Manual and Installation settings for

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

Updates compared to Yleaf v2.0

- Upgraded marker list with bugs from Yleaf v2.0 fixed, based on ISOGG (July 2019)

Updates compared to Yleaf v1.0

- Moved from Python2.7 to Python3.6
- Replaced R-code for Python code, significant increase in running speed
- Included the option to analyze data aligned to GRCh38
- Updated marker list according to ISOGG (March 2019)
- Included automated haplogroup prediction
- Included a log file for each sample for CQ purposes
- Included a filtered markers file (*.fmf) showing all markers that were excluded and for what purpose

Requirements

- Operating system: Linux only. Tested on Ubuntu 16.04LTS, but should also work on newer version of Ubuntu. It should be easy to made it work on other Linux distributions.
- Python, R, wget, git.
- Internet connection during installation (for downloading and extracting hg19 reference genome).
- Data storage: For installation we recommend a storage capacity of > 10 GB.

Installation

1. Install dependencies (you can skip this step if these packages are already installed on your system)

    apt-get install python3.6
    pip3 install pandas
    pip3 install numpy
    sudo apt-get install bwa

SAMtools

We recommend the newest version of SAMtools (e.g. >= 1.4.1)

1. wget https://github.com/samtools/samtools/releases/download/1.4.1/samtools-1.4.1.tar.bz2 -O\ samtools.tar.bz2
2. tar -xjvf samtools.tar.bz2
3. cd samtools-1.4.1/
4. ./configure
5. make
6. make install

2. Run following script install.py to download and build index from hg19 or hg38 genome reference with BWA (**FastQ files for alignment only**).

   python install.py

Installation using an environment (OPTIONAL)

Explanation:
#1) creates a new environment with a given name (e.g. yleaf-env)
#2) activate the environment with the given name
#3) install pandas library with the latest release
#4) Run Yleaf normally
#5) deactivates the virtual environment or just close the current terminal

To run in the terminal:
1) python3.6 -m venv yleaf-env
2) source yleaf-env/bin/activate
3) pip install pandas
4) python Yleaf.py
5) source deactivate

Usage and examples

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 or CRAM

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

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

Command line options explained

Provide an input file (FASTQ, BAM or CRAM) and output prefix name

-fastq [File]	Path of single raw reads or a folder which contains all raw reads as FASTQ
-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
-out [STRING]	Name, or location of the output path (e.g. out, or /home/name/out)

Then there are the following options

-h	Shows this help message and exit
-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

Output file formats

PREFIX.out

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

By sorting on the “Haplogroup” column and filtering on the derived alleles in the “State” column a list of

derived markers will be shown that can easily be follow to assign the most derived haplogroup detected by the software

PREFIX.chr contains a tab separated file including the following:

- 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.

Note: This file is not needed for haplogroup assignment, but can be useful for quality control purposes.

PREFIX.log

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

PREFIX.fmf

A tab separated file including the same columns as the Sample.csv 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.

Run Name.hg

We included a new option which is the automated haplogroup prediction, this could be especially useful when analyzing a large number of samples. However, 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 software will produce a single file for every run, in the case of a batch run this file will contain predictions for all samples. The output file is a tab separated file including the following columns:

- Sample name: Sample name used during analysis (same as bam file name)
- Hg: Final haplogroup prediction using ISOGG nomenclature [March 2019] (i.e. D1a2a1)
- 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].out file.

Usage and examples for predict_haplogroup.py

The haplogroup prediction can also be run separately from the Yleaf pipeline.

Haplogroup prediction accepts text file or a folder containing text files as input. These files should be generated by Yleaf (see above). Below we show examples on how to run it.

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
```

Command line options explained

Provide an input file (.out) output prefix name

-input [STRING] Path of single text file or folder which contains output files produced from Yleaf

-out [STRING] Name of the output file containing the haplogroup prediction

There is also a help option

python haplogroup_prediction.py -h Shows this help message and exit

Output file format

[Sample/folder name].hg

Same as described above.

Please feel free to send an email to d.montielgonzalez@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