are several advantages to this tool over similar error generators– its speed, its ability to simulate a wider variety of errors (SNPs, INDELS, and CNVs), and the biologically realistic nature of the errors it generates.

Pattnaik, Swetansu, et al. "SInC: an accurate and fast error-model based simulator for SNPs, Indels and CNVs coupled with a read generator for short-read sequence data." BMC bioinformatics 15.1 (2014): 40.

2.2 2. Simulating Tumors

Tumor Ploidy is implemented by re-running SInC to generate more than 2 alleles. Tumor subclone is done by iteratively going through all above simulation steps to generate different aberrant genomes for different subclones.

2.3 3. Generating Short Reads

This step utilizes ART articifial read generator This is a widely used read generator with the capacity to faithfully simulate the error profiles of current next generation sequencers

Huang, W., Li, L., Myers, J. R., & Marth, G. T. (2012). ART: a next-generation sequencing read simulator. Bioinformatics, 28(4), 593–594. http://doi.org/10.1093/bioinformatics/btr708

CHAPTER 3

FAQs

3.1 1. What is VarSimLab necessary?

There are a variety of tools available for variant simulation, and short read simulation. However, generating benchmark ready simulated datasets remains an arduous process, requiring the use of multiple tools in conjuction. This is especially true for exome seqnuencing. The typical workflow for generating whole exome simulated data is approximately as follows: 1) Exonic sequences have to be combined to use as input to an error simulator. 2) The resulting error containing exome can then be used as input to a short read simulator 3) the read simulator output is aligned back to the genome. 4) The error simulator's variant positions file is corrected to be relative to the genome, rather than relative to the exome.

Only then can the simulated data be used for benchmarking. The above pipeline, which requires at minimum 3 seperate tools, can be reduced into a single command with VarSimLab

3.2 2 I want to perform exonic sequencing of a human sequence. How should I generate the bed file of exonic positions

Fortunately USCS tablebrowser makes this easy. See the docs here for a walkthrough here

3.3 3 How can I simulate the exome of multiple chromosomes.

As of the current release, unfortunately, exome simulation must be run one chromosome at a time, with one bed file per chromosome. Whole genome sequencing can accommodate multiple chromosomes

3.4 4 I have a question not covered here, or I've found a bug

please email us at thomas.davis@uconn.edu or abdelrahman.hosny@ieee.org

CHAPTER 4

Need Help?

Help us improve the current version of VarSimLab by:

- Reporting bugs
- Suggesting improvements
- Contributing to the project

Please contact Abdelrahman at abdelrahman.hosny@ieee.org