Evotum (demo)

This perl script is the accompanying code for the theoretical framework presented in “Estimating growth patterns and driver effects in tumor evolution from individual samples”, with authors Leonidas Salichos, William Meyerson, Jonathan Warrell and Mark Gerstein.

Evotum dependencies

For running *EvoTum_demo.pl* the user needs to install additional perl packages *Statistics-R-0.33*, *Regexp-Common-0.01* and *IPC-Run-0.80.*
The following libraries need to be declared at lines 3-5:

```
use lib “your-path-to/R/Statistics-R-0.33/lib/";
use lib “your-path-to/R/Regexp-Common-0.01/lib/";
use lib "your-path-to/R/IPC-Run-0.80/lib/";
```

Input files

The script accepts (pseudo)VCF as text files -see test files- where the user has to provide which tab denotes the chromosome, the reference and altered nucleotide and the VAF frequency. The script does not correct for ploidy and CNV. On line 66, the user can de:

```
######### Define VCF columns
$chrom_col=0;   #       Chromosome column
$pos_col=1;     #       Mutational position column
$vaf_col=7;     #       VAF column
$ref_nt_col=3;  #       Reference nucleotide column
$alt_nt_col=4;  #       Altered nucleotide column
$pat_id_def="provided-or-file-ID";   #       Patient ID
```

Parameters

As also described in the script, the user can run different modes or modify a number of parameters:

I) Modes (script line 25)

- $fit_expo=1; #Runs the main algorithm. Can be set to 0 for only generating the intermediate vcf files. Their format is ID_Chromosome_MutationalPosition_RefNT_AltNT_VAF.
- $resume_mode=0; # When running a list or multiple jobs, resume_mode=1 deters from running the same VCF again. To rerun delete corresponding $resume_file or set to 0.
- $resume_vcf_mode=0; # If equal to ‘1’, this mode restarts the algorithm from the last mutation it analyzed based on a log text file called "RESUME_temp_XXXX". Not recommended at this demo stage. Instead, the user can directly re-set the starting mutations in the command line (see examples).
- $sampling_r_mode=-1; # Leave '-1' for the DEFAULT approach with direct Λ-ambda calculation when estimating growth r.
- $ generation_mode=1; # DEFAULT measure of k. Script is optimizing for local generational time t_g. Growth r is calculated independently with direct Λ-ambda calculation.
- $general_classic_mode=0; # Recommended for first or defining main clonal mutation. No optimization for generational time t_g. Vulnerable to initial perturbation (e.g. CNV). Hitchiker generations are starting from generation 1 corresponding to hitchhiker 1 without calibration or offset.
- $reoptimize_r_mode=0; # Similar to $general_classic_mode, no optimization for generational time t_g, but growth r is estimated through “Non Linear Squares” optimization and not Λ-ambda.
- $verbose_mode=1; # Print results on screen while running in addition to files; Helpful to immediately pick up abort-mutation when the algorithm exit due to bad optimization (see examples below).

Modes $generation_mode, $general_classic_mode and $reoptimize_r_mode are not exclusive and can be run at the same time.

II) Parameters (line 37)

- $mut_pass_window=150; # The size of the sliding window for number of m=150 hitchhikers; No k and r values will be reported for the first *m* of the sample's mutations. In our analysis, we normally used *m*=100 or *m*=150. Too large will often result in reduced k vales and averaged growth. Too small will be more unstable to small changes and model optimization and parametrization.
- $coverage=0; # Resample VAF Frequency to simulate equal or lower than 1000x coverage. Use '0' for no resampling.

Useful script changes

- *Switching between VAF and frequency*: Line 141, modify 2*VAF
- *Changing output directory***:** Line 46. Currently output is printed inside TESTFILES folder
- *Popping VAF frequency*: Sometimes it might be hard to define the frequency-column in a pseudo vcf file. At the same time, VAF is often found as/at the last column. User can switch between lines 137 and 138 to pop the frequency, if at the last column of the text file.

Examples

<u>The output files can be found inside TESTFILES/OUTFILES/.</u>

To run the algorithm the user needs to specify at least the file name. Using our play test data included in TESTFILES, the simplest form is:

perl EvoTum_demo.pl TESTFILES/108.psdVCF

This will run the script until ordered mutation number 517 or “Ordered mut.:517” where R will fail to optimize and exit. Consequently, the user can type:

perl EvoTum_demo.pl TESTFILES/108.psdVCF 518

This will restart the algorithm from mutation number 518. The user can superimpose a new sliding-window-size by providing an additional third argument as:

perl EvoTum_demo.pl TESTFILES/108.psdVCF 0 120, where 0 is the starting mutation and 120 the size of the sliding window.

File output

When run as previously, the algorithm prints in new **TAB** text file TESTFILES/OUTFILES/108.psdVCF_cov0 with the following output:
1st Column/**Ordered_mut.:** the number of the ordered mutation.
2nd Column/**genTime:** Estimated generational time **t_g**.
3rd Column/**Independent_pos._growth_r:** Independent calculation of positive growth (if *r* >0).
4th Column/ **Independent_neg._growth_r:** Independent calculation of negative growth (if *r* >0).
5th Column/ **DEFAULT_scalar_k_with_tg_opt | scalar_k_with_classic_mode(no_tg_opt) | scalar_k_re-opt_with_dependent_r:** scalar multiplier *k*. If more than one modes are used, different scalar *k*s are separated with space bar.
6th Column/ **k*r_growth:** k*r describes the driver’s new growth where scalar *k* is multiplied with hitchhiking growth to estimate new base-growth for fitness population.

7th Column/ **DEFAULT_growth_r_with_tg_opt| Independent_growth_r_calc| growth_r_dependent_re-opt(no_tg):** hitchhiker base growth *r* parameter similar to columns 3&4. This column is included in case the user selects mode reoptimize_r_mode which provides a new re-optimized growth rate *r'* based on NLS optimization of *m* hitchhikers instead of a direct calculation through Λ-ambda.
8th Column/**e.g. 108.psdVCF_16_18542407_G_A_0.808:** Provides information on the specific mutations, including sample id (108.psdVCF), exact position (chrom+position), nucleotide change and mutation frequency.