HaploGI – Haplotyping Given Inheritance

Version: 1.0.0
Last Modified: June 28, 2025
Author: Rafael A. Nafikov
Division of Medical Genetics, Department of Medicine, University of Washington

----

Contents

Introduction

HaploGI (Haplotyping Given Inheritance) is a C++ program for pedigree-based haplotyping of whole genome sequencing (WGS) data. It also identifies haplotype sharing among subjects in extended pedigrees.

Paper Citation

If you use **HaploGI** in your work, please cite:

Nafikov, R. A., Sohi, H., Nato Jr, A. Q., Horimoto, A. R., Bird, T. D., DeStefano, A., Blue, E. E., & Wijsman, E. M.
Variant prioritization by pedigree-based haplotyping. Submitted for publication to *Genetic Epidemiology*, 2025.

Software Citation

If you use **HaploGI** in your research, please also cite the following:

[package] Software

`DOI` `xx.xxx/zenodo.xxx`

```
@software{nafikov_2025_haplogi,
  author      = {Rafael A. Nafikov},
  title       = {HaploGI - Haplotyping Given Inheritance},
  version     = {1.0.0},
  year        = 2025,
  publisher   = {Zenodo},
  doi         = {xx.xxx/zenodo.xxx},
  url         = {https://doi.org/xx.xxx/zenodo.xxx}
}
```

This repository includes a `CITATION.cff` file.
On GitHub, click the **"Cite this repository"** button near the top to get citation details in various formats.

License

HaploGI is licensed under the GNU General Public License v3.0.
2025 Rafael A. Nafikov.

See LICENSE for full terms.

Software URL

Repository: https://github.com/RafPrograms/HaploGI

Files available for download: - `HaploGI.cpp` (source code) - `manual_HaploGI_v1.0.0.pdf` (PDF) (user manual) - `parameter_file_template.txt` (parameter file template) - `HaploGI_utility_scripts` (Python utility scripts) - `HaploGI_test_data.zip` (example dataset)

Build and Install

You can build HaploGI from source using CMake:

Requirements CMake 3.10

C++17-compliant compiler (e.g., g++ 7, clang 5)

Unix-like environment (Linux/macOS recommended)

Build Instructions

```
# Clone the repository (if not done yet)
git clone https://github.com/RafPrograms/HaploGI.git
cd HaploGI

# Create a separate build directory
mkdir build
cd build

# Generate Makefiles with CMake
cmake ..

# Compile the program
make
```

[rocket] Run from Build Directory After compiling, you can run HaploGI directly from the build directory:

```
./HaploGI -- [options] [parameter_file_path]
```

Optional: Install System-Wide

To install the compiled binary to your system path (default: /usr/local/bin):

```
sudo make install
```

This will allow you to run HaploGI from anywhere in your terminal.

[idea] Note: The binary is installed to the bin/ directory under your system's CMAKE_INSTALL_PREFIX (default: /usr/local/bin).

Custom Install Location You can specify an install prefix:

```
cmake -DCMAKE_INSTALL_PREFIX=/your/custom/path ..
make
sudo make install
```

Getting Started

Launch HaploGI using the following syntax:

```
./HaploGI -- [options] [parameter_file_path]
```

Run Options

Option	Description
`--haplotyping`	Pedigree-based haplotyping + core set of cases identification
`--haplosharing`	Evaluate haplotype sharing in predefined cases
`--full`	Combines both haplotyping and haplosharing

General Options

Option	Description
`--help`	Display help
`--version`	Show current version

Parameter File

See: `parameter_file_template.txt`

Required for all run types:

Entry	Description
1#	Pedigree file path
2#	SNV genomic positions file path
3#	SNV genotypes file path
4#	Linkage region boundaries (cM)
5#	Max LOD marker position (cM)
6#	Output directory

Required for `--haplotyping` and `--full`:

Entry	Description
7#	Linkage markers genomic positions file
8#	Meiosis indicators file
9#	Number of iterations in indicator file

Required for `--haplosharing`:

Entry	Description
10#	Haplotype sequences file
11#	Core set of cases file

Optional:

Entry	Description
12#	Seed number (default: 1234)

Input File Formats

All files must be **space-delimited**.

Required per Run Option

File	Required for
Pedigree file	All
SNV genomic positions	All
SNV genotypes	All
Linkage markers positions	`haplotyping`, `full`
Meiosis indicators	`haplotyping`, `full`
Haplotype sequences	`haplosharing`
Core cases	`haplosharing`

Example Input Files

Pedigree File Format:

```
subject father mother sex phenotype
********
101 0 0 1 0
102 0 0 2 0
201 101 102 1 0
202 101 102 2 0
2010 0 0 2 0
301 201 2010 1 0
302 201 2010 2 2
```

The pedigree file contains **five space-delimited columns** with the following information:

1. **Subject ID**

2. **Father ID**

3. **Mother ID**

4. **Sex**
   - `1` = Male

   - `2` = Female

   - `0` = Unsexed / unknown

5. **Phenotype**
   - `1` = Control

   - `2` = Case

   - `0` = No phenotype data

File Format Notes

- **Header lines (above the main data) are ignored** by the program if they appear **before a line starting with `*`**.

- You may include column headers before this marker.
- **IDs must not contain special characters** such as `#`, `*`, or `@`.

  Note: The `********` line acts as a marker—any content above this line is ignored during processing.

SNV Genomic Positions File Format:

```
1052701 3.767099
1052874 3.767696
1053095 3.768460
1053154 3.768664
```

This file contains genomic position data for each single nucleotide variant (SNV), with one SNV per line.

File Format

- **No header row**

- Each line contains **two space-delimited columns**:

1. **Base pair (bp) position** – The physical location of the SNV on the chromosome

2. **Genetic position in centimorgans (cM)** – The corresponding genetic map position

Note: Ensure the order of positions in this file matches the order of SNVs used in related genotype and haplotype files.

SNV Genotype File Format:

```
variant_position 302 302 303 303 306 306 307 307 402 402 403 403 404 404 406 406 407 407 408
16:10414 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
16:10638 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

The SNV genotype file contains variant genotype data for subjects with whole genome sequencing (WGS) data. Genotypes are encoded as:

- `0` = Missing

- `1` = Reference allele (REF)

- `2` = Alternative allele (ALT)

File Structure

- **First column**:
  Contains the SNV's genomic position in the format `chromosome:position` (e.g., `16:10414`).

- **Remaining columns**:
  Each subject is represented by **two consecutive columns**, one for each of their diploid genotype alleles.

- **Header row**:
  Lists subject IDs. Each subject ID appears **twice**, corresponding to their two genotype alleles.

  Note: Ensure that subject IDs are consistent across files and that each subject has exactly two columns representing diploid genotype data.

[tools] Use prepare_genotype_file.py to generate an SNV genotype file from a VCF file for use in HaploGI runs.

Linkage Markers Genomic Positions File Format:

```
0.219846
1.134855
1.793034
```

This file contains the genetic positions (in centimorgans, cM) of linkage markers used to compute inheritance vectors with the **Morgan package**.

File Format

- **No header row**
- Each line contains a **single centimorgan (cM) position** for one linkage marker
- Markers are listed in the order expected by downstream analysis tools

  Note: Ensure that the number and order of cM positions match the corresponding linkage marker set used in the analysis pipeline.

Meiosis Indicators File Format: The **meiosis indicators file** is generated by the `gl_auto` program from the Morgan package.
It follows the same format as described in the official Morgan package manual.

  Note: This file encodes inheritance information and is used in downstream linkage and haplotype analyses.

[tools] Use dicrease_number_of_MI_iterations.py to generate a meiosis indicators file with a reduced number of iterations (recommended: 1000) to ease the computational burden on HaploGI.

Haplotype Sequences File Format:

```
16:10414-23730 302_0 11111111112111111211
16:10414-23730 302_1 11112111222111111112
16:10414-23730 303_0 11111111222111111112
16:10414-23730 303_1 11111111222111111112
16:10414-23730 306_0 11112111222111111112
16:10414-23730 306_1 11111112112112111121
```

The haplotype sequences file is generated by **HaploGI** using either the `--haplotyping` or `--full` run options.

This file contains **three columns with no header**:

1. **Genomic Range**
   A string representing the chromosome and variant range in the format:
   `chr:start-end`

- `chr`: Chromosome number

- `start` and `end`: Positions of the first and last genomic variants in the haplotype

2. **Subject ID and Chromosome**
   The subject identifier followed by an underscore and a digit:
   - `_0`: Maternal chromosome

   - `_1`: Paternal chromosome

3. **Haplotype Sequence**
   A string of digits representing the sequence of genomic variants for the given region.

   Note: This file does not include a header row. Be sure to account for that when parsing the file programmatically.

Core Set of Cases File Format:

302 306 403 408 411 501 504 506 511 512 513 516

This file contains a list of **case subject IDs**, separated by spaces, all on a **single line**.

- **No header row**
- IDs must match those used in other input files (e.g., pedigree, genotype, haplotype files)

HaploGI uses this set of cases to **check for the existence of haplotype sharing** among them.

Output File Formats

File	Generated by
Log	All
Haplotype sequences	`haplotyping`, `full`
Core cases	`haplotyping`, `full`
Allele inconsistencies	`haplotyping`, `full`
Shared haplotypes	`haplosharing`, `full`
Haplotype sharing patterns	`haplosharing`, `full`

Example Output Files

Haplotype Sequences File – see Haplotype Sequences File Format
[tools] Use create_phased_vcf.py to convert phased whole-genome sequencing

(WGS) data generated by HaploGI into VCF format, enabling easier downstream analysis.

Core Set of Cases – see Core Set of Cases File Format

————————————————

Inconsistencies of Allele to FGL Assignments File Format:

```
bp_position
1445745
1455891
1458974
```

This file lists variants for which inconsistencies between alleles and Founder Genome Labels (FGL) were detected.

File Format

- The file **includes a header row**.
- Each subsequent line contains the **base pair position** of a variant with detected inconsistency.
- One base pair position per line.

Notes

- This file is intended for **exploratory purposes only**.
- It is **not required** for haplotyping or determining haplotype sharing.

————————————————

Shared Haplotype Sequence File Format:

```
16:1052701-1055604 22222212211222211212
16:1127696-1132994 11111222221112211212
16:1506499-1511322 11111121111111111111
16:1511338-1513919 11111111111111111111
16:1514349-1520077 11111111111111111111
```

This file contains haplotype sequences shared within genomic windows among cases listed in the core set of cases file.

File Format

- The file has **two columns** and **no header**.
- The **first column** specifies the genomic window location, formatted as:
  `chromosome_number:first_variant_bp-last_variant_bp`
  (e.g., `16:1052701-1055604`)

- The **second column** contains the haplotype sequence corresponding to that genomic window.

[tools] Use risk_haplotype_sequence_vcf.py to convert risk haplotype sequences generated by HaploGI into VCF format for convenient downstream analysis.

Haplotype Sharing Patterns File Format:

```
Genomic_window 302 302 303 303 306 306 307 307 402 402 403 403 404 404 406 406 407 407 408 4
1 0 1 0 2 1 0 2 0 0 0 0 1 0 2 0 0 0 0 1 0 2 0 1 1 0 0 0 2 0 2 0 2 0 1 0 0 0 2 0 1 0 2 1 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

This file contains information about haplotype sharing across all genomic windows evaluated.

File Format
- The file **includes a header row**.
- The **first column** contains genomic window numbers.
- Each subject with WGS data is represented by **two consecutive columns**:
  - One for the maternal chromosome
  - One for the paternal chromosome
- Entries indicate the presence of a shared haplotype:
  - `1` = Shared haplotype present in a **case**
  - `2` = Shared haplotype present in a **control**
  - `0` = No shared haplotype present.

Usage

You can use the provided Python utility script `plot_haplotype_sharing.py` to generate these plots.

[tools] Use plot_haplotype_sharing.py to visualize haplotype sharing data generated by HaploGI. This script helps identify the presence and boundaries of risk haplotypes. The resulting plots also provide a broader overview of haplotype sharing, supporting more informed decision-making during data analysis.

[tools] HaploGI Utility Scripts

A number of Python utility scripts are available to assist with preparing HaploGI input files, processing output data, and visualizing results, at: `HaploGI_utility_scripts`.

Other Python utility scripts that have not been introduced yet in this manual are:

[tools] risk_alleles_variants.py – Identifies and outputs variants whose alleles are uniquely present on the risk haplotype. The resulting file also includes associated metadata extracted from the input VCF file.

[tools] create_genomic_windows.py – Generates a genomic windows file from an SNV genomic positions file used in HaploGI runs. This facilitates easier cross-referencing of data to specific genomic regions.

Data Example

A full example with three run configurations is available in `data_example`.

Support

For questions, bug reports, or suggestions, please contact:
nrafscience@gmail.com
GitHub Issues

References

1. Nafikov et al., (2025). Variant prioritization by pedigree-based haplotyping. *Genetic Epidemiology*.
2. Nafikov et al., (2018). Dealing with Admixture in Caribbean Hispanic Families. *Genetic Epidemiology*. DOI:10.1002/gepi.22133
3. Tong & Thompson (2007). Multilocus lod scores. *Human Heredity*. DOI:10.1159/000109731

Web Resources

- **HaploGI**: https://github.com/RafPrograms
- **Morgan Package**: Morgan site
- **1000 Genomes Project**: https://www.internationalgenome.org