yhaplo™ | Software Manual

G. David Poznik
23andMe

Last updated: August 17, 2023

Table of Contents

1 Introduction

`yhaplo` identifies the Y-chromosome haplogroup of each male in a sample of one to millions. It does not rely on any particular genotyping modality or platform, and it is robust to missing data, genotype errors, mutation recurrence, and other complications. Although full sequences yield the most granular haplogroup classifications, genotyping arrays can yield reliable calls, provided a reasonable number of phylogenetically informative variants has been assayed.

Briefly, haplogroup calling involves two steps. The program first builds a representation of the Y-chromosome phylogeny by reading its primary structure from (Newick-formatted) text and then importing phylogenetically informative SNPs from the ISOGG database[1], storing each SNP within a specific node and growing the tree as necessary. It then traverses the tree for each individual, identifying for each the path of derived alleles leading to a haplogroup designation.

To learn more about the algorithm, please see our bioRxiv preprint[2]:

> Poznik GD. 2016. Identifying Y-chromosome haplogroups in arbitrarily large samples of sequenced or genotyped men. bioRxiv doi: 10.1101/088716
>
> http://biorxiv.org/content/early/2016/11/19/088716

2 Downloading and running `yhaplo`

`yhaplo` is available for non-commercial use pursuant to the terms of the non-exclusive license agreement included with the software distribution:

> https://github.com/23andMe/yhaplo

To clone the repository, issue the following command from an appropriate directory:

```
git clone git@github.com:23andMe/yhaplo.git
```

Alternatively, download `yhaplo` from the repository web site by clicking the green “Code” button then “Download ZIP.” Move the file to an appropriate place and unzip it.

To install:

```
cd yhaplo
pip install --editable .
```

To test-run the software on example data, issue the following command from any directory:

```
yhaplo --example_text
```

This option sets the input genotype file to a test dataset derived from full sequences generated by the 1000 Genomes Project[3]: `1000Y.subset.genos.txt`. It includes the Y-chromosome genotypes of 35 globally diverse males at ~60,000 SNPs.

The program will first read and process built-in metadata files (**section 3**). Then, based on the genotype filename's extension (`.genos.txt`), `yhaplo` concludes that the data to be analyzed are in a simple sample-major text format (**section 4.1.1**). It will read these data and identify the Y-chromosome haplogroup for each individual in the sample.

The software can also operate on VCF and BCF formats (**section 4.2.1**), among others (**section 4**) and produce several helpful auxiliary output files (**section 5**). To see the full list of command-line options, some of which are described in this manual, issue the following command:

```
yhaplo --help
```

To generate all auxiliary output files, use the `--all_aux_output` option, which is implicitly turned on with the `--example_text` option.

To test on a larger dataset, one could download the Y-chromosome genotypes for all 1,244 males from phase 3 of the 1000 Genomes Project[3]. To do so, navigate to this FTP directory:

```
https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/supporting/chrY/
```

Then download this file:

```
ALL.chrY_10Mbp_mask.glia_freebayes_maxLikGT_siteQC.20130502.60555_biallelic_snps.vcf.gz
```

For clarity, rename this file to `1000Y.all.vcf.gz`, then run `yhaplo` with the `--input` option:

```
yhaplo --input 1000Y.all.vcf.gz
```

3 Metadata files

3.1 Primary tree structure

This Newick-formatted text file codifies the primary structure of the Y-chromosome tree.

```
data/tree/y.tree.primary.DATE.nwk
```

3.2 Phylogenetically informative SNPs

This table includes details for ~20,000 phylogenetically informative SNPs. The contents were curated by the International Society of Genetic Genealogy (ISOGG) and scraped from their website[1] on the date specified in the filename.

```
data/variants/isogg.DATE.txt
```

3.3 Quality control for phylogenetically informative SNPs

Several files are necessary to correct a number of glitches in the SNP file described above. We encourage users to augment these files in light of any further inconsistencies they may find.

Two files correct physical coordinates and ancestral- and derived-allele designations, respectively. These files have the same format as the post-QC SNPs file described in **section 6.1.1**:

```
data/variants/isogg.correct.coordinate.txt
data/variants/isogg.correct.polarize.txt
```

Three files, also formatted as in **section 6.1.1**, indicate SNPs to drop in light of inconsistencies observed in test data:

```
data/variants/isogg.omit.bad.txt
data/variants/isogg.omit.bad.23andMe.txt
data/variants/isogg.omit.branch.conflict.txt
```

This file lists physical coordinates of multiallelic sites to exclude:

```
data/variants/isogg.multiallelic.txt
```

This file lists SNPs with known shared recurrences not reflected in the ISOGG table. Although the software does not use this file, we provide it for completeness.

```
data/variants/isogg.split.txt
```

3.4 Representative markers

These two files list the names of markers deemed representative of corresponding haplogroups:

```
data/variants/representative.SNPs.isogg.YEARtree.txt
data/variants/representative.SNPs.additional.txt
```

The first was generated by parsing the ISOGG tree of the given year, and the second is a manually edited supplementary list. `yhaplo` ignores the first column of each file.

3.5 Preferred SNP names

This file includes a list of preferred SNP names:

```
data/variants/preferred.snp_names.txt
```

4 Genotype input files

`yhaplo` can read genotype data in sample-major or variant-major orientations. Since haplogroups are properties of individuals, it is more memory efficient to use a sample-major format, but the program will run on variant-major data as long as the full data matrix can fit in memory. This should be the case for sample sizes on the order of ten thousand or fewer individuals, but to call haplogroups in samples on the order of hundreds of thousands or millions, it is probably best to transpose or split the data before running.

In all cases, specify the genotype input file with the `--input` option.

4.1 Sample-major formats

4.1.1 Genos

Example: `project_name.genos.txt[.gz]`.

Expected format:

Row 0	Physical coordinates (GRCh37).
Column 0	Individual identifiers. The first element is ignored.
Cell (i, j)	Genotype for individual i at position j. Values include {"A", "C", "G", "T", "."}, with "." indicating an unobserved value.

4.1.2 Ped/Map

In this sample-major format, the `.ped` file has six identifier columns and 2m allele columns, one pair for each of m markers. The `.map` file has m rows, one for each marker, ordered as in the `.ped` file. For data in PLINK[4] format (`.bed`, `.bim`, `.fam`), one can export to ped/map format with the following command:

```
plink --bfile project_name --recode --out project_name
```

Then, to convert the ped/map data to genos format, use the utility script included with `yhaplo`:

```
yhaplo_convert_to_genos project_name.ped
```

4.2 Variant-major formats

4.2.1 VCF, BCF

VCF[5]-formatted and BCF-formatted input must be indexed:

VCF: `project_name.vcf.gz, project_name.vcf.gz.tbi`
BCF: `project_name.bcf, project_name.bcf.csi`

Note: When calling genotypes from sequencing data, please be sure to **emit all confident sites**, not just those at which alternative alleles were observed. Ref/alt status is unimportant to `yhaplo`, but ancestral/derived status is, and the reference sequence contains many derived alleles. `yhaplo` will not be happy if you discard these valuable data by generating "variants-only" VCFs. To limit compute time and file size, you could safely restrict to positions in:

```
output/isogg.snps.unique.DATE.txt,
```

as these are the only SNPs `yhaplo` considers. To generate this file, just run:

```
yhaplo
```

with no arguments.

4.2.2 23andMe text download

The utility script referred to in **section 4.1.2** can also convert single-sample text files downloaded from 23andMe. Just rename the file to `iid.23andMe.txt` and run:

```
yhaplo_convert_to_genos iid.23andMe.txt
```

This file can have header lines whose first characters are "#" or "rsid," and its data columns are assumed to be:

1. SNP identifier. The script ignores this column.
2. Chromosome. The script retains only those rows with values of "24" or "Y."
3. Physical coordinate. Reference human assembly build 37 (GRCh37) is assumed.
4. Genotype. The script drops rows with values not in {A, C, G, T, D, I}.

A fifth column bearing a "second allele" is allowed. The script ignores any row for which the value of the fifth column differs from that of the fourth.

5 Haplogroup output files

`yhaplo` generates a log file and a haplogroup calls file. Users also have the option to generate any of several helpful auxiliary output files, or all with the `--all_aux_output` option. The `--out_dir` option specifies the output directory, which is `output/` by default.

5.1 Haplogroup calls

File name: `output/haplogroups.project_name.txt`

Columns

1. ID
2. Haplogroup short form, with the name of a SNP observed in the derived state
3. Haplogroup short form, with the name of a representative SNP
4. Haplogroup long form, using Y-Chromosome Consortium nomenclature[6]

Example line

```
HG01791 R-M167 R-M167 R1b1a2a1a2a1b1a1
```

Column 2 vs. column 3
The SNP names of columns 2 and 3 may differ. `yhaplo` picks for each branch of the tree a single SNP to represent the equivalence class of SNPs associated with the branch. Column 3 uses this representative SNP, but the individual may not have been genotyped for it. Furthermore, it is actually possible that the individual has been genotyped for the SNP, and an ancestral allele was observed. This can happen when the ISOGG representation of the tree is incomplete, and the set of SNPs associated with a branch is not a true equivalence class. For example, if there are 10 SNPs associated with a given branch, and we observe 5 ancestral alleles and 5 derived alleles, the true haplogroup is one that shares some portion of the given branch before diverging from it, and the representative SNP may have arisen after this divergence. In contrast, the SNP cited in column 2 will necessarily have been observed in the derived state for this individual, although the SNP itself may be less commonly known.

5.2 Counts of ancestral- and derived-allele genotypes

This optional output file records for each individual counts of ancestral- and derived-allele genotypes encountered along each branch visited in the search path. The file contains one block for each individual, with the last line indicating the final haplogroup call.

Command-line option: `-c, --anc_der_counts`

File name: `output/counts.anc_der.project_name.txt`

Columns

Body of each block

1. ID
2. Branch label (haplogroup)
3. Number of ancestral-allele genotypes observed
4. Number of derived-allele genotypes observed

Last line of each block

1. ID
2. YCC haplogroup
3. Pipe separator (|)
4. Haplogroup short form, with the name of a SNP observed in the derived state
5. Haplogroup short form, with the name of a representative SNP

Example block

```
NA18960  A00        2   0
NA18960  A0-T       0   1
NA18960  A0        30   0
NA18960  A1         0  12
NA18960  A1a       15   0
NA18960  A1b        0  30
NA18960  BT         0 333
NA18960  B          1   0
NA18960  CT         0 191
NA18960  DE         0  26
NA18960  E         95   0
NA18960  D          0  36
NA18960  D1b        0   5
NA18960  D1b2      29   0
NA18960  D1b1a      1   0
NA18960  D1b1b      1   0
NA18960  D1b1d      0   1
NA18960  D1b1a2     2   0
NA18960  D1b1d1    29   0
NA18960  D1b1d | D-CTS6609 D-CTS6609
```

5.3 Haplogroup paths

This optional output file provides a compact summary of the path through the tree leading to the each individual's haplogroup call. It lists each branch on which derived-allele genotypes were observed and counts thereof. This information is a subset of that described in **section 5.2**.

Command-line option: `-hp, --haplogroup_paths`

File name: `output/paths.project_name.txt`

Columns

1. ID
2. YCC haplogroup
3. Haplogroup short form, with the name of a representative SNP

4. Pipe separator (|)
5+. Branch labels (haplogroups) and numbers of derived-allele genotypes, colon-separated

Example line

```
NA18960 D1b1d D-CTS6609 | A0-T:1 A1:12 A1b:30 BT:333 CT:191 DE:26 D:36 D1b:5
   D1b1d:1
```

A variation of the `--haplogroup_paths` option was added in version 1.0.15:

Command-line option: `-hpd, --haplogroup_paths_detail`

With this option, the number of derived-allele genotypes observed on each branch is followed by a comma-separated list of SNPs observed with the derived allele. This option is very similar to the `--der_snps` option described in **section 5.4**.

Example line, with many SNPs omitted for clarity

```
NA18960 D1b1d D-CTS6609 | A0-T:1:L1120 A1:12:P305,…,L1112 A1b:30:P108,…,Z17900
   BT:333:M42,…,Z17390 CT:191:M168,…,Y1528 DE:26:M145,…,PF1833
   D:36:M174,…,JST021355 D1b:5:M55,M64.1,M179,M359.1,P37.1 D1b1d:1:CTS6609
```

5.4 Derived-allele genotypes observed

This optional output file includes for each individual a list of SNPs observed in the derived state on the path leading to the individual's lineage. This information is an expanded version of that described in **section 5.3**.

Command-line option: `-ds, --der_snps`

File name: `output/derived.snps.project_name.txt`

Columns

1. ID
2. YCC haplogroup
3. Haplogroup short form, with the name of a representative SNP
4. Pipe separator (|)
5+. SNPs, each with haplogroup and common name, separated by a colon

Example line, with many SNPs omitted for clarity

```
NA18960 D1b1d D-CTS6609 | A0-T:L1120 A1:P305 … A1:L1112 A1b:P108 … A1b:Z17900
   BT:M42 … BT:Z17390 CT:M168 … CT:Y1528 DE:M145 … DE:PF1833 D:M174 …
   D:IMS-JST021355 D1b:M55 D1b:M64.1 D1b:M179 D1b:M359.1 D1b:P37.1
   D1b1d:CTS6609
```

5.5 Derived-allele SNP details

This optional output file includes detailed information about each SNP whose derived allele was observed on the path through the tree. Each block is essentially the transpose one row of the file described in **section 5.4**, with additional information for each SNP.

Command-line option: `-dsd, --der_snps_detail`

File name: `output/derived.snps.detail.project_name.txt`

Columns

First line of each block

As in the haplogroup calls file (**section 5.1**).

Body of each block

1. ID
2. SNP name
3. Branch label (haplogroup)
4. Physical coordinate (GRCh37)
5. Mutation type: ancestral allele and derived allele, separated by a two-character arrow (->)

Example block, with many SNPs omitted for clarity

```
NA18960 D-CTS6609 D-CTS6609 D1b1d
NA18960  L1120           A0-T       14496439 G->T
NA18960  P305            A1          2710154 A->G
…
NA18960  L1112           A1          8466995 G->A
NA18960  P108            A1b        15426248 C->T
…
NA18960  Z17900          A1b        23572106 G->A
NA18960  M42             BT         21866840 A->T
…
NA18960  Z17390          BT         28746408 G->C
NA18960  M168            CT         14813991 C->T
…
NA18960  Y1528           CT         15932327 A->G
NA18960  M145            DE         21717208 C->T
…
NA18960  PF1833          DE         22014732 C->G
NA18960  M174            D          14954280 T->C
…
NA18960  IMS-JST021355   D           2828425 A->G
NA18960  M55             D1b        21872738 T->C
NA18960  M64.1           D1b        21903383 A->G
NA18960  M179            D1b        14838700 C->T
NA18960  M359.1          D1b        14491671 T->C
NA18960  P37.1           D1b        14491684 T->C
NA18960  CTS6609         D1b1d      17008697 A->T
```

5.6 Ancestral-allele genotypes observed

This optional output file includes for each individual a list of SNPs encountered in the ancestral state during the search. This information is an expanded version of that described in **section 5.5**, and it complements the file described in **section 5.4**.

Command-line option: `-as, --anc_snps`

File name: `output/ancestral.snps.project_name.txt`

Columns

As in the derived-allele genotypes file (**section 5.4**).

Example line, with many SNPs omitted for clarity

```
NA18960 D1b1d D-CTS6609 | A00:L1110 A00:L1111 A0:L991 … A0:L1055 A1a:M31 …
   A1a:V215 B:M181 E:M96 … E:CTS3337 D1b2:CTS583 … D1b2:Z17171 D1b1a:M125
   D1b1b:M151 D1b1a2:IMS-JST022457 D1b1a2:CTS107 D1b1d1:CTS1897 …
   D1b1d1:Z14880
```

5.7 Ancestral-allele SNP details

This optional output file includes detailed information about each SNP whose ancestral allele was encountered in the search. Each block is essentially the transpose one row of the file described in **section 5.6**, with additional information for each SNP. The file complements the one described in **section 5.5**.

Command-line option: `-asd, --anc_snps_detail`

File name: `output/ancestral.snps.detail.project_name.txt`

Columns

As in the derived-allele SNP details file (**section 5.5**).

Example block, with many SNPs omitted for clarity

```
NA18960 D-CTS6609 D-CTS6609 D1b1d
NA18960  L1110          A00         9847534 C->T
NA18960  L1111          A00         8268136 C->T
NA18960  L991           A0         14497059 C->A
…
NA18960  L1055          A0         21622096 G->A
NA18960  M31            A1a        21739754 G->C
…
NA18960  V215           A1a         2868499 T->G
NA18960  M181           B          14851554 T->C
NA18960  M96            E          21778998 C->G
…
NA18960  CTS3337        E          14845090 C->T
NA18960  CTS583         D1b2        6863631 T->G
```

```
…
NA18960  Z17171          D1b2        9813700 A->G
NA18960  M125            D1b1a      21930287 T->C
NA18960  M151            D1b1b      21892634 G->A
NA18960  IMS-JST022457   D1b1a2     24464597 G->C
NA18960  CTS107          D1b1a2      2731887 C->T
NA18960  CTS1897         D1b1d1     14115321 C->T
…
NA18960  Z14880          D1b1d1     22589060 G->T
```

6 Additional output files

6.1 Phylogenetically informative SNPs

Upon reading and cleaning the raw set of phylogenetically informative SNPs (**section 3**), the program emits three files.

6.1.1 Post-QC SNPs

This file lists all SNPs considered for haplogroup classification. An individual SNP can have multiple entries—one for each common name associated with it.

File name: `output/isogg.snps.cleaned.DATE.txt`

Columns

1. SNP name
2. Branch label (haplogroup)
3. Physical coordinate (GRCh37)
4. Mutation type: ancestral allele and derived allele, separated by a two-character arrow (`->`)

6.1.2 Unique post-QC SNPs

This file is similar to the post-QC SNPs file (**section 6.1.1**), but with one entry per SNP.

File name: `output/isogg.snps.unique.DATE.txt`
Format: As in **section 6.1.1**, with a fifth column:
a comma-separated list of all common names for the SNP

6.1.3 Markers dropped from consideration

This file lists all markers dropped from consideration due to inconsistencies observed in test data.

File name: `output/isogg.snps.dropped.DATE.txt`
Format: As in **section 6.1.1**.

6.2 Trees and traversals

6.2.1 Trees

Upon building the tree, the program writes four Newick-formatted files:

```
output/y.tree.ycc.DATE.nwk
output/y.tree.ycc.hg_snp.DATE.nwk
output/y.tree.aligned.ycc.DATE.nwk
output/y.tree.aligned.hg_snp.DATE.nwk
```

Each details the topology of the tree, with the latter two, labeled "aligned," including artificial branch lengths to align the leaf nodes when plotting. Branches are labeled with YCC-style haplogroup names in the files labeled "ycc" and with representative-SNP-style haplogroup names in the files labeled "hg_snp." To visualize the final tree in plain text, users can plot it with the provided utility script, which wraps a Biopython[7] function:

```
yhaplo_plot_tree --newick_fp output/y.tree.aligned.hg_snp.DATE.nwk
```

Alternatively, they can plot the tree graphically by importing to third-party tree plotting software.

6.2.2 Traversals

For an even simpler text-based visualization of the tree, use one of the traversal options:

```
-b, --breadth_first    Write bread-first traversal
-d, --depth_first      Write depth-first (pre-order) traversal
```

These yield:

```
output/y.tree.bf.traversal.DATE.txt
output/y.tree.df.traversal.DATE.txt
```

7 Search Parameters

`yhaplo` has two command-line options to alter parameters of its modified breadth-first-search algorithm. The default values should generally be fine, but some users may wish to try tweaking them. Please see the code and docstring of the following method for details:

```
tree.Tree.identify_phylogenetic_path
```

For a discussion of these options, please see:

https://github.com/23andMe/yhaplo/pull/6

7.1 Stopping condition

```
-ast, --anc_stop_thresh
      BFS stopping condition parameter (default: 2)
```

`yhaplo` will curtail a search path when at least one of three conditions is met:

1. It has reached a leaf.
2. It found more than `anc_stop_thresh` ancestral alleles on the most recent branch.
3. It found exactly `anc_stop_thresh` ancestral alleles and zero derived alleles on the most recent branch.

7.2 Collapsing condition

```
-dct, --der_collapse_thresh
      BFS collapsing parameter (default: 2)
```

When `yhaplo` observes `der_collapse_thresh` derived alleles on a branch, it will hone in on the descendants of that branch and discard parallel search paths.

8 References

1. ISOGG. International Society of Genetic Genealogy. (2016). http://www.isogg.org
2. Poznik, G. D. Identifying Y-chromosome haplogroups in arbitrarily large samples of sequenced or genotyped men. *biorXiv* (2016). doi:10.1101/088716
3. Poznik, G. D. *et al.* Punctuated bursts in human male demography inferred from 1,244 worldwide Y-chromosome sequences. *Nat. Genet.* **48,** 593–9 (2016).
4. Purcell, S. *et al.* PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **81,** 559–75 (2007).
5. Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics* **27,** 2156–2158 (2011).
6. The Y Chromosome Consortium. A nomenclature system for the tree of human Y-Chromosomal binary haplogroups. *Genome Res.* **12,** 339–348 (2002).
7. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25,** 1422–1423 (2009).