Manual and Installation settings for

Forensic Y-SNP analysis beyond SNaPshot: High-resolution Y-chromosomal haplogrouping from low quality and quantity DNA using Ion AmpliSeq and massively parallel sequencing technology

Requirements

- Operating system: Tested in Linux only (it may also work on Mac or in a docker). Tested on Ubuntu 16.04LTS, but it should also work on newer versions of Ubuntu. And it should be easy to implement on other Linux distributions.
- Python, wget
- 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
   apt-get install p7zip-full
   apt-get install mawk
   ```

   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

Usage and examples for clean_tree.py

Clean_tree accepts BAM files as input. Below we show examples on how to run it on each case (reads **must** to be aligned to **hg19**).

Single BAM file

```
python clean_tree.py -bam /[path]/[sample name].bam -out out -r 20 -q 20 -b 95
```

Multiple BAM files (batch mode)

```
python clean_tree.py -bam /[path to folder with bam files] -out out -r 20 -q 20 -b 95
```

Command line options explained

Provide an input file (BAM) and output prefix name

-bam [STRING]	Path of single BAM file or a folder which contains BAM files
-out [STRING]	Name of the output folder or absolute path to store all additional files (run name)

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 otherwise that marker will be filtered out and store in the .fmf output file
-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 (bases below this threshold will be excluded from the analysis)
-b [INT]	The minimum percentage of a base result for acceptance. For example, if you give it 95, then 95% of the reads for each market should be the same, otherwise that marker will be filtered out and store in the .fmf output file

Output file format

[Sample name].out

A tab separated file including the following columns, including all QC-accepted markers:

- Chr: Chromosome used (Y-chromosome in all cases)
- Pos: Location of the marker on in the hg19 reference genome
- Marker name
- Haplogroup: Haplogroup that corresponds to the marker (Note: the haplogroup name may chance when the haplogroup tree is updated, it is recommended to check the latest version of the ISOGG tree for the current nomenclature using the marker name).
- Mutation: Excepted mutation
- Ancestral: Base call for the ancestral allele
- Derived: Base call for the derived allele

- Reads: Number of reads covering the marker 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 thresholds
- 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.

[Sample name].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.

[Sample name].log

Contains information about performance during the analysis of this specific sample; such as the number of markers that have failed due to reads coverage or base calling percentage and the number of markers that did provided haplogroup information.

[Sample name].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 [August 2018] (i.e. D1a2a1)
- Hg Marker: Final haplogroup prediction using marker nomenclature (i.e. D-Y15320(xPH3836))

- 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 haplogroup_prediction.py

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

Haplogroup prediction accepts text file or a folder containing text files as input. These files should be generated by clean tree (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 clean_tree

-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 at 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