Yleaf manual

Version 3.0

Contents

1 Installation

Yleaf requires a Linux operating system or Windows Subsystem for Linux. The latest version of Yleaf has been tested on Ubuntu 18.04.6 LTS. Yleaf is coded in python and needs minimally python 3.6 to function properly.

1.1 Installing with conda

An easy way of installing Yleaf is with the use of conda. The steps to installing yleaf with conda are the following:

```
# create the conda environment
conda create --name yleaf_env python=3.6

# activat the environment
conda activate yleaf_env

# install required python libraries
conda install pandas
conda install numpy

# install Burrows-Wheeler Aligner for fastQ files
conda install -c bioconda bwa

# install SAMtools
conda install -c bioconda samtools

# clone the yleaf repository
git clone https://github.com/genid/Yleaf.git

# finally if you want to use fastQ files use the provided install.py
    script to download required genome data
cd Yleaf
python install.py
```

1.2 Manual install

Another option is to install everything manually:

```
# install python and libraries
apt-get install python3.6
pip3 install pandas
pip3 install numpy

# install Burrows-Wheeler Aligner for fastQ files
sudo apt-get install bwa

# install SAMtools
wget https://github.com/samtools/samtools/releases/download/1.4.1/
    samtools-1.4.1.tar.bz2 -O samtools.tar.bz2
tar -xjvf samtools.tar.bz2 3. cd samtools-1.4.1/
./configure 5. make
make install

# clone the yleaf repository
git clone https://github.com/genid/Yleaf.git

# finally if you want to use fastQ files use the provided install.py
    script to download required genome data
cd Yleaf
python install.py
```

2 Usage and examples

2.1 Command line options

Here is a written version of all the command line options. All of this information can also be accessed with the use of the -h/–help command.

```
-h  Shows this help message and exit

-fastq [File]  Path of single raw reads or a folder which contains all
    raw reads asFASTQ

-bam [File]    Path of single BAM file or a folder which contains BAM
    files

-cram [File]   Path of single CRAM file or a folder which contains CRAM
    files

-f[hg19/hg38].fasta Reference build genome aligned from bwa index [only -
    fastq and -cram]

-pos [hg19/hg38]  Positions files genome reference containing SNPs based
    on ISOGG tree. These files can be found in the Position_files folder
    , you can also make these files yourself.

-out [STRING]  Name, or location of the output path (e.g. out, or /home/
    name/out)

-r [INT]   The minimum number of reads for each base above on the quality
     threshold

-q [INT]   Minimum quality for each read, integer between 10 and 39,
    inclusive. If you give it 0, the quality of reads will not be
    checked

-b [INT]   The minimum percentage of a base result for acceptance. For
    example, if you give it 90, then 90% of the reads for each market
    should be the same, otherwise that market will be filtered out

-t [INT]   Set number of additional threads to use [CPUs] during the
    alignment process with BWA MEM and indexing of BAM files with
    SAMtools

-old   New in version 3.0. Allows to run the old prediction script. This
     will not work with new position files unfortunately
```

2.2 Usage

Yleaf accepts FASTQ, BAM and CRAM files as input. Below we showed some examples of how to run it on each case.

```
# FASTQ
python Yleaf.py -fastq raw_reads.fastq -f reference_indexed/genome.
    fasta -r 1 -q 20 -b 90 -t 1 -pos Position_files/{WGS_hg19.txt -
    WGS_hg38.txt} -out out

# BAM
python Yleaf.py -bam alignment.bam -pos Position_files/{WGS_hg19.txt
    -WGS_hg38.txt} -out out -r 1 -q 20 -b 90

# CRAM
python Yleaf.py -cram alignment.cram -pos Position_files/{WGS_hg19.
    txt - WGS_hg38.txt} -f genome.fasta -out out -r 1 -q 20 -b 90
```

2.3 Usage of predict_haplogroup.py specifically

Haplogroups are always predicted at the end of a Yleaf run but if you want to run the script separately for some reason see the following example:

```
# Single output file:
python haplogroup_prediction.py -input [path]/[sample name].out -out
    sample name.hg

# Multiple output files (batch mode):
python haplogroup_prediction.py -input [path to folder with output
    files] -out sample name.hg
```

Keep in mind that this relies on files generated by Yleaf.

2.4 Output file formats

2.4.1 .out file

The main output file. It contains a tab separated file including the following:

- Chr: Chromosome used (Y-chromosome in all cases)
- Pos: Given position during mpileup function for each base
- Marker name: Given name that concord a specific region to the positions files
- Haplogroup: Haplogroup that corresponds to the positions file.
- Mutation: Base mutation given in the positions file
- Ancestral: Base for the ancestor given in the positionsfile

- Derived: Base for derive given in the positions file
- Reads: Number of reads after quality filtering
- Called percentage: Percentage of reads that agrees with the final base call
- Called base: The final base call that meets predefined quality threshold
- State: A for ancestral state, D for derived state
- Depth: New in version 3.0. The depth of the hit in the yfull tree. The higher the number the more specific the haplogroup.

2.4.2 .chr file

Tab separated file containing the percentage of mapped reads. This file is not used for prediction but is usefull as a quality control

- Chr: Chromosome location from the alignment file
- Reads: Number of mapped reads in each chromosome given by the SAMtools command idxstats
- Perc: Percentage of the number of mapped reads per chromosome in the alignment.

2.4.3 .log file

A text file containing general information about the sample and the run, such as the total number of reads, the number of markers that met the threshold, and the number of markers that failed

2.4.4 .fmf file

A tab separated file including the same columns (minus the depth column) as the PREFIX.out with an addition column "description" which gives information of why the marker did not pass the criteria for haplogrouping. This could have happened due to zero read and/or low coverage and below the threshold for base calling. In some cases the user may decide to use information from this file to optimize the QC-settings.

2.4.5 .hg file

This is the file generated by the automated haplogroup prediction. The software will produce a single file for every run, in the case of a batch run this file will contain predictions for all samples. Keep in mind that this prediction is not perfect and it is still recommended to manually verify the prediction that are made by inspecting the other files that the software tool produces (i.e. the prediction pipeline does not take into account markers from the ".fmf file" which may be relevant). The output file is a tab separated file including the following columns:

- Sample name: Sample name used during analysis (same as bam filename)
- Hg: Final haplogroup prediction using yfull nomenclature (new in version 3.0). The -old option can be used in order to use the ISSOG nomenclature using only ISSOG data.
- Hg Marker: Final haplogroup prediction using marker nomenclature (i.e.D-Y15320(xPH3836))
- Total number of reads
- Total number of valid markers, thus the number of markers considered in the haplogrouppprediction
- QC-score: Overall quality score which is the factor of the three scores below. If the overall score falls below 0.75, first the algorithm will attempt to make an alternative prediction that does meet the threshold, if no prediction with the required quality can be made it will show no haplogroup and a manual interpretation of the sample specific output files is needed.
- QC-1: This score indicates whether the predicted haplogroup follows the expected backbone of the haplogroup tree structure (i.e. if haplogroup E is predicted the markers defining: A0-T, A1, A1b, BT, CT, DE should be in the derived state, while other intermediate markers like: CF, F, GHIJK, etc, are expected to be in the ancestral state). The score is calculated by dividing the number of markers that show the expected state, by the sum of all intermediate markers. A score of 1 shows that all markers are in the expected state and indicates high confidence if the prediction of the correct broad haplogroup, if lower values are observed it is highly recommended to manually inspect the [sample_name].out file.
- QC-2: This score indicates whether equivalent markers to the final haplogroup prediction were found in the ancestral state. I.e. if the final haplogroup is R1b1a1a2a2, there are two markers in the assay defining this haplogroup: Z2103 and Z2105, if both are found to be derived the QC2 value will be 1. However if one is in the derived and the other in the ancestral state the QC2 value will be calculated as number of derived equivalent markers divided by the total number of equivalent markers, in this example the QC2 value would be 0.5. As the overall QC-score uses a threshold of 0.75, regardless of the other QC-metrics this haplogroup prediction would be rejected. In such a case the algorithm would look for a another prediction which does meet the overall QC-threshold which in most cases will be the parental branch, so in this exampleR1b1a1a2a.
- QC-3: This score indicates whether the predicted haplogroup follows the expected within-haplogroup tree structure. I.e. if the predicted haplogroup is O2a1c (O-JST002611), it is expected that markers defining: O2a1, O2a, O2 and O are also in the derived state. A score of 1 shows that all markers are in the expected state and indicates high confidence

in the haplogroup prediction, if lower values are observed it is highly recommended to manually inspect the [sample_name]. outfile.

3 Update information

Yleaf version 3.0 (April 2022):

- Slight update to the haplogroup prediction Q3 score.
- Haplogroup relation tree is now based on yfull and ISOGG data. Not using the ISOGG marker identifiers anymore as a result.
- Tree has been updated with new information from both Yfull and ISSOG

Yleaf version 2.2 (March 2020):

- Minor bug fixes in haplogroup prediction algorithm
- Minor bug fixes in the genotyping algorithm
- Included CRAM format files as supported input
- Included position files for two targeted genotyping assays

Yleaf version 2.1(December 2019):

- Bug fix in the haplogroup prediction algorithm

Before version 2.0:

- Upgraded marker list with bugs from Yleaf v2.0 fixed, based on ISOGG (July2019)
- Moved from Python2.7 to Python3.6
- Replaced R-code for Python code, significant increase in run speed

4 Contact information

Please feel free to send an email to b.vanwersch@erasmusmc.nl if there are problems getting the software up and running.For questions concerning the interpretation of the results and more general feedback concerning to tool please write to a.ralf@erasmusmc.nl