COMPUTEL V1.3

User Manual

Contents

Introduction

Computel is R-based software for computation of mean telomere length and relative abundance of telomeric repeat variants from whole genome next generation sequencing (NGS) data.

Development

Computel has been developed by the members of the Bioinformatics Group at the Institute of Molecular Biology of the National Academy of Sciences of the Republic of Armenia (IMB NAS RA). You can visit the group's webpage at the following link: http://big.sci.am.

Citation

When using Computel in your research, please refer to the GitHub repository at https://github.com/lilit-nersisyan/computel.

License

Requirements

Computel v1.3 works for **Unix** systems (tested for Linux Ubuntu), refer to Computl v02 for Windows and Mac OS tested versions.
It uses the following dependencies, which should be installed on your system: **R 3.0.3** or higher, **Samtools 1.3** or higher, **bowtie2** 2.4 or higher.

Download and installation

Computel can be downloaded from GitHub at https://github.com/lilit-nersisyan/computel.
Download and extract the package into a local directory. The folder contains the needed scripts, binaries and files for checking proper setup. Do not change the relative location of the folders within the package.

Setting up samtools

If samtools 1.3 or higher is not installed on your system, download it from https://github.com/samtools/samtools. You may install it by running "make install" or you may run "make" and specify the samtools executable with -sam option of Computel.

Running Computel from command line

To make computel.sh executable, navigate to the Computel folder and run:

```
chmod +x computel.sh
```

The following lines demonstrate the input-output arguments and how to run Computel, the next sections will provide more detailed explanations on arguments and the algorithm.

Getting started

Basic usage:

```
./computel.sh [options] {-1 <fq1> -2 <fq2> -3 <fq3> -o <o>}
```

Input:

```
<fq1> fastq file (the first pair or the only fastq file (for single end
reads)
<fq2> fastq file (optional: the second pair of fastq files, if exists)
<fq3> fastq file (optional: the third pair of fastq files, if exists)
<o>   output directory (optional: the default is computel_out)
```

Output:

Computel v.1.3 produced two outputs:

- output.dir/tel.length.txt

A tab delimited text file. Contains the mean telomere length in bp’s, as well as other important metrics.

- output.dir/tel.variants.txt

A tab delimited text file, where you may find the absolute and relative percentages of each telomeric variant.

Options (advanced):

<-proc> number of processors to be used (default: 4)

<-sam> samtools path (optional: if not supplied, Computel will use the samtools installed on the system)

<-bowal> bowtie2-align path (optional: the bowtie2-align is located at computel's bin directory by default.)

<-bowb> bowtie2-build path (optional: the bowtie2-build is located at computel's bin directory by default.)

<-nchr> number of chromosomes in a haploid set (the default is 23)

<-lgenome> whole genome length (the default is 3244610000)

<-pattern> telomere repeat pattern (the default is 'TTAGGG'; change this if you're using Computel for a non-human organism)

<-minseed> the min seed length (read length minus the number of flanking N's in the telomeric index; should be in the range [12-read.length]; This is a tested and carefully set parameter (defualt = 12); Change this only if you REALLY KNOW what you're doing!)

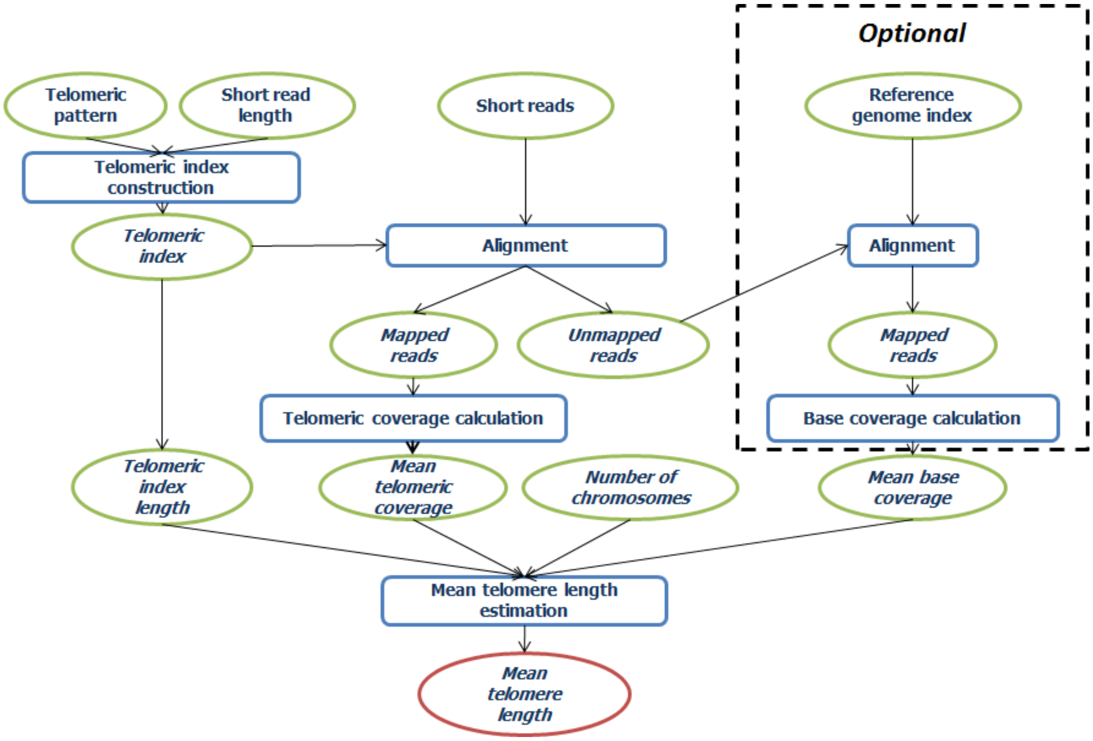
`<-qualt>` Phred+33 quality threshold for telomeric repeat variant calling (default is 25. We recommend changing this value to 56 for Solexa+64 and Phred+64 quality formats)

`<-ref>` Reference genome index prefix path (generated with bowtie2-build) that you'd like to use for more accurate estimation of base coverage.

Algorithm description

Computation of mean telomere length

The algorithm workflow is presented in the figure. For its detailed description, refer to the main paper [Nersisyan L, Arakelyan A (2015) Computel: Computation of Mean Telomere Length from Whole-Genome Next-Generation Sequencing Data. PLoS ONE 10(4): e0125201. doi:10.1371/journal.pone.0125201].



Computation of telomeric repeat variant composition

Telomeric repeat variants are sequence fragments within telomeric reads that differ from the canonical telomeric repeat pattern. The number of variants may increase in certain conditions, and their nature may be condition-specific. For example, it has been shown that in certain ALT positive human cell lines, the C-type variants (TCAGGG) are abundant. Computel, thus, provides abundance of canonical and variant repeat patterns from the sam files generated after alignment of reads to the telomeric index. For each telomeric read, Computel takes all the repeats, with positions accurately adjusting by insertions and deletions, as indicated by CIGAR strings. If the repeat differs from the canonical repeat pattern, than Computel performs a quality check (in case of mismatched or inserted bases) to ensure the variation is not a result of sequencing error, and adds the pattern to the list of available repeat variants. By default, the quality threshold is 25, accounting for Phred+33 quality format. This value can be adjusted with the "qualt" option. Computel converts the quality threshold to ASCII and adds 33. Therefore, for Solexa+64 and Phred+64 quality formats, the quality threshold should be specified with a 31 offset (56 corresponds to 25). The variant patterns at the ending fragments of the reads are not taken into consideration, as those could be from subtelomeric sequences. The output is a text file “tel.variants.xls”, where all the patterns with their abundance, relative abundance, and relative abundance among non-canonical variants only are reported.

Input and advanced options

Fastq files:

Computel uses whole genome fastq files as input. If the reads are single end, usually only one fastq file is available (use only -1 <fq1> option and skip the rest). If the reads are paired-end there will be two or three fastq files (the third one contains unpaired reads), which can be supplied with -1 <fq1>, -2 <fq2> and -3 <fq3> options.

The output directory will contain the index and alignment files generated by Computel and the final output file (tel.length.xls). The directory may be specified to replace the default (/computel_out) with the option -o <outputdir>.

Options (advanced):

<-proc>: number of processors to be used for parallelization of the alignment process (default: 4)

<-sam>: samtools path. Specifying Samtools directory rather then using the default binary available in the Computel package should only be done if the latter doesn’t work. Please, refer to the “Setting up samtools” section for details.

<-bowal>: bowtie2-align path. This is also optional: the bowtie2-align is located at computel's bin directory by default. Replace it only if the binary does not work.

<-bowb>: bowtie2-build path. This is also optional: the bowtie2-build is located at computel's bin directory by default. Replace it only if the binary does not work.

<-nchr>*: number of chromosomes in a haploid set (the default is 23 for human genomes). Use this for organisms other than humans or for special cases where the number of chromosomes is not a multiple of 23. Note, you might need to change the genome length accordingly.

<-lgenome>: whole genome length (the default is 3244610000 for humans)

<-pattern>: telomere repeat pattern (the default is 'TTAGGG'; change this if you're using Computel for a non-human organism). Note, this is the repeat pattern present at the 3’ end (on the 5’-3’ strand).

<-minseed>**: the min seed length (read length minus the number of flanking N's in the telomeric index; should be in the range [12-read.length]; This is a tested and carefully set parameter (defualt = 12); Change this only if you REALLY KNOW what you're doing!)

<-qualt> Phred+33 quality threshold for telomeric repeat variant calling (default is 25. We recommend changing this value to 56 for Solexa+64 and Phred+64 quality formats). This is the quality scheme used to distinguish between sequencing errors and telomeric repeat variants.

<-ref> Reference genome index prefix path (generated with bowtie2-build) that you’d like to use for more accurate estimation of base coverage. You may run the `bowtie2-build [PATH_TO_GENOME_FASTA] [PREFIX_PATH]` command to generate your genome prefix. Please, refer to the bowtie2-build manual for details.

*Notes on number of haploid chromosomes

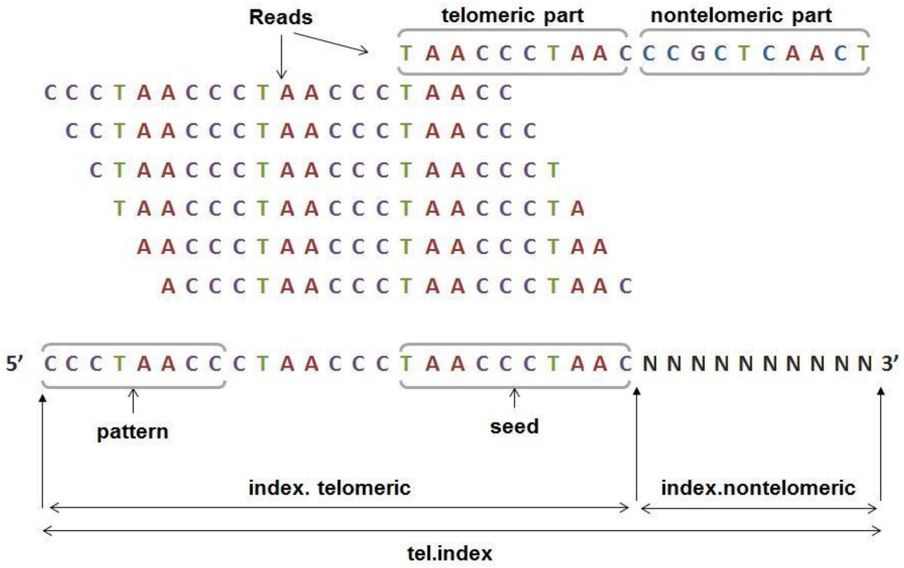
Telomere length is calculated with the formula:

```
(mean(tel.cov/base.cov))*(rl+pl-1) / (2*num.haploid.chr),
```

where *tel.cov* is coverage at telomeric index and *base.cov* is coverage at reference genome; *rl* is read length and *pl* is telomeric pattern length, *num.haploid.chr* is the number of haploid chromosomes.
The division by *num.haploid.chr* is for deriving at a mean value of telomere length for each chromosome end. The number 2 is to account for the two ends of each chromosome. Note that the ploidy of the genome will have no influence on the results, meaning that, adjusted for the rest of the parameters, the results will not differ for haploid, diploid and polyploid genomes.

**Min.seed

This is an advanced option and specifies the minimum number of telomeric repeat bases. More specifically, this value is used for building the telomeric index with trailing N bases of length (*read.length* - *min.seed*), which is the "index.nontelomeric" region of the index in the figure below.



If this option is omitted, the default value of 12 will be used. Increasing **min.seed** will increase specificity but decrease sensitivity of capturing telomeric reads, and vice-versa.
We believe that for the majority of biological cases, the default value is optimal.

Notes on some configuration options

Pattern

Telomeric repeat pattern is specified for the 3'-end of the chromosome. E.g. in case of human telomeres, the sequence at 3'-end it is TTAGGG, while the 5'-end has pattern CCCTAA.

If the pattern is omitted in the configuration file, the default value TTAGGG for human telomere repeats will be used. In case of a different study organism another pattern should be specified.

Number of haploid chromosomes

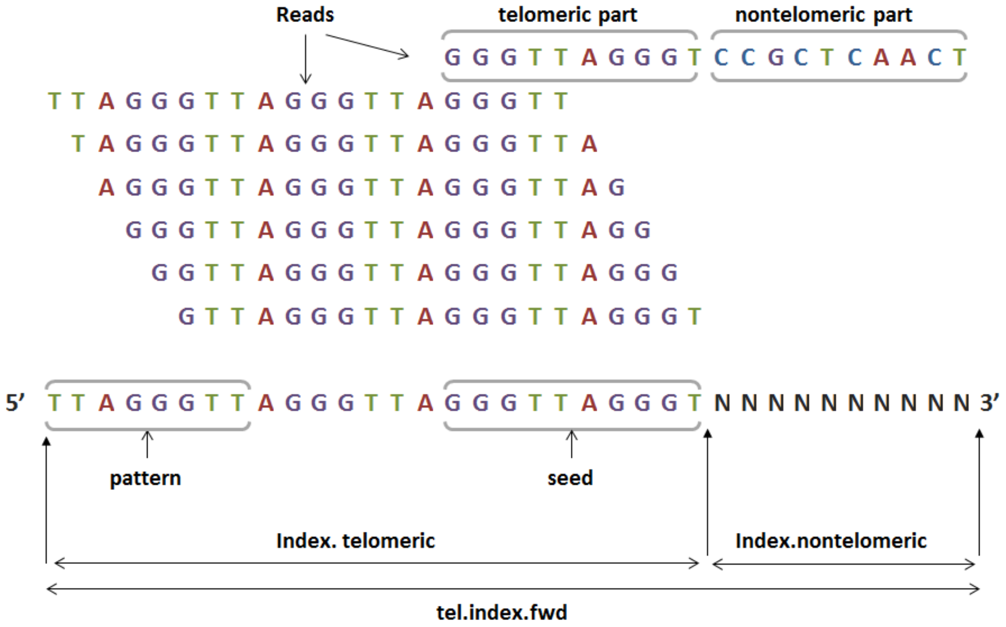
Telomere length is calculated with the formula:

(mean(*tel.cov*/*base.cov*))*(rl+pl-1) / (2*num.haploid.chr*),*

where *tel.cov* is coverage at telomeric index and *base.cov* is coverage at reference genome; *rl* is read length and *pl* is telomeric pattern length, *num.haploid.chr* is the number of haploid chromosomes.
The division by *num.haploid.chr* is for deriving at a mean value of telomere length for each chromosome end. The number 2 is to account for the two ends of each chromosome. Note that the ploidy of the genome will have no influence on the results, meaning that, adjusted for the rest of the parameters, the results will not differ for haploid, diploid and polyploid genomes.

Min.seed

This is an advanced option and specifies the minimum number of telomeric repeat bases. More specifically, this value is used for building the telomeric index with trailing N bases of length (*read.length* - *min.seed*), which is the "index.nontelomeric" region of the index in the figure below.


Reads
telomeric part
nontelomeric part
G G G T T A G G G T C C G C T C A A C T
T T A G G G T T A G G G T T A G G G T T
T A G G G T T A G G G T T A G G G T T A
A G G G T T A G G G T T A G G G T T A G
G G G T T A G G G T T A G G G T T A G G
G G T T A G G G T T A G G G T T A G G G
G T T A G G G T T A G G G T T A G G G T
5' T T A G G G T T A G G G T T A G G G T T A G G G T N N N N N N N N N N 3'
pattern
seed
Index. telomeric
Index.nontelomeric
tel.index.fwd

If this option is omitted, the default value of 12 will be used. Increasing **min.seed** will increase specificity but decrease sensitivity of capturing telomeric reads, and vice-versa.
We believe that for the majority of biological cases, the default value is optimal.

FAQ

The FAQ section will be generated once a question arises.

Feel free to ask questions and make suggestions by mailing to Lilit Nersisyan at l_nersisyan@mb.sci.am.