SNAPPY: Single Nucleotide Assignment of Phylogenetic Parameters on the

Y chromosome

User Manual

Alissa L. Severson, Jonathan A. Shortt, Christopher R. Gignoux

Updated: November 16, 2021

### Table of Contents:

[Table of Contents: 2](#_Toc527623675)

[Summary: 3](#_Toc527623676)

[Terms of Use: 3](#_Toc527623677)

[Quick Start: 3](#_Toc527623678)

[Software Description 3](#_Toc527623679)

[Parameters 6](#_Toc527623680)

[Notes and Considerations: 6](#_Toc527623681)

[Future Improvements 7](#_Toc527623682)

[Citation 7](#_Toc527623683)

### 

### Summary:

SNAPPY is a software program used to assign Y-chromosome phylogeny-informed haplotgroups using dense genotype data. The program efficiently tests all haplogroups from a provided Y-chromosome haplogroup database to find the one that is best supported by input genotypes. Importantly, the method considers both the amount of support for the specific haplogroup, as well as its ancestral haplogroups. This accounts for the underlying genealogy the haplogroups represent, strengthening the accuracy of the assignments.

### Terms of Use:

SNAPPY is made available for use to blah blah blah. It’s free.

### Quick Start:

Download SNAPPY from [www.github.com/chrisgene](http://www.github.com/chrisgene), and navigate in the terminal to the folder where SNAPPY is saved. The folder should also contain the files parse\_ref\_files.py, parse\_plink\_files.py, and a directory called ‘ref\_files’ that contains four additional files. Run SNAPPY with the following command:

python SNAPPY\_v123.py --infile plink\_library

where ‘123’ is substituted for the correct downloaded version of SNAPPY, and ‘plink\_library’ is the prefix name of the genotypes to be analyzed. Note that SNAPPY uses plink (v1.9) in a preprocessing step so a plink (v1.9) executable must be listed in the user’s path as ‘plink’ for preprocessing steps. If plink (v1.9) is not available in the path, the input can be created using plink with the ‘--recodeAD’ option.

### Software Description

**Required input files: For convenient use, SNAPPY accepts input data formatted as a common plink binary library consisting of a .bed file, a .bim file, and a .fam file, each with the same base name, or as a .vcf file. Positions on autosomes, the mitochondrial genome, or the X-chromosome should be filtered out prior to running SNAPPY. Other necessary input files that are used to read and store SNP-haplogroup assignments, and haplogroup ancestor-descendant relationships on the Y-chromosome tree are included in the SNAPPY distribution in the ‘ref\_files’ directory.**

**Output files:** After performing assignments, SNAPPY writes two output files. The first, the .out file (default= chrY\_hgs.out, but controlled by the ‘out’ parameter), is a tab-separated file where each line gives a sample id, the sample’s haplogroup assignment, the haplogroup score, and the list of that haplogroup’s informative alleles used in determining the score. The second file, the .all file (default=chrY\_hgs.all, but controlled by the ‘out’ parameter), is a tab-separated file where each line lists the sample number followed by every haplogroup that exceeded a threshold score (see Parameters section) in the format ‘Hapologroup:Score.’ This allows users to manually adjust haplogroup assignments where necessary.

**Reference File Sources: Files included in the ‘ref\_files’ directory include: pos\_to\_allele.txt, id\_to\_pos.txt, y\_hg\_and\_snps.sort, and tree\_structure.txt. The first three files contain information about positions and id’s of snps on the Y-chromosome, and on to which haplogroups are informed by the snps. The final file, tree\_structure.txt, details information on haplogroup descent where parent or child haplogroup names do not follow the Y-chromosome haplogroup naming conventions. These files were created from Y-chromosome trees maintained by the International Society of Genetic Genealogy (ISOGG), and from discussions with experts in Y-chromosome history.**

**We anticipate updating reference files periodically and will make them available to the public in the SNAPPY GitHub repository. In addition, users may easily create their own reference files and haplogroup databases by following the format of each of these files. Note that tree\_structure.txt is formatted as “parent haplogroup-TAB-child haplogroup.” Please also note that custom Y-chromosome libraries must follow the exact names of the provided reference files.**

**Algorithm summary:** After being called, **SNAPPY searches for a .raw file with the correct prefix (controlled by parameter ‘infile’). If the .raw file is not available, SNAPPY calls plink to create the .raw file from the plink library determined by the ‘infile’ parameter. SNAPPY then** creates a set of reference dictionaries to track the relationships between various SNP identifiers, positions, ancestral and derived alleles, and associated haplogroups. These reference dictionaries are built from the set of reference files that are included in the program distribution.

Genotypes from all samples, read in from the .raw file, are then stored as a list of dictionaries, with each dictionary containing key-value pairs consisting of Y-chromosome positions (keys) and allele (values) for each sample. SNAPPY then cycles through each sample’s genotypes and each Y-chromosome haplogroup stored in the reference dictionaries, counting the number of haplogroup-informative alleles present in the sample. This number, when divided by the sample’s number of non-missing haplogroup-informative positions, is the haplogroup’s score for the given sample. To illustrate, consider a haplogroup that is defined by 5 SNPs, and an individual who has been genotyped at these 5 sites. If the individual is missing one genotype, and has the derived allele for three of the sites, and the ancestral allele at the fifth site, then the score is . Importantly, a particular haplogroup’s score uses alleles from both its own haplogroup-informative positions as well as all its ancestral haplogroups. If all informative positions are missing for a given haplogroup, the score for the haplogroup is set to zero. Additionally, if no informative alleles from a particular haplogroup’s two most recent ancestors are present, that haplogroup will not be considered for assignment to a sample (e.g., referenced node in Figure 1C would not be considered because its two most recent ancestors lack informative alleles in the sample). Each haplogroup is evaluated independently for every individual, and the scores are stored in a two-dimensional numpy array to allow for efficient storage and quick processing.

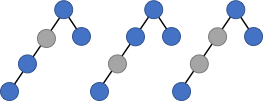


Figure 1 Possible haplogroup patterns. Blue indicates presence of informative alleles, gray indicates lack of informative alleles.

A

B

C

SNAPPY then makes haplogroup assignments to the deepest node with sufficient support by first considering all haplogroup nodes that have a score greater than a user-defined threshold (default=0.6, controlled by parameter ‘min\_hap\_score’) and no descendant haplogroups with scores greater than the threshold. SNAPPY makes the haplogroup assignment based on the haplogroup with the highest score, or may make the assignment to the deepest haplogroup with a score higher than a user-defined threshold (default=0.8, controlled by parameter ‘min\_deep\_score’), depending on the value of the min\_hap\_score and min\_deep\_score parameters. The values of both of these parameters can be adjusted at the command line at runtime if the user wishes to prioritize deeper haplogroup assignments vs. higher-scoring assignments.

#### **Dependencies:** SNAPPY is implemented in python2 (SNAPPY\_v0.2.1) and in python3 (SNAPPY\_v0.2.2) and makes use of the python modules ‘numpy’, ‘sys’, ‘os’, ‘os.path’, ‘re’, and ‘subprocess’. In addition, a plink (v1.9) executable must be listed in the user’s path as ‘plink’ for preprocessing steps.

### Parameters

The following table outlines user-controllable parameters that can be adjusted at run time:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Default Value | Description |
| infile | N/A, required | Prefix to plink library or .raw file to be used as input |
| out | chrY\_hgs | Prefix to .out and .all files generated by SNAPPY |
| min\_hap\_score | 0.6 | Minimum match score for a haplogroup to be considered for assignment |
| min\_deep\_score | 0.8 | Minimum score to switch from highest scoring haplogroup to the deepest haplogroup for assignment |

To adjust a parameter, append a double hyphen (--) followed immediately by the parameter name, a space, and the desired value for that parameter. Example:

python SNAPPY\_v123.py --infile plink\_prefix --min\_hap\_score 0.7

### Notes and Considerations:

* All reference files included in the current distribution of SNAPPY use positions from human genome version GRCh37. Genotype positions from other versions of the human genome may result in inaccurate results.
* Prior to running SNAPPY, it may be necessary to check for strand concordance with the Y-chromosome of GRCh37, and to flip and/or remove ambiguous sites and those whose variants correspond to genotyping from the non-reference strand.
* A key aspect of the SNAPPY’s success is the robust nature of the Y-chromosome tree and the inclusion of informative variants on the Multi-Ethnic Genotyping Array (MEGA). SNAPPY’s current implementation was designed and tested using genotyping data from the MEGA, which includes over 11,000 variants on the Y-chromosome. SNAPPY should readily apply to other arrays, but care should be taken to ensure that arrays have a sufficient number of loci that are included in the reference library.
* Genotyping by sequencing (GBS) is increasingly popular, and data generated through GBS is compatible with SNAPPY, provided that all sites passing quality filters are included in the output genotypes during variant calling (this can be accomplished, for example, using the --emit-all argument in GATK’s variant calling pipeline). Otherwise, haplogroup-informative sites where the reference sequence used in variant calling has a derived allele may not be included in the genotype file.

### Future Improvements

The SNAPPY team welcomes suggestions for improvements from the user community. SNAPPY developers plan to implement additional functionality into SNAPPY as the need for such functionality arises. To date, potential improvements include:

* Different scoring systems (ex: jaccard similarity coefficient, Kulczynski measure) to enable a more robust scoring system around haplogroup calling.
* Development for an automated method for downloading and integrating up-to-date data from ISOGG.

### Citation

If you use SNAPPY, please cite its manuscript in bioRXiv.