telomerecat-docs Documentation

Release 1.0

JHR Farmery

June 07, 2017

Contents

Author Henry Farmery

Date June 07, 2017

Version 1.0

Welcome to the documentation for telomerecat.

Email jhrf2@cam.ac.uk with any queries.

Contents:

CHAPTER 1

Telomerecat - Estimate telomere length from WGS samples

Telomerecat estimates the average length of telomeres given whole genome sequencing (WGS) files as input. Length estimation takes into account noise genereated by interstital telomeric sequences and the subtelomere to provide a more accurate estimate. Additioanlly, *Telomerecat* accounts for anneuploidy in WGS samples, without the users having to input an estimate of choromsomes. The biology and technical aspects of WGS are considered in an attempt to only measure true telomereic reads.

Telomerecat is designed to run quickly on either your desktop or remote high powered computing facility. The amount of processors is adjustable to the computing resources available to the user.

Telomerecat is available as a pre-compiled binary for Linux and MacOSX. Effort has been made to increase backwards compatibility by compiling the binary against old versions of glibc (v2.12).

If the pre-compiled binary does not work for you can install *Telomerecat* as a normal python package using *pip*:

```
pip install Telomerecat
```

Telomerecat's python dependancies will be installed automatically, however you will need to ensure that your system has an installed fortran compiler so that scipy will install. It is also good practise to install telomerecat inside a virtualenv (this is best practise for all python pacakges!).

For a full description of the method please read the article of biorxiv.

Preface - Quickstart

For those who don't want to read all of the docs before jumping in, try the code snippets below to get going with telomerecat as quickly as possible.

First you'll need to ensure that telomerecat is installed. Find out more in the *Installation* section .

Once you're sure telomerecat is installed, open your terminal and create a directory in which to run your telomerecat analysis:

```
cd ~
mkdir telomerecat-analysis
cd telomerecat-analysis
```

Now let's run telomerecat:

```
telomerecat bam2length -v2 /path/to/bam_file.bam
```

A .csv file with an estimate of length will be produced for your specified BAM file. With these default parameters and depending on how powerful your computer is, expect telomerecat to take an hour and a half to process ~2 billion reads.

Useful Document Links

- genindex
- modindex
- search

CHAPTER 2

Installing telomerecat

These instructions will allow you to install and run telomerecat on your Linux or Mac computer.

Install using pip

telomerecat can be installed on most systems using pip (just like thousands of other python programs).

pip is a platform for installing python packages from the Python Package Index. It is widely available and comes as standard with many distribtuions of Mac OSX and Linux. You probably already have this program on your computer. Try typing the following in your console to see if you have *pip* installed:

```
pip -V
```

If pip is installed on your system you should see some text describing the version of pip installed on your system. If your terminal reports that the command is not found you'll have to either install pip (instructions are readily available on the web) or use one of the other installation methods.

If you have root permissions on the machine that you hope to run telomerecat on the following command will install telomerecat and all dependencies:

```
pip install telomerecat
```

If this doesn't work because of a permissions error you will need to either gain root permission (try adding sudo infront of the above command) or use a virtual environment. Virtual environments are a very common way of installing and running python based software. Full instructions on downloading and setting up a python virtual environment may be found here.

Install using docker

Ensure that docker is installed and working on your computer. Instructions for docker can be found on the docker website.

Telomerecat is then installed as any other with any other docker package:

```
docker pull telomerecat
```

CHAPTER 3

Using Telomerecat

Telomerecat estimates telomere length from whole genome sequencing files (WGS). The first step is to scan the input BAM file for any reads that looks even remotely like a telomere. It collects all of these reads and deposits them in a TELBAM. TELBAMs can take around an hour and a half to generate from a BAM file of around 150GB. However, once a TELBAM has been generated, it can be used to quickly generate telomere length estimates. Using these small TELBAMs means you don't need to store a full BAM file to conduct reproducible results. TELBAMS are generated with the command *bam2telbam*.

Once we have generated a TELBAM we can use it to generate a length estimation. This can be done using the *telbam2length*. It usually takes around 2 minutes to generate a length estimate from a TELBAM.

To streamline this process users may wish to conduct both steps using the *bam2length* command. This will take a BAM file as input and output a length estimate .csv file. This command will automatically output TELBAMs for each of the inserted BAM files.

bam2length

The most straightforward way of generating a telomere length estimate from a BAM file is by using the *bam2length* command. This command takes a BAM file and generates a length estimate in a .csv file.

Invoke the *bam2length* command by inputing the following command into your terminal:

```
telomerecat bam2length /path/to/example.bam
```

If output to the command line is desired the option *-v* should be used. You should substitute a real path name in the place of `path/to/example.bam`.

A full list of parameters that can be supplied to this command can be found with the following command:

```
telomerecat bam2length --help
```

The bam2length command is actually just a convenient wrapper for the bam2telbam and telbam2length commands. If working with a large batch of data (where large is more than approximately 5 samples) it is best practise to first generate a batch of TELBAMs using the bam2telbam command and then to run the telbam2length command on the entire batch simultaneously. This will make use of information from within the cohort for F2a correction which may help estimation in low coverage samples.

bam2telbam

Most of the computational effort and time expended during a run of telomerecat is spent generating the TELBAM. Telomerecat must iterate over the entire BAM file to identify all telomere reads. Users may wish to split the time intensive TELBAM generation from the relatively short process of length estimation.

To enable this separation, telomerecat allows the user to generate the TELBAM using a separate command.

```
telomerecat bam2telbam /path/to/example.bam
```

This command is straightforward and takes very few parameters. However, the user should provide the desired number of processing cores to telomerecat using the *-p* option. Specifying more processing cores will enable telomerecat to run more quickly.

telbam2length

The telbam2length command is used to generate a TL estimate from a TELBAM or multiple TELBAMs. A TELBAM is simply a subset of all the sequencing reads in the BAM which contain the sequence "TTAGGG" or "CCCTAA" at least twice. The pairs of any of the reads matching the above criteria are also included. TELBAMs are generated automatically by both the bam2length and bam2telbam commands.

telbam2length is invoked with the following call to the command line:

```
telomerecat telbam2length /path/to/example1_telbam.bam ...
```

The user may pass multiple TELBAMs to a single run of telbam2length.

In some cases the user may find it useful to run cohort correction to the F2a measurement for each sample. We find that cohort correction is especially useful on lower coverage data and low quality samples. One important consideration when using cohort correction is that samples should be from the same sequencing batch. The cohort correction method uses information about the ratio of F2 and F4 reads on a population levels and different sequencing chemistries and platform seem to produce differing ratios of these reads. Thus, using cohort correction across sequencing batches may cause less accurate estimation.

The user can specify NOT to run F2a correction with the *-d* option.

csv2length

The csv2length command allows users to generate estimations using a CSV that was previosuly output teloemrecat. This is useful when considering whether or not to use cohort correction. For instance, a user may generate a cohort corrected CSV using the telbam2length command and then insert the resultant csv into the csv2length command without cohort correction. This will save time as meta data does not need to be generated from individual TELBAMs.

csv2length is invoked with the following call:

. code-block:: shell

    telomerecat csv2length /path/to/example1_telbam.csv ...

CHAPTER 4

Understanding Telomerecat Output

Telomerecat outputs length estimates in the form of a .csv file. The name of the .csv file always takes the form `telomerecat_[UNIXTIME].csv`. This section explains the format of this file.

What follows is a series of sections detailing what each of the columns in the output file means. The sections are organised in order of appearence in the header.

A greater understanding of these terms may be gained from reading the following paper ...(paper forthcoming)

F1

This is the amount of *F1* reads in the sample. We can think of F1 reads as reads which are completely telomeric. F1 reads originate from the nucleotide after the boundary to the most distal end of the telomere.

F2

The amount of F2 reads in the sample. F2 reads are reads where one end is completely telomeric and the other is not. Additionally, the completely telomeric end is comprised of the pattern CCCTAA.

F4

The amount of F2 reads in the sample. F4 reads are reads where one end is completely telomeric and the other is not. Additionally, the completely telomeric end is comprised of the pattern TTAGGG.

Psi

This is a measure of fidelity from your sample, it is used in the F2a correction method. The greater this value, the more we believe the observed measurement of F2a. A description of how this variable is derived for each sample is given in the paper.

Insert_mean

The insert size of the sample. This is either estimated from the TELBAM (default) or input by the user.

Insert_sd

The standard deviation of the insert size. As for the mean, this is either estimated from the TELBAM (default) or input by the user.

F2a

F2a reads are the estimated number of reads covering the boundary between telomere and nontelomere. The number is given by F2 - F4. The logic being that any F4 read is known to have come from an interstitial or subtelomere region. Any F4 read will have a companion F2. By subtracting F4 from F2 we find the approximate number of reads without a F4 companion. These are reads over the boundary. A more

F2a_c

This is F2a count used in the simulation, it represents the number of F2a reads after undergoing batch correction. If F2a correction is disabled by the user, this number will be the same as F2a.

A full description of the F2a correction formula is given in the paper.

Length

The telomere length as estimated by telomerecat. Length is output in basepairs