Manual: Hapl-o-Mat via Hapl-o-MatGUI

Please also see the README.

Hapl-o-Mat is a software for HLA haplotype inference coded in C++. Besides estimating haplotype frequencies via an expectation-maximization algorithm, it is capable of processing HLA genotype population data. This includes translation of alleles between various typing resolutions and resolving allelic and genotypic ambiguities.

For more information refer to our publication on Hapl-o-Mat:

Schaefer C, Schmidt AH, Sauter J. Hapl-o-Mat: open-source software for HLA haplotype frequency estimation from ambiguous and heterogeneous data. BMC Bioinformatics. 2017;18(1):284. Published 2017 May 30. doi:10.1186/s12859-017-1692-y

If you use Hapl-o-Mat for your research, please cite preferably the journal article.

System requirements and installation

After successful installation, Hapl-o-Mat can be executed both from command line or in the Hapl-o-MatGUI.

Installation on Linux OS:

In order to use Hapl-o-Mat with the GUI on Linux, please install both, Hapl-o-Mat C++ source code (1) and Hapl-o-MatGUI python3 source code (2).

Installation on Windows OS:

For use Hapl-o-Mat with the GUI on Windows please install the Hapl-o-Mat Windows binary version (3). This version contains both, Hapl-o-Mat and the GUI and has no further system requirements.

Alternatively, you can install and use Hapl-o-Mat C++ source code (1) and Hapl-o-MatGUI python3 source code (2) on you Windows device.

(1) **Hapl-o-Mat C++ source code:** For system requirements and installation of Hapl-o-Mat see the respective tutorials on GitHub (https://github.com/DKMS/Hapl-o-Mat), e.g. "detailedGettingStartedWindows.pdf" and "detailedGettingStartedLinux.pdf".

(2) **Hapl-o-MatGUI python3 source code:** For system requirements and installation of Hapl-o-MatGUI see the tutorial "README.md" on GitHub (https://github.com/DKMS/Hapl-o-Mat_GUI. In order to use Hapl-o-Mat via GUI with GUI python3 source code, both programs have to be installed on your device.

(3) **Hapl-o-Mat Windows binary version**: For installation of Hapl-o-MatGUI see the tutorial "README.md" on GitHub https://github.com/DKMS/Hapl-o-Mat_WinBin. The Hapl-o-Mat_WinBin version contains both, Hapl-o-Mat and the GUI.

Execution of Hapl-o-Mat via GUI

1. Starting the GUI

Python source code: Enter the directory Hapl-o-MatGUI/GUIsrc in your command line interpreter and start the program via "python main.py". Please note that you require python3.x to run the GUI.

Windows binary version: During installation, Hapl-o-MatGUI will be added to your start menu and can be started from there. Alternatively, you can execute the Hapl-o-MatGUI.exe file in the directory you chose during Hapl-o-MatGUI installation.

2. Set path to 'haplomat' folder

To enable the GUI to interact with Hapl-o-Mat, the path to the folder containing the executable 'haplomat' must be selected. Button [...] opens a file dialogue. After the selection, status and actuality of the IPD-IMGT/HLA data is automatically checked and indicated in the "Update IPD-IMGT/HLA data" frame (see (2.)).

3. Update IPD-IMGT/HLA data

Hapl-o-Mat haplotype frequency inference relies on information about the HLA nomenclature provided by external files. After the first download of Hapl-o-Mat the initial data update is mandatory. As the HLA nomenclature is constantly evolving, e.g. by inclusion of new alleles or new multiple allele codes, it is important to update this data regularly.

Hapl-o-Mat relies on the following files in the folder "Hapl-o-Mat/data" will be created by the updating process:

File name	Description
AllAllelesExpanded.txt	A list of relevant existing HLA alleles with their enclosed more-digit typing resolutions
AlleleList.txt	If your input data in GLS format includes a missing single-locus genotype, it can be replaced by combining all alleles of the same locus from this file. You only must create it in this case.
Ambiguity.txt	Data for the ambiguity filter
LargeG.txt	A list of G-groups with their enclosed alleles in 8-digit resolution
MultipleAlleleCodes.txt	A list of multiple allele codes and their translation to alleles in 4-digit resolution
P.txt	A list of P-groups with their enclosed alleles in 8-digit resolution
Smallg.txt	A list of g-groups with their enclosed alleles in 8-digit resolution

Starting the GUI and selecting the Hapl-o-Mat folder automatically triggers a status and actuality check of the data files. The date of the last update as well as the loci available for haplotype estimation are indicated in the "Update IPD-IMGT/HLA data" frame. If the data is older than three months a warning sign and an update recommendation will be indicated. Otherwise, a green tick will indicate that the updating step can be skipped.

The [Update] button starts the process of downloading and automatic data processing which may take a few minutes. A working internet connection is required for the update process.

Troubleshooting 1: If the automatic data updating process fails, e.g. due to a connection time out, a proxy or a firewall issue, there is a possibility to download the necessary data files manually, place

them in the "Hapl-o-Mat/prepareData" directory and then initiate the data processing via the GUI [Update] button.

The four files and their respective locations are the following:

1. hla_nom_p.txt: http://hla.alleles.org/wmda/hla_nom_p.txt
2. hla_nom_g.txt: http://hla.alleles.org/wmda/hla_nom_g.txt
3. alpha.v3.zip: https://bioinformatics.bethematchclinical.org/HLA/alpha.v3.zip
4. hla_ambigs.xml.zip: https://github.com/jrob119/IMGTHLA/raw/Latest/xml/hla_ambigs.xml.zip

Please make sure to save the four files on your computer and only place a copy in "Hapl-o-Mat/prepareData", because the files will be deleted automatically after processing.

For detailed instructions on how to download the necessary data files manually, please see the tutorial "detailedExplanationPrepareData.doc" in the "Hapl-o-Mat/ textsForGettingStarted" directory.

Troubleshooting 2: After an aborted update via the [Update] button, download files may remain undeleted in the Hapl-o-Mat subdirectory "prepareData". This will impede the download of updated data files and is indicated by a warning message prior to the start of the download process. In this case please abort the download and remove all following files from the "prepareData" directory:

- alpha.v3.zip
- alpha.v3.txt
- hla_ambigs.xml.zip
- hla_ambigs.xml
- hla_nom_g.txt
- hla_nom_p.txt
- Smallg.txt
- LargeG.txt
- P.txt
- MultipleAlleleCodes.txt
- allAlleles.txt
- AllAllelesExpanded.txt
- OneElementG.txt

4. Set run parameters

Run parameters can be either set manually or loaded from an existing valid parameter file.

The [Parameters] button opens an input mask:

Load parameters: On the input mask, the [Load parameters] button offers the choice of an existing parameter file via a file dialogue. Parameters are then loaded into the respective fields of the mask and can be edited and saved for the run.

Set parameters: In the "Set or change parameters" frame, all parameters can be set manually and saved for the run.

It is recommended to choose a unique Run-ID for each Hapl-o-Mat run, because all results files as well as a copy of the parameters file will be saved under specification of the Run-ID in the chosen results folder.

4.1 Input Genotype Data

Hapl-o-Mat infers haplotypes from population genotype data. It supports different formats of recording genotype data. To use Hapl-o-Mat in the GUI, your data should be in one of the following data formats:

Data format	Description
MAC	**M**ultiple **A**llele **C**odes: ambiguities are encoded by multiple allele codes (MAC). Except for the first line, input files hold an individual's identification number and genotype per line. Genotypes are saved allele by allele without locus name. Identification number and alleles are TAB-separated. The first line of the file is a header line indicating the name of the first column and the loci of the other columns. Same loci must be placed next to each other. For an example refer to "Hapl-o-Mat/examplePopulations/populationData_a.dat".
GLSC	**G**enotype **L**ist **S**trings **C**olumn-wise: genotypes with or without ambiguities are saved by genotype list strings (GLS). Input files hold an individual's identification number and genotype per line. Identification number and single-locus genotypes are TAB-separated. For an example refer to "Hapl-o-Mat/examplePopulations/populationData_b.dat".

4.2 Parameters

Each input format for genotype data requires a different set of parameters. The parameters are saved in the corresponding files "parametersMAC" and "parametersGLSC".All input formats have the following parameters in common:

Parameter	Description
FILENAME_INPUT	The file name of the input population data.
FOLDER_RESULTS	The folder in which the results files and a copy of the parameter file of the run are to be be saved.
RunID	The individual run-ID for the Hapl-o-Mat run.
LOCI_AND_RESOLUTIONS	Loci included into analysis and desired typing resolution per locus. Supported typing resolutions and their abbreviations are g-groups (g), P-groups (P), G-groups (G), 2-digit fields (1field), 4-digit fields (2field), 6-digit fields (3field), and 8-digit fields (4field). In order to exclude loci that are available in the input file, please choose the option "ignore locus". Alleles are not translated via the option asItIs (applying the ambiguity filter includes an intrinsic translation to G-groups).
MINIMAL_FREQUENCY_GENOTYPES	Genotypes which split into more genotypes than the inverse of this number are discarded from analysis.

DO_AMBIGUITYFILTER	Takes values "true" and "false". The option "true" activates the ambiguity filter.
EXPAND_LINES_AMBIGUITYFILTER	Takes values "true" and "false". If set to "true", matching lines with additional genotype pairs in the ambiguity filter are considered.
WRITE_GENOTYPES	Takes values "true" and "false". If set to "true", resolved genotypes will be written to a file under the path specified by FILENAME_GENOTYPES (see below).
INITIALIZATION_HAPLOTYPEFREQUENCIES	Initialization routine for haplotype frequencies. It takes the following values: - "equal": All haplotype frequencies are initialized with the same frequency, the inverse number of haplotypes. - "numberOccurrence": Haplotype frequencies are initialized according to the initial number of occurrence of haplotypes. - "random": Haplotype frequencies are initialized randomly. - "perturbation": Haplotype frequencies are initialized as in numberOccurrence and then randomly modified by a small (<10%) positive or negative offset.
EPSILON	Value for the stopping criterion, i.e. the maximal change between consecutive haplotype frequency estimations is smaller than the assigned value. Values larger than 1/(2*n), with n being the number of genotypes in the input file, are not valid.
CUT_HAPLOTYPEFREQUENCIES	Estimated haplotype frequencies smaller than this value are removed from the output.
RENORMALIZE_HAPLOTYPEFREQUENCIES	Takes values "true" and "false". If "true", normalize estimated haplotype frequencies to sum to one. Within machine precision, this becomes necessary, if estimated haplotypes are removed, e.g. via the option CUT_HAPLOTYPEFREQUENCIES.
SEED	Set the seed of the used pseudo random number generator. If set to "0", the seed is initialized by the system time. Valid entries are integer values between 0 and 1,000,000,000.

The following parameters are generated automatically upon entry of results folder and RunID:

Parameter	**Description**
FILENAME_HAPLOTYPES	Name of the file which temporarily saves haplotype names
FILENAME_GENOTYPES	Name of the file which saves resolved genotypes
FILENAME_HAPLOTYPEFREQUENCIES	Name of the file which saves haplotypes and estimated haplotype frequencies
FILENAME_EPSILON_LOGL	Name of the file which saves stopping criterion and log-likelihood per iteration

5 Run Hapl-o-Mat

The [Run] button starts the Hapl-o-Mat run. Hapl-o-Mat terminal output is displayed. Simultaneously with the EM algorithm the current course of the epsilon value is displayed in the right plot of the "results" window. When the run is finished, the terminal output prints "Finished!" and displays the time stamp. Results are displayed in the "Results" frame.

All terminal output of Hapl-o-Mat is saved as "RunID_Log.dat" in the chosen results folder.

6 Results

When the Hapl-o-Mat run is finished, result files are saved and results are automatically displayed in the "Results" frame.

6.1 Result files

All results files will be found in the specified results folder with a tag of the chosen RunID.

Results files are:

File type	File name	Description
Parameters	*RunID*_parametersGLSC or *RunID*_parametersMAC	A copy of the Hapl-o-Mat parameter file created in the "Set parameters" step. The extension changes between "GLSC" or "MAC" depending on the format of the input file. These parameter files can be loaded and edited under "Set parameters" for subsequent runs.
Haplotype frequencies	*RunID*_htf.dat	Complete list of haplotypes and their frequencies as estimated from the input genotypes.
Log	*RunID*_Log.dat	Copy of Hapl-o-Mat terminal output.
Epsilon log	*RunID*_epsilon.dat	Log of the changes in stopping criterion Epsilon and the log-likelihood during the iterating expectation and maximization steps. The first column is the stopping criterion Epsilon and the second one the not normalized log-likelihood.
Genotypes	*RunID*_genotypes.dat	Information on how the input genotypes were resolved for haplotype estimations. The first column corresponds to the individual's identification number. The second column indicates how ambiguities per single-locus genotypes have been resolved. If no ambiguity occurred or no additional genotypes are formed, the type is N. If an ambiguity occurred and was resolved via building all possible allele combinations, the type is I. Activating the ambiguity filter gives additional types: A, if one matching line in the ambiguity file was found, and M if multiple matching lines were found. The third column gives the frequency of the genotype and the fourth column the genotype itself. The genotype is saved in the GLS format. If an individual's genotype splits into a set of genotypes,

	each genotype is written to one line starting with the same identification number. The corresponding frequencies become non-integer and sum to one.

6.2 Result display

The results display mainly consists of a list of the haplotypes and their estimated frequencies, the plotted haplotype frequencies and the plotted epsilon values of the EM algorithm.

The top line "Statistics" of the "Results" frame provides information on

- the **total number of haplotypes** in the results file,
- the **number of genotypes (*n*)** that contributed to the results (e.g. genotypes which were discarded from analysis because of excess ambiguity are not counted),
- the critical value **1/(2*_n_)** which specifies the theoretically smallest possible frequency of a haplotype represented once in the sample,
- and the **sum of the cut haplotype frequencies** (which will be 1 if none of the haplotypes were cut or if the frequencies were normalized).

The line below ("Display") offers different options to limit the number of haplotypes displayed in the list view and the frequency plot. The options are

- **Top**: display of the Top *x* haplotypes (*x* can be set manually)
- **All**: all haplotypes in the Hapl-o-Mat output as defined by parameter CUT_HAPLOTYPEFREQUENCIES.
- **All haplotypes with frequency ≥ 1/(2*_n_)**: with n = the number of genotypes that contributed to the results, as indicated in the "Statistics" line.
- **All haplotypes with cumulated frequency ≤ *y***: *y* can be set manually

Values from the haplotype frequencies list can be copied (STRG+A, STRG+C) to clipboard.

The axes of the two plots can be scaled logarithmically by clicking the options on the bars below the plots.

7 Unsolved Issues

7.1 Plot scaling issue: Plot scaling adjustment is incorrect when switching between monitors with different DPI settings. This issue is a known PyQtGraph bug that has not been resolved yet (see e.g.: https://github.com/pyqtgraph/pyqtgraph/issues/756).