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:

```
# first clone this repository to get the environment_yleaf.yaml
git clone https://github.com/genid/Yleaf.git
cd Yleaf
# create the conda environment from the .yaml the environment will be
     called yleaf
conda env create --file environment_yleaf.yaml
# activate the environment
conda activate yleaf
# pip install the cloned yleaf into your environment. Using the -e
    flag allows you to modify the config file in your cloned folder
pip install -e .

# verify that Yleaf is installed correctly. You can call this command
     from any directory on your system
Yleaf -h
```

Keep in mind that you will need to use conda activate yleaf every time you want to use the software in a new terminal.

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 minimap2
# 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
# pip install the yleaf repository
```

```
cd Yleaf
pip install -e .

# verify that Yleaf is installed correctly. You can call this command
     from any directory on your system
Yleaf -h
```

After installation you can navigate to the yleaf/config.txt folder and add custom paths for the files listed there. This will make sure that Yleaf does not download the files on the first go or downloads the files in the provided location. This allows you to use a custom reference if you want. Please keep in mind that custom reference files might cause other issues or give problems in combination with already existing data files. Positions are based on either hg38 or hg19.

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, --help          show this help message and exit
-fastq PATH, --fastq PATH
                    Use raw FastQ files
-bam PATH, --bamfile PATH
                    input BAM file
-cram PATH, --cramfile PATH
                    input CRAM file
-force, --force     Delete files without asking
-rg {hg19,hg38}, --reference_genome {hg19,hg38}
                    The reference genome build to be used. If no
                        reference
                    is available they will be downloaded. If you added
                    references in your config.txt file these will be
                        used
                    instead as reference or the location will be used
                        to
                    download the reference if those files are missing
                        or
                    empty.
-o STRING, --output STRING
                    Folder name containing outputs
-r INT, --reads_treshold INT
                    The minimum number of reads for each base.
                    (default=10)
-q INT, --quality_thresh INT
                    Minimum quality for each read, integer between 10
                        and
```

```
                  40. [10-40] (default=20)
-b INT, --base_majority INT
                  The minimum percentage of a base result for
                  acceptance, integer between 50 and 99. [50-99]
                  (default=90)
-t INT, --threads INT
                  The number of processes to use when running Yleaf.
-old, --use_old   Add this value if you want to use the old
    prediction
                  method of Yleaf (version 2.3). This version only
                      uses
                  the ISOGG tree and slightly different prediction
                  criteria.
-dh, --draw_haplogroups
                  Draw the predicted haplogroups in the haplogroup
                      tree.
-hc, --collapsed_draw_mode
                  Add this flag to compress the haplogroup tree image
                  and remove all uninformative haplogroups from it.
-p, --private_mutations
                  Add this flag to search for private mutations.
                      These
                  are variations that are not considered in the
                  phylogenetic tree and thus not used for haplogroup
                  prediction, however can be informative and
                  differentiate individuals within the same
                      haplogroup
                  prediction.
-maf FLOAT, --minor_allele_frequency FLOAT
                  Maximum rate of minor allele for it to be
                      considered
                  as a private mutation. (default=0.01)
```

2.2 Usage

Yleaf accepts FASTQ, BAM and CRAM files as input. Below we show some examples of how to run it on each case. If you want to use the old version of the prediction script you will need to add an additional -old flag. The commands shown here are minimal working examples. There are additional options to configure sensitivity of predictions, an option to draw haplogroups of all individuals predicted in a tree as well as an option to filter for private mutations.

```
# FASTQ
Yleaf -fastq raw_reads.fastq -o fastq_output --reference_genome hg38

# BAM
Yleaf -bam file.bam -o bam_output --reference_genome hg19

# CRAM
```

```
Yleaf -cram file.bam -o cram_output --reference_genome hg38

# BAM file and drawing all predictions in a tree
Yleaf -bam file.bam -o bam_output --reference_genome hg19 --
    draw_haplogroups

# BAM file and finding private mutations
Yleaf -bam file.bam -o bam_output --reference_genome hg19 --
    private_mutations --minor_allele_frequency 0.05
```

3 Output file formats

3.1 .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 positions file
- 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(v10.01) tree. The higher the number the more specific the haplogroup.

3.2 .chr file

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

- Chr: Chromosome location from the alignment file
- Reads: Number of mapped reads in each chromosome given by the SAM-tools command idxstats

- Perc: Percentage of the number of mapped reads per chromosome in the alignment.

3.3 .info 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

3.4 .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.

3.5 .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(v10.01) 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 haplogroup prediction
- 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 example R1b1a1a2a.

- 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]. output file.

3.6 .pmu file

This is a file that is generated if the -p flag is specified. It lists all private mutations that have been identified in the sample. These are mutations where the minor allele has a frequency under 1% (can be changed with -mr option) and is not associated with a haplogroup. This file is a tab separated file containing the following columns:

- Chromosome: is always going to be Y

- Position: Position on the provided reference genome.

- dbSNP reference number: The dbSNP reference number if available for that position

- Mutation: What the mutation is at the position. major ->minor.

- Reference: major allele base.
- Actual: minor allele base.
- Reads: Total number of reads
- Called percentage: Total number of valid markers, thus the number of markers considered in the haplogroup prediction.
- Frequency: population frequency from the dbSNP database if it is available.

3.7 hg_tree_image.pdf

This file will be added if the -dh flag is provided. It will draw all the haplogroups that are predicted for the different samples in a haplogroup tree. This will give an overview of how samples are related to one another.

4 Version changes

Yleaf version 3.1 (November 2022):

- Added an option to find private mutations.
- Added an option to draw a haplogroup tree showing the relation between samples.
- Simplified command line input. You now only have to provide what reference genome you are interested in and the rest of the files will be located automatically
- install.py is not used anymore. Instead files are downloaded the first time or you can use a config file to point to where you want Yleaf to look for reference genome data.
- Yleaf can now be called from anywhere after pip installing it.

Yleaf version 3.0 (April 2022):

- Haplogroup relation tree is now based on YFull(v10.01) and ISOGG data. Not using the ISOGG marker identifiers anymore as a result.
- Tree has been updated with new information from both YFull(v10.01) and ISSOG
- A haplogroup can now be ancestral, derived or undefined. This depends on the percentage of markers in that state. If more than 60% of markers are in ancestral or derived the haplogroup will the majority group. If the 60% threshold is not met the haplogroup is undefined and will not be taken into account for quality score calculations. This change affects the scores of QC1 and QC3.

- A script to convert ISOGG marker tables into a format usable by Yleaf. These tables can unfortunately only be used together with the -old flag.

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

5 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