SigProfilerSimulator

Sept 26, 2018

INTRODUCTION

The purpose of this document is to provide a guide for using the SigProfilerSimulator for simulating mutational signatures in cancer. This tool allows for realistic simulations of single point mutations, double point mutations, and insertions/deletions with the goal of providing a background model for statistical analysis. The simulations are performed in an unbiased fashion, relying on random chance as the main distribution and can be performed across the entire genome or limited to user-provided ranges. This tool currently supports the GRCh37, GRCh38, mm9, and mm10 assemblies, however, additional genomes may be installed (see Simulating Additional Genomes). In addition, this tool makes use of SigProfilerMatrixGenerator and SigProfilerPlotting.

PREREQUISITES

The framework is written in PYTHON, however, it also requires the following additional software with the given versions (or newer) and access to BASH:

PYTHON	version 3.4 or newer
FASTRAND	Python module: https://github.com/lemire/fastrand/blob/master/README.md
WGET	version 1.19
SigProfilerMatrixGenerator	current version from GitHub: https://github.com/AlexandrovLab/SigProfilerMatrixGenerator

While the code was developed for application on a local computer, access to a cluster with greater computational power may be required for simulating a large number of mutations.

QUICK START GUIDE

This section will guide you through the minimum steps required to begin simulating mutations:

1. Install SigProfilerMatrixGenerator following the README file given in the link above.

2. Place the SigProfilerSimulator repository within the same folder as the SigProfilerMatrixGenerator.

3. Run the install_simulator_ref.py script using python3.

4. A) IF WORKING WITH VCF/MAF/SIMPLE TEXT FILES:

    a) Create a folder with a unique name for your project (ex: BRCA, ncRCC, etc.) within the *vcf_files* folder. Under this project, create an *SNV* and *INDEL* folder.

b) Place your vcf/maf/simple text files into this newly created *project/[mut_type]/* folder. If you are dealing with vcf files, ensure that you have an individual file for each sample of interest. Separate INDELs from SNVS and place in the appropriate folder under your project directory.

B) IF WORKING WITH MUTATIONAL MATRICES:

a) Place the matrix for the given context under the *references/matrix/[project]/* folder. The file name must follow the given format:

"[project].[type][context].[region]"
BRCA.SBS96.exome

Types: SBS (single base substitutions), DBS94 (INDELs), and DBS78 (DINUCs)
Contexts: 96 (trinucleotides), 192 (TSB trinucleotides), 1536 (pentanucleotides), 3072 (TSB pentanucleotides), DINUC (dinucleotides), and INDEL
Regions: exome, region (specified by BED file), all (entire genome)

5. From a command prompt run the mutation_simulation.py script as follows:

   python3 mutation_simulation.py –g GRCh37 –p BRCA –c 96 INDEL –s 100 **

6. The script will begin generating any missing reference files or will instruct you if there are certain folders missing (First time users only).

7. Simulated samples are saved within the *output* folder. Log files are saved within the *log* folder.

**NOTE: See commands below for list of all available parameters.

COMMANDS

-g or --genome	-> required: Followed by the reference genome (ex: GRCh37, GRCh38, mm10).
-p or --project	-> required: Followed by a unique project name (ex: BRCA).
-c or --context	-> required: Followed by the desired contexts to simulate. Only one type of single point mutations may be provided (96, 192, 1536, or 3072) along with DINUCs or INDELs (ex: -c 192 DINUC INDEL).
-e or --exome	-> optional: Simulates based solely on the exome regions of the genome.
-s or --simulations	-> optional: Followed by the desired number of iterated simulations per sample. The default number of iterations is 1 (ex: -s 100 will produce 100 iterations per sample).
-u or --update	-> optional: Updates the chromosomes as each mutation is assigned.
-b or --bed	-> optional: Followed by the file name of the BED file. Allows for simulations within a set of ranges. (ex: -b regions_for_sim.txt).
-i or --indel	-> optional: Creates the matrix for the limited list of INDELs (classifies insertions at micro-homologies as insertions at repeats of 0).

-ie or –extended_indel -> optional: Creates the matrix for the complete list of INDELs.
-S or --Signatures -> optional: Simulates based upon a matrix that consists of a set of mutational signatures with their respective activities for each sample. This parameter requires that a file is saved within the *references/matrix/* folder with the following extension: "*.mutSignatures"*.

FOLDER STRUCTURE

The framework contains three folders (scripts, references, output), the main simulator code, and this readme file [include pictures snapshot]. The scripts folder contains all code required to generate the necessary input files, reference files, and mutational matrix catalogues. The references file contains a *vcf_files* folder, *matrix* folder, and *chromosomes* folder. The *vcf_files* folder is where the user must place their input files (must be in vcf, maf, or simple text file format). The *matrix* folder will contain all of the generated mutational matrix catalogues. The user may upload their own matrix as long as it follows the format given in the *examples* folder, otherwise the simulation code will generate the necessary matrices. The *chromosomes* folder contains a *chrom_string* folder, *context_distributions* folder, *exome* folder, *fasta* folder, *transcripts* folder, and *tsb* folder. NOTE: ALL OF THESE FILES MUST BE PLACED UNDER THE GENOME NAME WITHIN EACH SUBDIRECTORY. The *output* folder is where all simulated data will be stored after completion.

INPUT FILE FORMAT

This tool currently supports *maf, vcf,* and *simple text* file formats (See the *examples* folder for examples of each format). The user must provide variant data adhering to one of these three formats. If the user's files are in *vcf* format, each sample must be saved as a separate file.

DESCRIPTION OF PROVIDED EXAMPLES

An *example_test* folder is provided that contains 10 test samples. This folder can be placed within the *references/vcf_files/* folder and run on the command line as follows:

```
python mutation_simulation.py –g GRCh37 –p example_test –c 192 DINUC –s 10
```

**The SNV context can be changed to 96, 192, 1536, or 3072, and the simulation parameter can be changed to any value.

The final simulated output will be saved within the *simulation_output/example_test_simulations_GRCh37_[context]/* folder.

SIMULATING ADDITIONAL GENOMES

If the user desires to use a genome other than those currently supported (GRCh37, GRCh38, mm9, or mm10), they must:

1)Download the individual FASTA files for each chromosome
2)Place them into the *references/chromosomes/fasta* folder
3)Download the transcriptional data following the format presented in the *transcripts_original* folder.

4)Place this file within the *transcripts_original* folder
5)Download the exome ranges for the given genome and place in the *references/chromosomes/exome* folder.

ADDITIONAL DATA AND EXAMPLES

COPYRIGHT

CONTACT INFORMATION

Please address any queries or bug reports to Erik Bergstrom at ebergstr@eng.ucsd.edu