Hapl-o-Mat - Getting Started

Please also see the README.

Hapl-o-Mat

Hapl-o-Mat is 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. Both common formats for recording HLA genotypes, multiple allele (NMDP) codes and genotype list strings, are supported.

This guide explains how to use Hapl-o-Mat under Linux. For more information refer to our publications on Hapl-o-Mat:

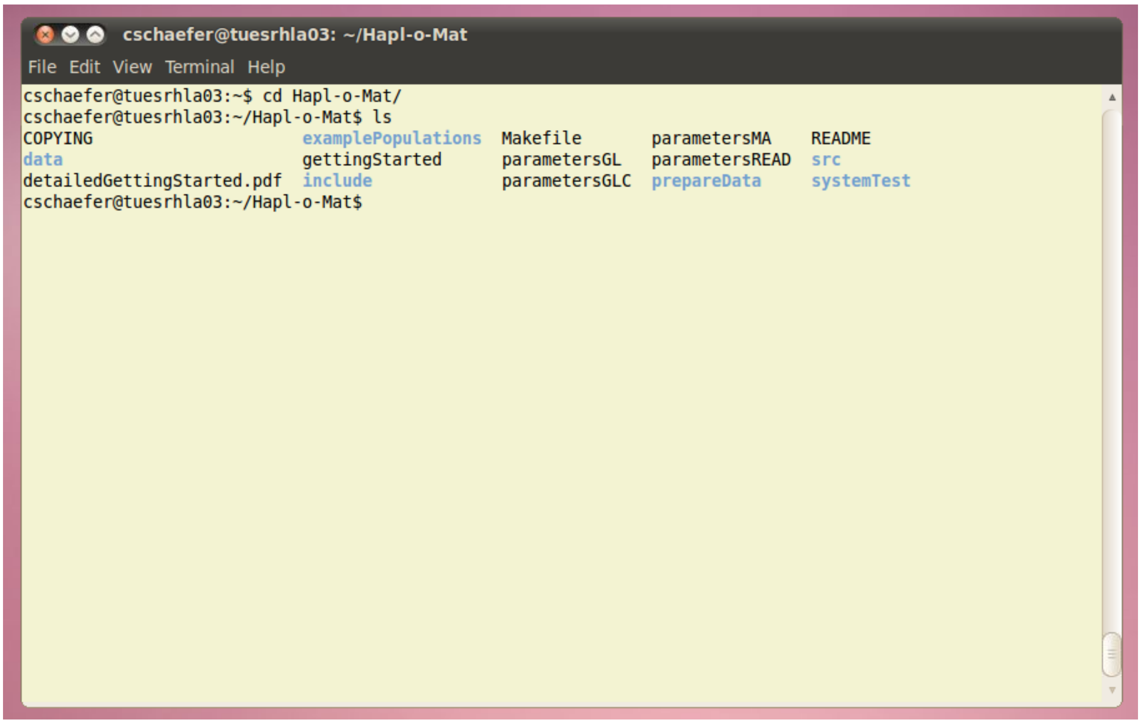
Journal article to come

C. Schaefer, A.H. Schmidt, J. Sauter: Hapl-O-mat: A Versatile Software for Haplotype Frequency Estimation. HLA (2016), 87, 236-320

Getting Started

This guide is an introduction on how to use Hapl-o-Mat. In order to follow this guide, you need a Linux system and a C++ compiler supporting C++11. In this tutorial, we use Ubuntu 14.04.4 LTS and GNU compiler collection (GCC) version 4.8.4. If you are a seasoned Linux-User, feel free to refer to the shorter version of this guide, gettingStarted.

After successfully downloading Hapl-o-Mat, start a terminal and browse to the location where Hapl-o-Mat is stored. Enter the folder Hapl-o-Mat by typing “cd Hapl-o-Mat”. Check what is inside by typing “ls”. You should see the following:



cschaefer@tuesrhla03: ~/Hapl-o-Mat
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~$ cd Hapl-o-Mat/
cschaefer@tuesrhla03:~/Hapl-o-Mat$ ls
COPYING                    examplePopulations  Makefile        parametersMA    README
data                       gettingStarted      parametersGL    parametersREAD  src
detailedGettingStarted.pdf include             parametersGLC   prepareData     systemTest
cschaefer@tuesrhla03:~/Hapl-o-Mat$
```

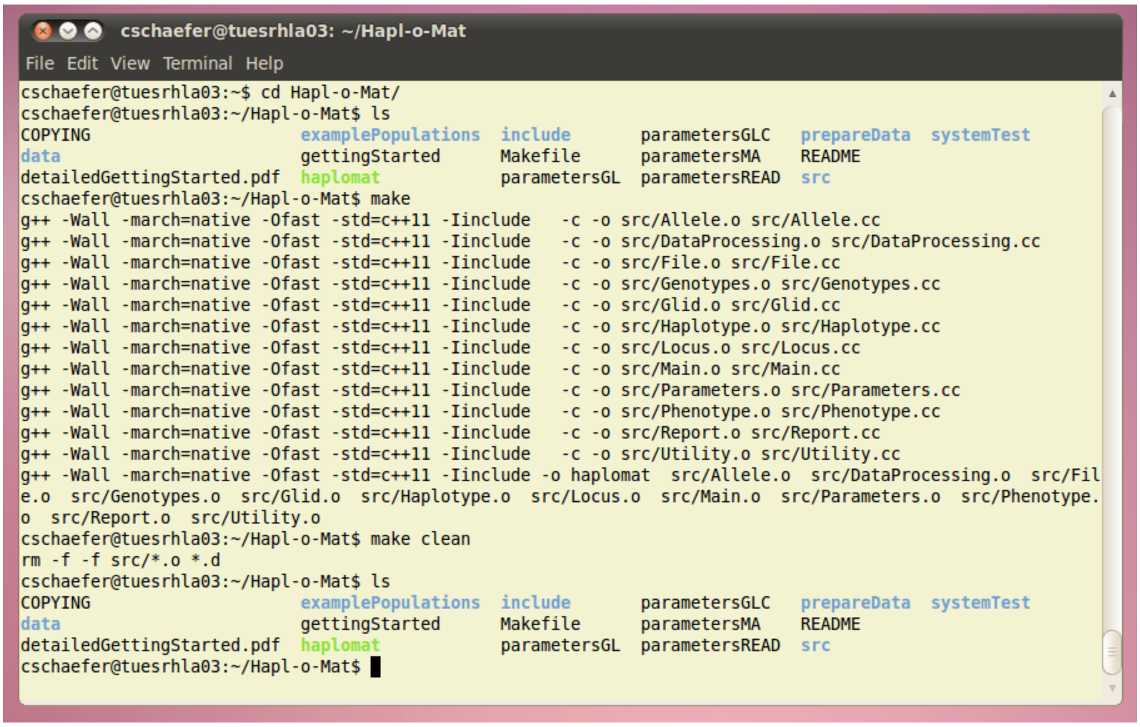
You see folders in blue ink and files in black ink. What we have is:

File name	Description
COPYING	The GNU General Public License.
data	Here goes the information on the HLA nomenclature required by Hapl-o-Mat. We are going to build it in section Data Preparation
detailedGettingStarted	This guide
examplePopulations	Some genotype population data we are going to work with in the section Tutorials.
gettingStarted	A shorter form of this tutorial
include	A part of Hapl-o-Mat's source code. If you do not want to change code, do not touch it
Makefile	Instructions for building Hapl-o-Mat. You might need to adapt ii, if you use another compiler than GCC
parametersGL, parametersGLC, parametersMA, parametersMA	Parameter files for Hapl-O-mat. We are going to discuss this in section Parameters
prepareData	Here is everything to create the data required by Hapl-o-Mat
README	Read me
src	A part of Hapl-o-Mat's source code. If you do not want to change code, do not touch it
systemTest	Run the system test after changing code to check, if you broke something.

To estimate haplotype frequencies we only need to consider the folders data, prepareData and the files Makefile, parametersGL, parametersGLC, parametersMA, and parametersREAD. To finish this tutorial we need the folder examplePopluations, too.

Install Hapl-o-Mat

We compile Hapl-o-Mat with GCC using a Makefile. Just type “make” to create the executable “haplomat” and “make clean” to clean up. Type again “ls” to find a new file, haplomat (indicated in green in the figure below).


cschaefer@tuesrhla03: ~/Hapl-o-Mat
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~$ cd Hapl-o-Mat/
cschaefer@tuesrhla03:~/Hapl-o-Mat$ ls
COPYING                  examplePopulations  include       parametersGLC   prepareData  systemTest
data                     gettingStarted      Makefile      parametersMA    README
detailedGettingStarted.pdf  haplomat         parametersGL  parametersREAD  src
cschaefer@tuesrhla03:~/Hapl-o-Mat$ make
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Allele.o src/Allele.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/DataProcessing.o src/DataProcessing.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/File.o src/File.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Genotypes.o src/Genotypes.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Glid.o src/Glid.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Haplotype.o src/Haplotype.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Locus.o src/Locus.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Main.o src/Main.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Parameters.o src/Parameters.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Phenotype.o src/Phenotype.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Report.o src/Report.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Utility.o src/Utility.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude -o haplomat  src/Allele.o  src/DataProcessing.o  src/Fil
e.o  src/Genotypes.o  src/Glid.o  src/Haplotype.o  src/Locus.o  src/Main.o  src/Parameters.o  src/Phenotype.
o  src/Report.o  src/Utility.o
cschaefer@tuesrhla03:~/Hapl-o-Mat$ make clean
rm -f -f src/*.o *.d
cschaefer@tuesrhla03:~/Hapl-o-Mat$ ls
COPYING                  examplePopulations  include       parametersGLC   prepareData  systemTest
data                     gettingStarted      Makefile      parametersMA    README
detailedGettingStarted.pdf  haplomat         parametersGL  parametersREAD  src
cschaefer@tuesrhla03:~/Hapl-o-Mat$ █
```

Data Preparation

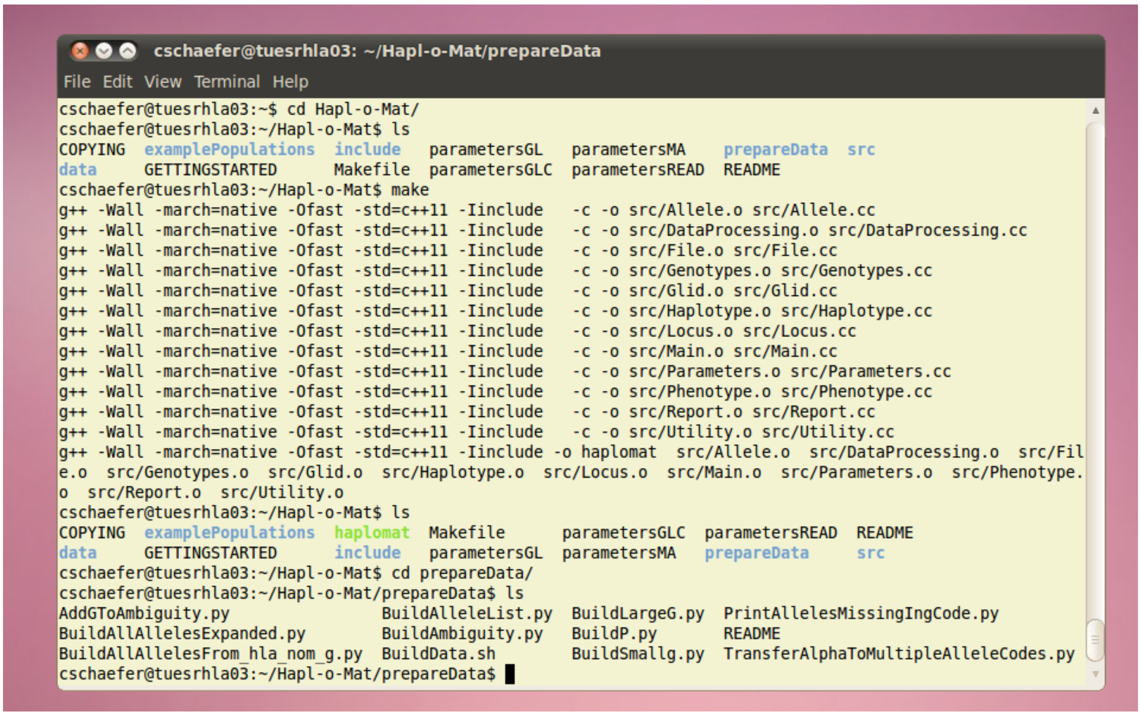
Hapl-o-Mat relies on information on the HLA nomenclature. This information is provided by data files, which are placed in the folder “data”. As the HLA nomenclature evolves over time, e.g. by finding new alleles or adding new NMDP codes, it is important to update data from time to time. Hapl-o-Mat relies on the following files, which must be placed in the folder “Hapl-o-Mat/data”:

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 GL format includes a missing single-locus genotype, it can be replaced by combining all alleles of the same locus from this file
Ambiguity.txt	Data basis 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
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

In the following we are going to create these data files.

Download Data

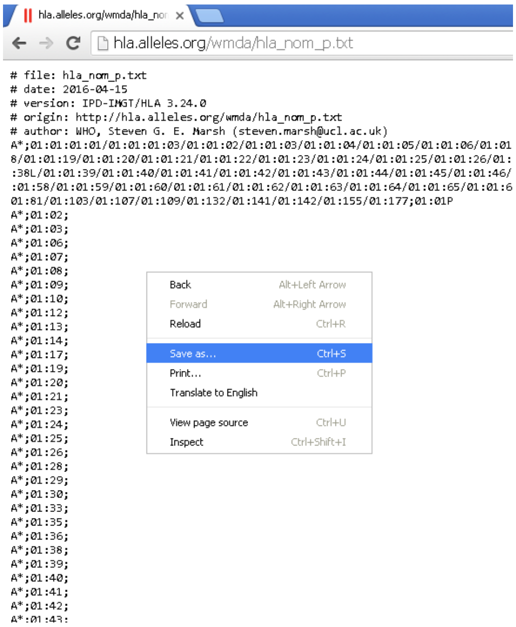
Go to the folder “prepareData” by typing “cd prepareData” and check its content by typing “ls”.


cschaefer@tuesrhla03: ~/Hapl-o-Mat/prepareData
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~$ cd Hapl-o-Mat/
cschaefer@tuesrhla03:~/Hapl-o-Mat$ ls
COPYING  examplePopulations  include   parametersGL   parametersMA    prepareData  src
data     GETTINGSTARTED      Makefile  parametersGLC  parametersREAD  README
cschaefer@tuesrhla03:~/Hapl-o-Mat$ make
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Allele.o src/Allele.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/DataProcessing.o src/DataProcessing.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/File.o src/File.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Genotypes.o src/Genotypes.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Glid.o src/Glid.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Haplotype.o src/Haplotype.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Locus.o src/Locus.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Main.o src/Main.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Parameters.o src/Parameters.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Phenotype.o src/Phenotype.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Report.o src/Report.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude   -c -o src/Utility.o src/Utility.cc
g++ -Wall -march=native -Ofast -std=c++11 -Iinclude -o haplomat  src/Allele.o  src/DataProcessing.o  src/Fil
e.o  src/Genotypes.o  src/Glid.o  src/Haplotype.o  src/Locus.o  src/Main.o  src/Parameters.o  src/Phenotype.
o  src/Report.o  src/Utility.o
cschaefer@tuesrhla03:~/Hapl-o-Mat$ ls
COPYING  examplePopulations  haplomat  Makefile      parametersGLC  parametersREAD  README
data     GETTINGSTARTED      include   parametersGL  parametersMA   prepareData     src
cschaefer@tuesrhla03:~/Hapl-o-Mat$ cd prepareData/
cschaefer@tuesrhla03:~/Hapl-o-Mat/prepareData$ ls
AddGToAmbiguity.py                 BuildAlleleList.py  BuildLargeG.py  PrintAllelesMissingIngCode.py
BuildAllAllelesExpanded.py         BuildAmbiguity.py   BuildP.py       README
BuildAllAllelesFrom_hla_nom_g.py   BuildData.sh        BuildSmallg.py  TransferAlphaToMultipleAlleleCodes.py
cschaefer@tuesrhla03:~/Hapl-o-Mat/prepareData$ 
```

You find a number of Python scripts (files ending with .py) which we are going to use to produce the data. But first we need some information from the web. Save all files in one folder, e.g. “InputData”.

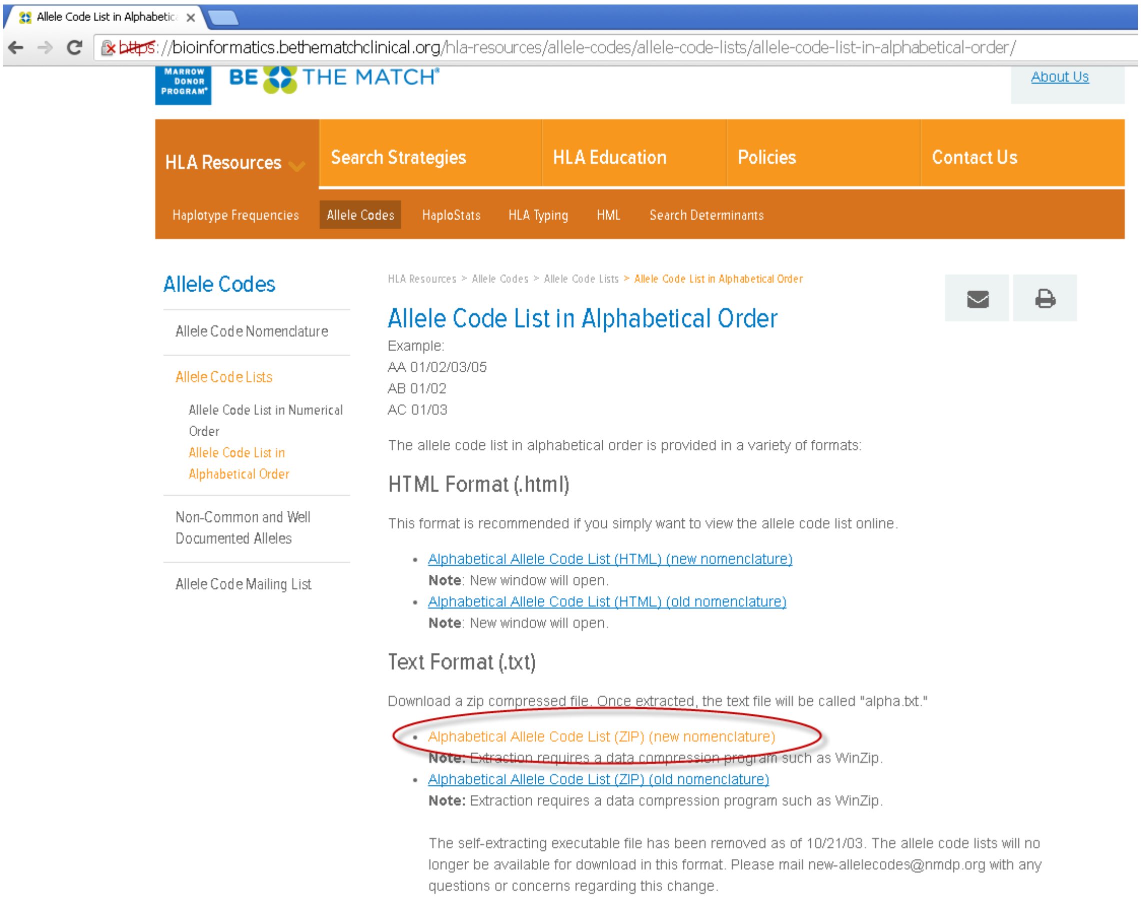
1) Go to the website http://hla.alleles.org/wmda/hla_nom_p.txt and save the file hla_nom_p.txt by right-clicking and choosing “Save as…”



```
# file: hla_nom_p.txt
# date: 2016-04-15
# version: IPD-IMGT/HLA 3.24.0
# origin: http://hla.alleles.org/wmda/hla_nom_p.txt
# author: WHO, Steven G. E. Marsh (steven.marsh@ucl.ac.uk)
A*;01:01:01:01/01:01:01:03/01:01:02/01:01:03/01:01:04/01:01:05/01:01:06/01:01
8/01:01:19/01:01:20/01:01:21/01:01:22/01:01:23/01:01:24/01:01:25/01:01:26/01:
:38L/01:01:39/01:01:40/01:01:41/01:01:42/01:01:43/01:01:44/01:01:45/01:01:46/
:01:58/01:01:59/01:01:60/01:01:61/01:01:62/01:01:63/01:01:64/01:01:65/01:01:6
01:81/01:103/01:107/01:109/01:132/01:141/01:142/01:155/01:177;01:01P
A*;01:02;
A*;01:03;
A*;01:06;
A*;01:07;
A*;01:08;
A*;01:09;
A*;01:10;
A*;01:12;
A*;01:13;
A*;01:14;
A*;01:17;
A*;01:19;
A*;01:20;
A*;01:21;
A*;01:23;
A*;01:24;
A*;01:25;
A*;01:26;
A*;01:28;
A*;01:29;
A*;01:30;
A*;01:33;
A*;01:35;
A*;01:36;
A*;01:38;
A*;01:39;
A*;01:40;
A*;01:41;
A*;01:42;
A*;01:43;
```

2) Go to the website http://hla.alleles.org/wmda/hla_nom_g.txt and save the file hla_nom_g.txt (same as in 1))

3) Go to the website https://bioinformatics.bethematchclinical.org/HLA-Resources/Allele-Codes/Allele-Code-Lists/Allele-Code-List-in-Alphabetical-Order/. Click on "Alphabetical Allele Code List (ZIP) (new nomenclature)" and save alpha.v3.zip.



Allele Code List in Alphabetic
~~https~~://bioinformatics.bethematchclinical.org/hla-resources/allele-codes/allele-code-lists/allele-code-list-in-alphabetical-order/
MARROW DONOR PROGRAM
BE THE MATCH
About Us
HLA Resources
Search Strategies
HLA Education
Policies
Contact Us
Haplotype Frequencies
Allele Codes
HaploStats
HLA Typing
HML
Search Determinants
Allele Codes
Allele Code Nomenclature
Allele Code Lists
Allele Code List in Numerical Order
Allele Code List in Alphabetical Order
Non-Common and Well Documented Alleles
Allele Code Mailing List
HLA Resources > Allele Codes > Allele Code Lists > Allele Code List in Alphabetical Order

Allele Code List in Alphabetical Order

Example:
AA 01/02/03/05
AB 01/02
AC 01/03

The allele code list in alphabetical order is provided in a variety of formats:

HTML Format (.html)

This format is recommended if you simply want to view the allele code list online.

- Alphabetical Allele Code List (HTML) (new nomenclature)
  Note: New window will open.
- Alphabetical Allele Code List (HTML) (old nomenclature)
  Note: New window will open.

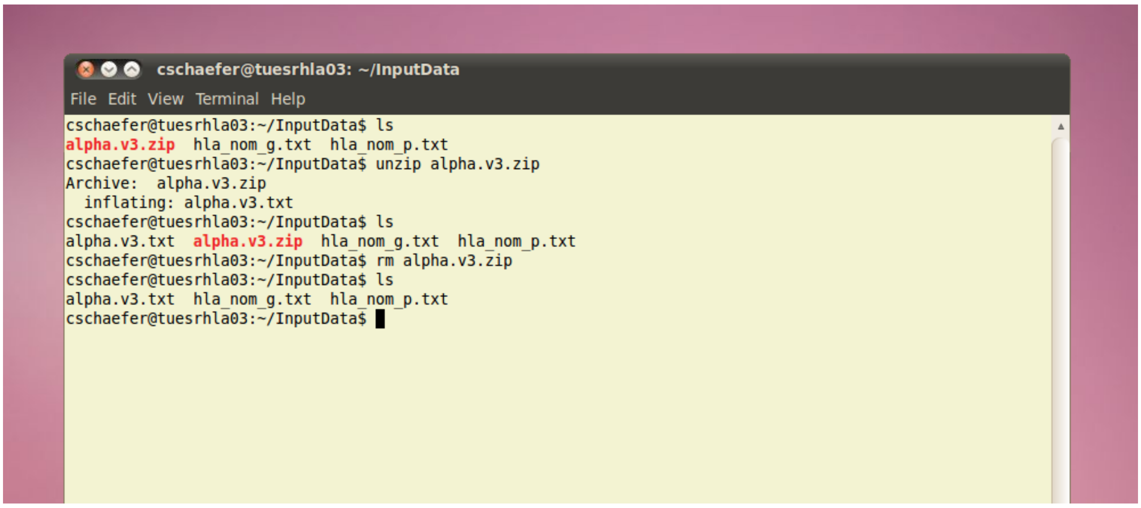
Text Format (.txt)

Download a zip compressed file. Once extracted, the text file will be called "alpha.txt."

- Alphabetical Allele Code List (ZIP) (new nomenclature)
  Note: Extraction requires a data compression program such as WinZip.
- Alphabetical Allele Code List (ZIP) (old nomenclature)
  Note: Extraction requires a data compression program such as WinZip.

The self-extracting executable file has been removed as of 10/21/03. The allele code lists will no longer be available for download in this format. Please mail new-allelecodes@nmdp.org with any questions or concerns regarding this change.

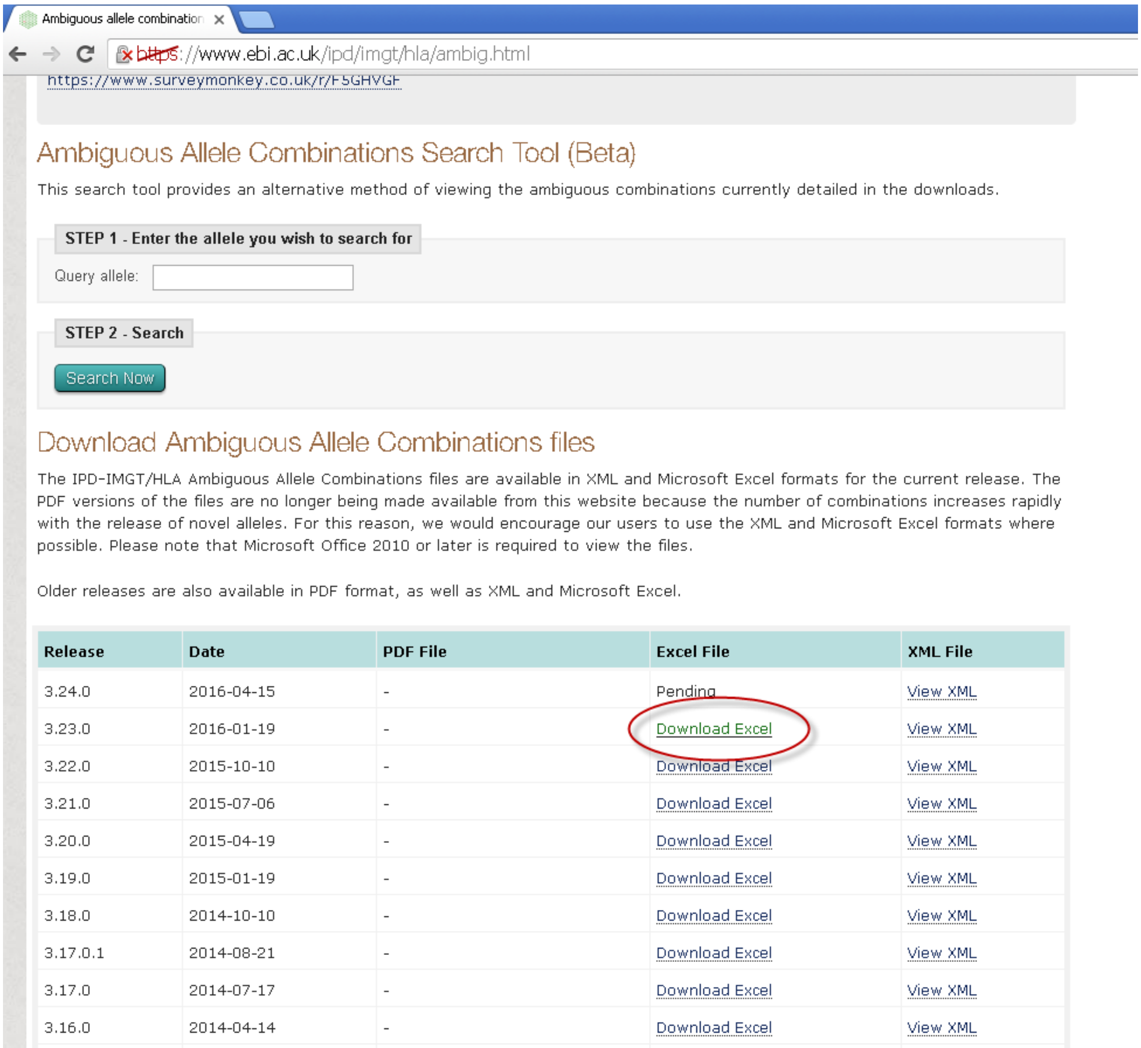
Enter the folder “Input data” and type “unzip alpha.v3.txt” to extract the archive. You can remove the archive “alpha.v3.zip” afterwards.



cschaefer@tuesrhla03: ~/InputData
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~/InputData$ ls
alpha.v3.zip  hla_nom_g.txt  hla_nom_p.txt
cschaefer@tuesrhla03:~/InputData$ unzip alpha.v3.zip
Archive:  alpha.v3.zip
  inflating: alpha.v3.txt
cschaefer@tuesrhla03:~/InputData$ ls
alpha.v3.txt  alpha.v3.zip  hla_nom_g.txt  hla_nom_p.txt
cschaefer@tuesrhla03:~/InputData$ rm alpha.v3.zip
cschaefer@tuesrhla03:~/InputData$ ls
alpha.v3.txt  hla_nom_g.txt  hla_nom_p.txt
cschaefer@tuesrhla03:~/InputData$ 
```

4) Go to the website https://www.ebi.ac.uk/ipd/imgt/hla/ambig.html. Click on "Download Excel" for the wanted release (usually the latest) and save ambiguity_v<>.xls (replace <> by version).



~~https~~://www.ebi.ac.uk/ipd/imgt/hla/ambig.html

https://www.surveymonkey.co.uk/r/F5GHVGF

Ambiguous Allele Combinations Search Tool (Beta)

This search tool provides an alternative method of viewing the ambiguous combinations currently detailed in the downloads.

STEP 1 - Enter the allele you wish to search for

Query allele:

STEP 2 - Search

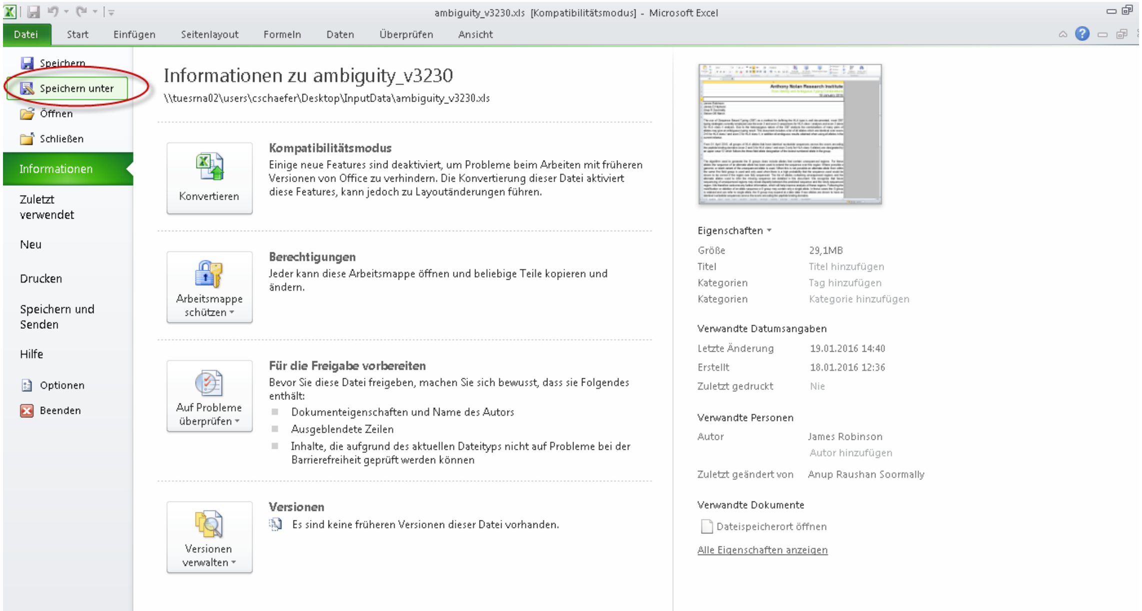
Download Ambiguous Allele Combinations files

The IPD-IMGT/HLA Ambiguous Allele Combinations files are available in XML and Microsoft Excel formats for the current release. The PDF versions of the files are no longer being made available from this website because the number of combinations increases rapidly with the release of novel alleles. For this reason, we would encourage our users to use the XML and Microsoft Excel formats where possible. Please note that Microsoft Office 2010 or later is required to view the files.

Older releases are also available in PDF format, as well as XML and Microsoft Excel.

Release	Date	PDF File	Excel File	XML File
3.24.0	2016-04-15	-	Pending	View XML
3.23.0	2016-01-19	-	Download Excel	View XML
3.22.0	2015-10-10	-	Download Excel	View XML
3.21.0	2015-07-06	-	Download Excel	View XML
3.20.0	2015-04-19	-	Download Excel	View XML
3.19.0	2015-01-19	-	Download Excel	View XML
3.18.0	2014-10-10	-	Download Excel	View XML
3.17.0.1	2014-08-21	-	Download Excel	View XML
3.17.0	2014-07-17	-	Download Excel	View XML
3.16.0	2014-04-14	-	Download Excel	View XML

Next we extract the information from the Excel sheet. Open ambiguity_v<>.xls in Excel and save as ambiguity_v<>.xlsm to run macros.



Informationen zu ambiguity_v3230

\\tuesrna02\users\cschaefer\Desktop\InputData\ambiguity_v3230.xls

Kompatibilitätsmodus

Einige neue Features sind deaktiviert, um Probleme beim Arbeiten mit früheren Versionen von Office zu verhindern. Die Konvertierung dieser Datei aktiviert diese Features, kann jedoch zu Layoutänderungen führen.

Berechtigungen

Jeder kann diese Arbeitsmappe öffnen und beliebige Teile kopieren und ändern.

Für die Freigabe vorbereiten

Bevor Sie diese Datei freigeben, machen Sie sich bewusst, dass sie Folgendes enthält:

- Dokumenteigenschaften und Name des Autors
- Ausgeblendete Zeilen
- Inhalte, die aufgrund des aktuellen Dateityps nicht auf Probleme bei der Barrierefreiheit geprüft werden können

Versionen

Es sind keine früheren Versionen dieser Datei vorhanden.

Eigenschaften

Größe	29,1MB
Titel	Titel hinzufügen
Kategorien	Tag hinzufügen
Kategorien	Kategorie hinzufügen

Verwandte Datumsangaben

Letzte Änderung	19.01.2016 14:40
Erstellt	18.01.2016 12:36
Zuletzt gedruckt	Nie

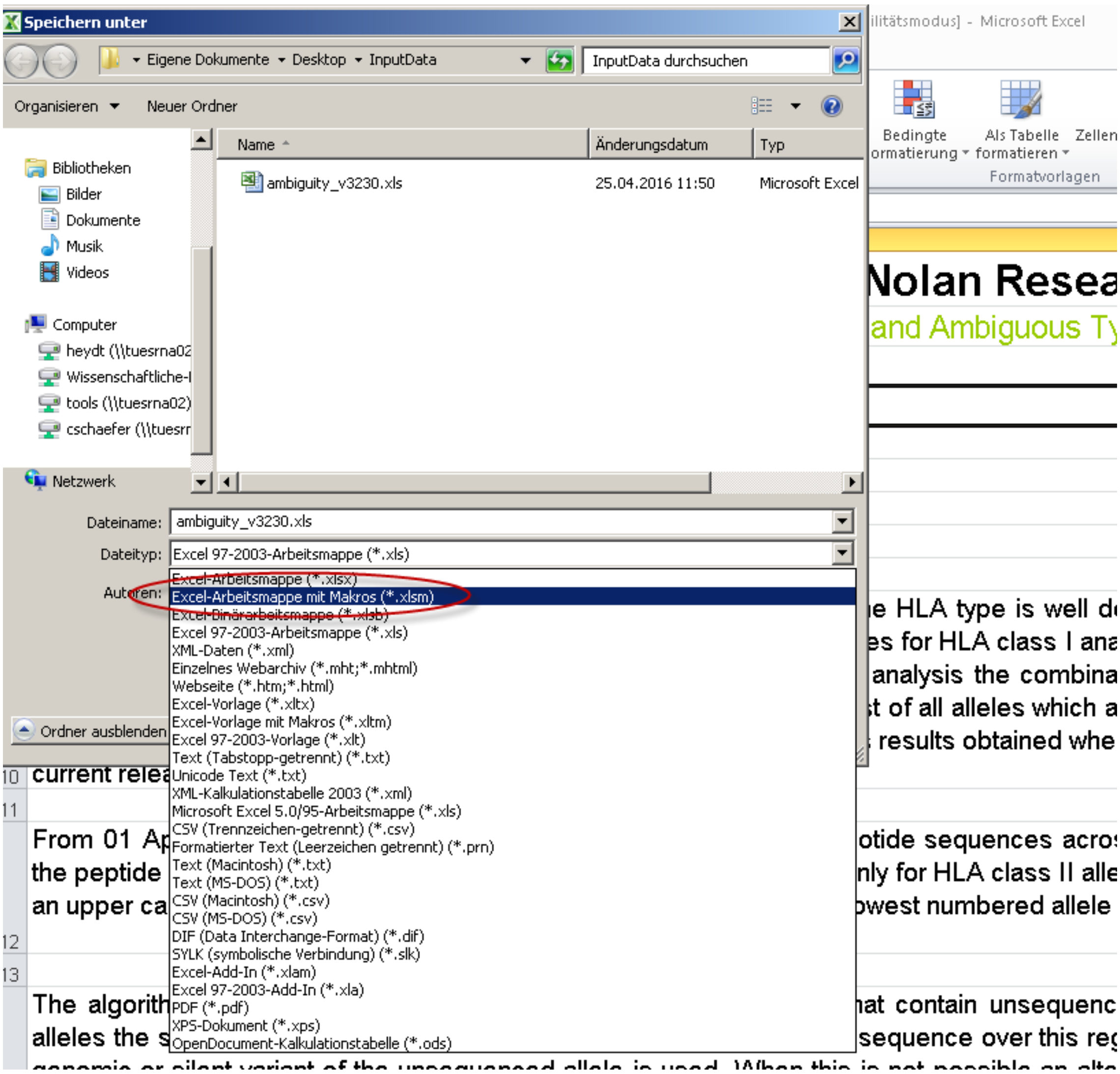
Verwandte Personen

Autor	James Robinson Autor hinzufügen
Zuletzt geändert von	Anup Raushan Soormally

Verwandte Dokumente

Dateispeicherort öffnen

Alle Eigenschaften anzeigen

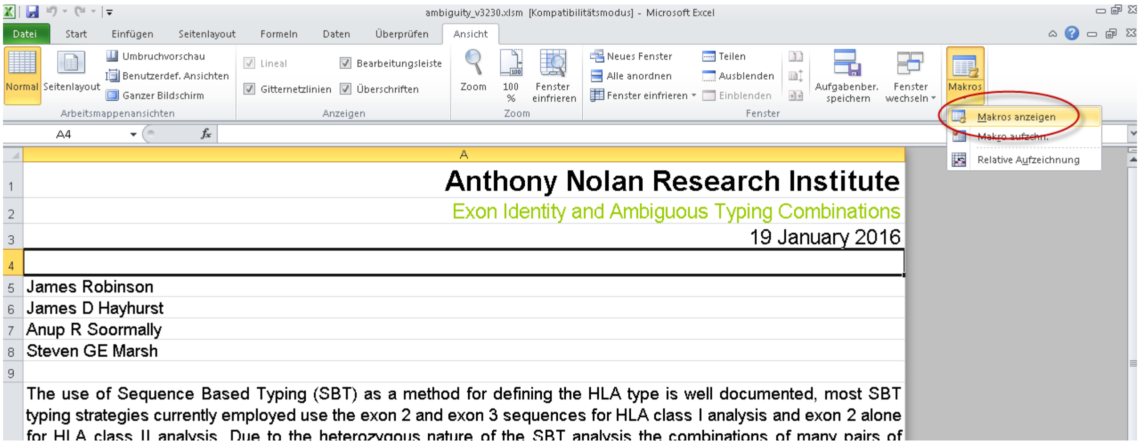

Speichern unter
Eigene Dokumente ▸ Desktop ▸ InputData
InputData durchsuchen
Organisieren
Neuer Ordner
Bibliotheken
Bilder
Dokumente
Musik
Videos
Computer
heydt (\\tuesrna02
Wissenschaftliche-l
tools (\\tuesrna02)
cschaefer (\\tuesrr
Netzwerk
Name
Änderungsdatum
Typ
ambiguity_v3230.xls
25.04.2016 11:50
Microsoft Excel
Autoren:
Ordner ausblenden
Excel-Arbeitsmappe (*.xlsx)
Excel-Arbeitsmappe mit Makros (*.xlsm)
Excel-Binärarbeitsmappe (*.xlsb)
Excel 97-2003-Arbeitsmappe (*.xls)
XML-Daten (*.xml)
Einzelnes Webarchiv (*.mht;*.mhtml)
Webseite (*.htm;*.html)
Excel-Vorlage (*.xltx)
Excel-Vorlage mit Makros (*.xltm)
Excel 97-2003-Vorlage (*.xlt)
Text (Tabstopp-getrennt) (*.txt)
Unicode Text (*.txt)
XML-Kalkulationstabelle 2003 (*.xml)
Microsoft Excel 5.0/95-Arbeitsmappe (*.xls)
CSV (Trennzeichen-getrennt) (*.csv)
Formatierter Text (Leerzeichen getrennt) (*.prn)
Text (Macintosh) (*.txt)
Text (MS-DOS) (*.txt)
CSV (Macintosh) (*.csv)
CSV (MS-DOS) (*.csv)
DIF (Data Interchange-Format) (*.dif)
SYLK (symbolische Verbindung) (*.slk)
Excel-Add-In (*.xlam)
Excel 97-2003-Add-In (*.xla)
PDF (*.pdf)
XPS-Dokument (*.xps)
OpenDocument-Kalkulationstabelle (*.ods)
ilitätsmodus] - Microsoft Excel
Bedingte ormatierung
Als Tabelle formatieren
Zellen
Formatvorlagen
Nolan Resea
and Ambiguous Ty

Dateiname: ambiguity_v3230.xls

Dateityp: Excel 97-2003-Arbeitsmappe (*.xls)

Now insert the following macro, which saves relevant information from the Excelsheets as text files:

```
Sub saveSheets()
Dim strPath As String, NameSheet As String, NameFile As String
Dim ws As Excel.Worksheet
strPath = Application.ActiveWorkbook.Path

For i = 2 To Worksheets.Count
Worksheets(i).Activate
Set ws = ActiveSheet
ws.Copy
NameSheet = ActiveSheet.name
NameFile = strPath & "\" & NameSheet & ".txt"
ActiveWorkbook.SaveAs Filename:=NameFile, FileFormat:=xlText
ActiveWorkbook.Close False
Next i
End Sub
```



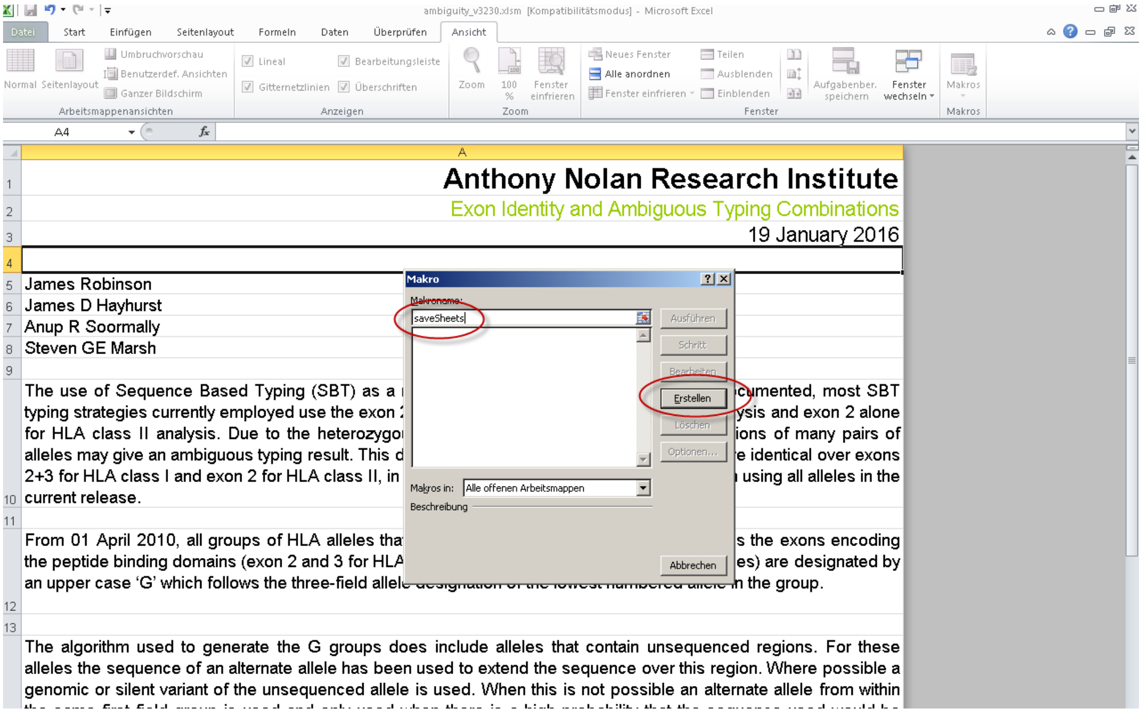
Anthony Nolan Research Institute

Exon Identity and Ambiguous Typing Combinations

19 January 2016

James Robinson
James D Hayhurst
Anup R Soormally
Steven GE Marsh

The use of Sequence Based Typing (SBT) as a method for defining the HLA type is well documented, most SBT typing strategies currently employed use the exon 2 and exon 3 sequences for HLA class I analysis and exon 2 alone for HLA class II analysis. Due to the heterozygous nature of the SBT analysis the combinations of many pairs of



Anthony Nolan Research Institute

Exon Identity and Ambiguous Typing Combinations

19 January 2016

James Robinson
James D Hayhurst
Anup R Soormally
Steven GE Marsh

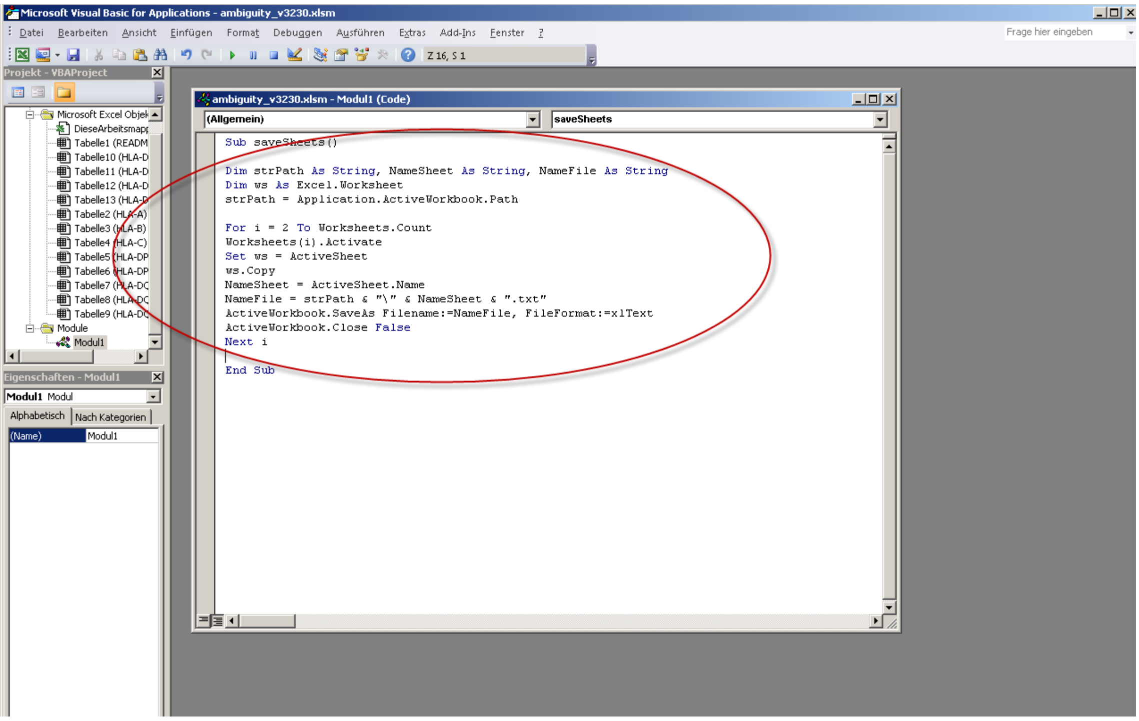
Makroname: saveSheets

Makros in: Alle offenen Arbeitsmappen

The use of Sequence Based Typing (SBT) as a … cumented, most SBT typing strategies currently employed use the exon … ysis and exon 2 alone for HLA class II analysis. Due to the heterozygo… ions of many pairs of alleles may give an ambiguous typing result. This d… e identical over exons 2+3 for HLA class I and exon 2 for HLA class II, in … using all alleles in the current release.

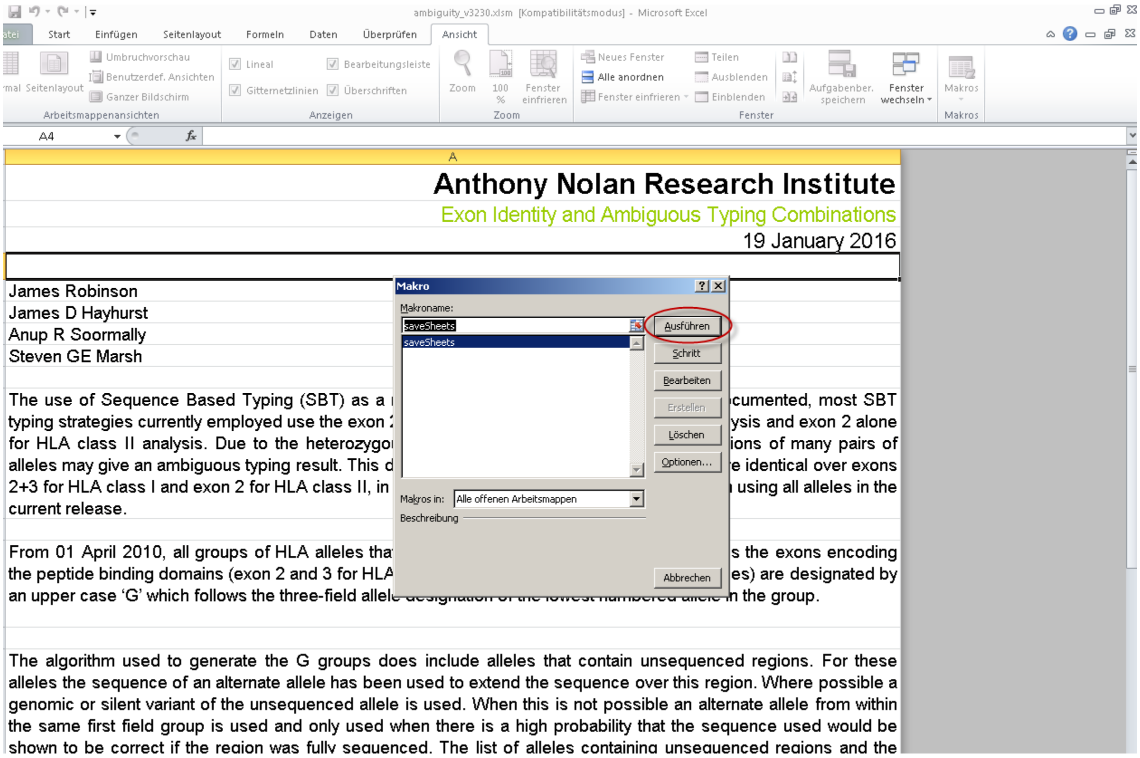
From 01 April 2010, all groups of HLA alleles tha… s the exons encoding the peptide binding domains (exon 2 and 3 for HLA … es) are designated by an upper case 'G' which follows the three-field allele designation of the lowest numbered allele in the group.

The algorithm used to generate the G groups does include alleles that contain unsequenced regions. For these alleles the sequence of an alternate allele has been used to extend the sequence over this region. Where possible a genomic or silent variant of the unsequenced allele is used. When this is not possible an alternate allele from within



```
Sub saveSheets()

Dim strPath As String, NameSheet As String, NameFile As String
Dim ws As Excel.Worksheet
strPath = Application.ActiveWorkbook.Path

For i = 2 To Worksheets.Count
Worksheets(i).Activate
Set ws = ActiveSheet
ws.Copy
NameSheet = ActiveSheet.Name
NameFile = strPath & "\" & NameSheet & ".txt"
ActiveWorkbook.SaveAs Filename:=NameFile, FileFormat:=xlText
ActiveWorkbook.Close False
Next i

End Sub
```

Close the window and execute the macro:



Anthony Nolan Research Institute

Exon Identity and Ambiguous Typing Combinations

19 January 2016

James Robinson
James D Hayhurst
Anup R Soormally
Steven GE Marsh

The use of Sequence Based Typing (SBT) as a | cumented, most SBT
typing strategies currently employed use the exon | ysis and exon 2 alone
for HLA class II analysis. Due to the heterozygo | ions of many pairs of
alleles may give an ambiguous typing result. This d | e identical over exons
2+3 for HLA class I and exon 2 for HLA class II, in | using all alleles in the
current release.

From 01 April 2010, all groups of HLA alleles tha | s the exons encoding
the peptide binding domains (exon 2 and 3 for HLA | es) are designated by
an upper case ‘G’ which follows the three-field allele designation of the lowest numbered allele in the group.

The algorithm used to generate the G groups does include alleles that contain unsequenced regions. For these alleles the sequence of an alternate allele has been used to extend the sequence over this region. Where possible a genomic or silent variant of the unsequenced allele is used. When this is not possible an alternate allele from within the same first field group is used and only used when there is a high probability that the sequence used would be shown to be correct if the region was fully sequenced. The list of alleles containing unsequenced regions and the

Makroname: saveSheets

saveSheets

Makros in: Alle offenen Arbeitsmappen

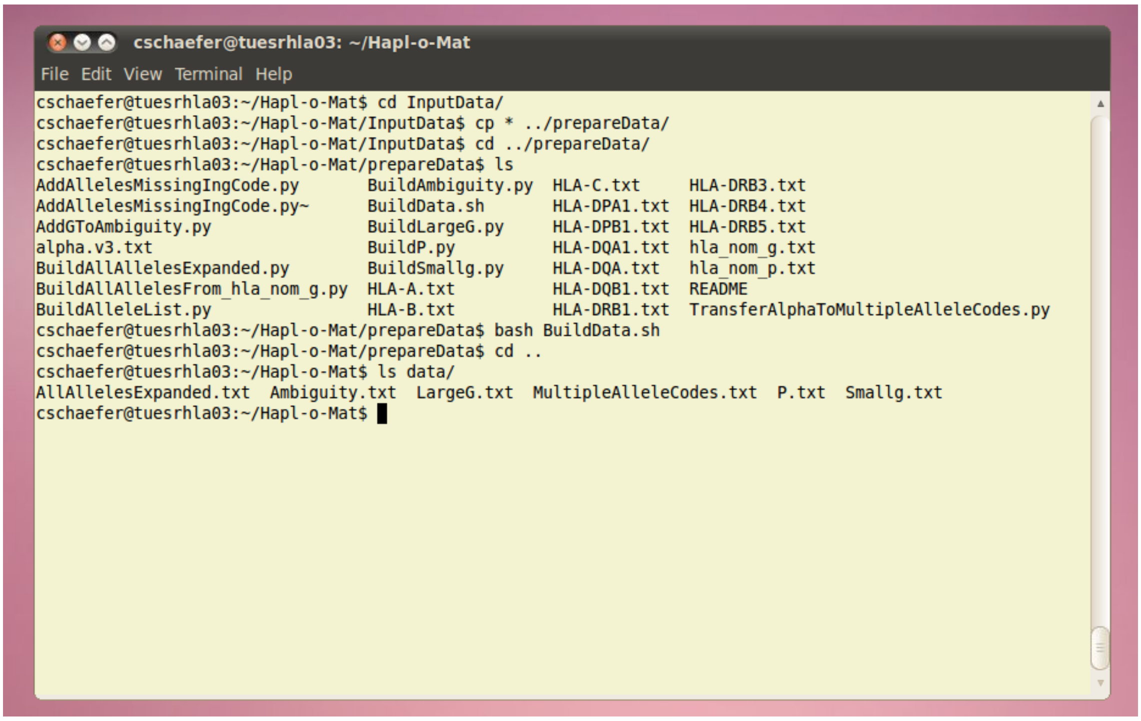
Beschreibung

Some new text files should have appeared in your folder „inputData“. Afterwards you can remove the Excel file.

Build Data for Hapl-o-Mat

Enter the folder InputData via “cd InputData” and copy all files via “cp * ../prepareData” to the folder “prepareData”. Then enter folder “prepareData” via “cd ../prepareData” and check what is there via typing “ls”. Next, create the data required for Hapl-o-Mat by running the bash script via “bash

BuildData.sh”. It automatically calls the python scripts and moves the created files to the folder “data”. Check for the created files by going one folder back typing “cd ..” and typing “ls data”.


cschaefer@tuesrhla03: ~/Hapl-o-Mat
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~/Hapl-o-Mat$ cd InputData/
cschaefer@tuesrhla03:~/Hapl-o-Mat/InputData$ cp * ../prepareData/
cschaefer@tuesrhla03:~/Hapl-o-Mat/InputData$ cd ../prepareData/
cschaefer@tuesrhla03:~/Hapl-o-Mat/prepareData$ ls
AddAllelesMissingIngCode.py        BuildAmbiguity.py  HLA-C.txt     HLA-DRB3.txt
AddAllelesMissingIngCode.py~       BuildData.sh       HLA-DPA1.txt  HLA-DRB4.txt
AddGToAmbiguity.py                 BuildLargeG.py     HLA-DPB1.txt  HLA-DRB5.txt
alpha.v3.txt                       BuildP.py          HLA-DQA1.txt  hla_nom_g.txt
BuildAllAllelesExpanded.py         BuildSmallg.py     HLA-DQA.txt   hla_nom_p.txt
BuildAllAllelesFrom_hla_nom_g.py   HLA-A.txt          HLA-DQB1.txt  README
BuildAlleleList.py                 HLA-B.txt          HLA-DRB1.txt  TransferAlphaToMultipleAlleleCodes.py
cschaefer@tuesrhla03:~/Hapl-o-Mat/prepareData$ bash BuildData.sh
cschaefer@tuesrhla03:~/Hapl-o-Mat/prepareData$ cd ..
cschaefer@tuesrhla03:~/Hapl-o-Mat$ ls data/
AllAllelesExpanded.txt  Ambiguity.txt  LargeG.txt  MultipleAlleleCodes.txt  P.txt  Smallg.txt
cschaefer@tuesrhla03:~/Hapl-o-Mat$ 
```

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 your data should be in one of the following data formats:

Data format	Description
MA	Ambiguities are encoded by multiple allele (MA) codes. 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 file 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 "examplePopulations/populationData_a.dat".
GLC	Genotypes with or without ambiguities are saved by genotype list strings. 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 "examplePopulations/populationData_b.dat”
GL	Genotypes with or without ambiguities are saved by genotype list (GL) strings. Population data is saved in two files. The pull-file contains an individual's identification number and a list of integer numbers, GL-ids, referring to its single-locus genotype. The GL-ids are separated from the identification number via ";" and from each other via ":". The second file, the glid-file, contains a translation from GL-ids starting with "1" to actual single-locus genotypes. GL-id and genotype are separated via ";". A GL-id of "0" is interpreted as a missing typing at the corresponding locus and does not require a translation in the glid-file. For an example refer to

	"examplePopulations/populationData_c.pull" and "examplePopulations/populationData_c.glid".
READ	Ambiguities are completely resolved and alleles are already translated to the wanted typing resolutions. The input data is of the format as Hapl-o-Mat records processed genotype data. This allows for easily repeating a run without the need to resolve genotype data again.

Parameters

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

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.
INITIALIZATION_HAPLOTYPEFREQUENCIES	Initialization routine for haplotype frequencies. It takes the following values: • "equal": All haplotype frequencies are initialized with the same frequency • "numberOccurrence": Haplotype frequencies are initialized according to the initial number of occurrence of the haplotype • "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.
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.

Depending on the input format (indicated in brackets), additional parameters are:

Parameter	Input format	Description
FILENAME_INPUT	MA, GLC, READ	The file name of the input population data
FILENAME_PULL	GL	The file name of the pull-file
FILENAME_GLID	GL	The file name of the glid-file
LOCI_AND_ RESOLUTIONS	MA, GL, GLC	Loci included into analysis and desired typing resolution per locus. The list is separated by "," and contains the locus name followed by ":" and the desired typing resolution, e.g. A:g,B:4d,C:g. Supported typing resolutions and their abbreviations are g-groups (g), P-groups (P), G-groups (G), 2-digit fields (2d), 4-digit fields (4d), 6-digit fields (6d), and 8-digit fields (8d). Alleles are not translated via the option asItIs (applying the ambiguity filter includes an intrinsic translation to G-groups)
LOCIORDER	GL	Specify the order of loci the individual's GL-ids correspond to. Loci are separated via ",".
RESOLVE_MISSING_ GENOTYPES	GL	Takes values "true" and "false". If set to true, a missing typing is replaced by a combination of all alleles from AlleleList.txt at the locus. Else, individuals with a missing typing are discarded from analysis
MINIMAL_FREQUENCY_ GENOTYPES	MA, GL, GLC	Genotypes which split into more genotypes than the inverse of this number are discarded from analysis
DO_AMBIGUITYFILTER	MA, GL, GLC	Takes values "true" and "false". The option "true" activates the ambiguity filter
EXPAND_LINES_ AMBIGUITYFILTER	MA, GL, GLC	Takes values "true" and "false". If set to "true", matching lines with additional genotype pairs in the ambiguity filter are considered

Whenever specifying a file name including folders, you have to create the folders before running Hapl-o-Mat.

Tutorials

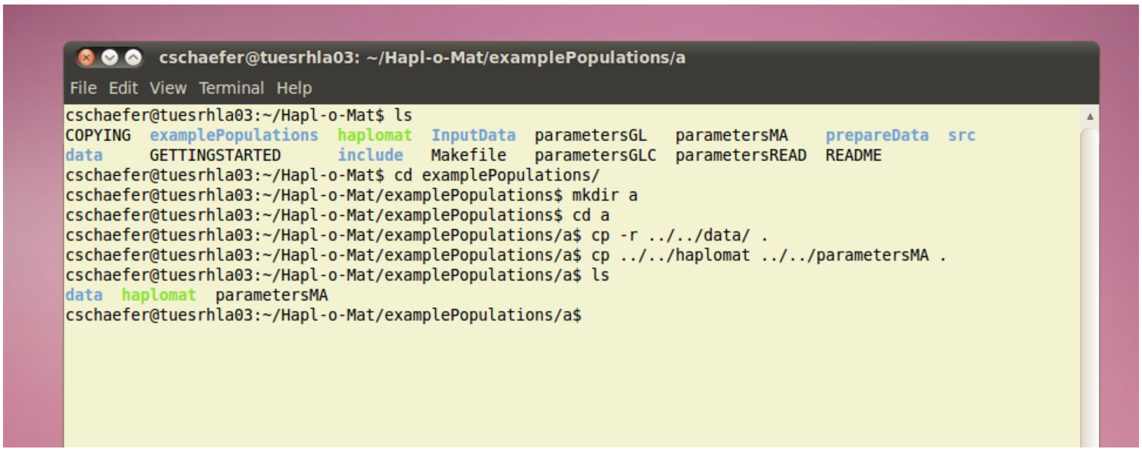
We have everything ready to use Hapl-o-Mat. In the following we estimate haplotype frequencies from some included genotype data recorded in different input formats.

Input Format MA

You find the relevant population data in "examplePopulations/populationData_a.dat". As ambiguities are recorded as multiple allele codes, the input format is MA. We are going to infer three locus (A, B, DRB1) haplotypes from this data. Alleles at loci A and B shall be translated to typing resolution g and alleles at locus DRB1 to 4-digits typing resolution.

Preparations

Enter the folder “examplePopulations” by typing “cd examplePopulations”, create a folder named "a" by typing “mkdir a”, and enter the folder by typing “cd a”. Then provide the data required by Hapl-o-Mat by copying the folder data to “a”, “cp -r ../../data .”. Additionally, copy the executable "haplomat" and the file "parametersMA" to folder "a", “cp ../../haplomat ../../parametersMA .”. Check that everything is there by typing “ls”.



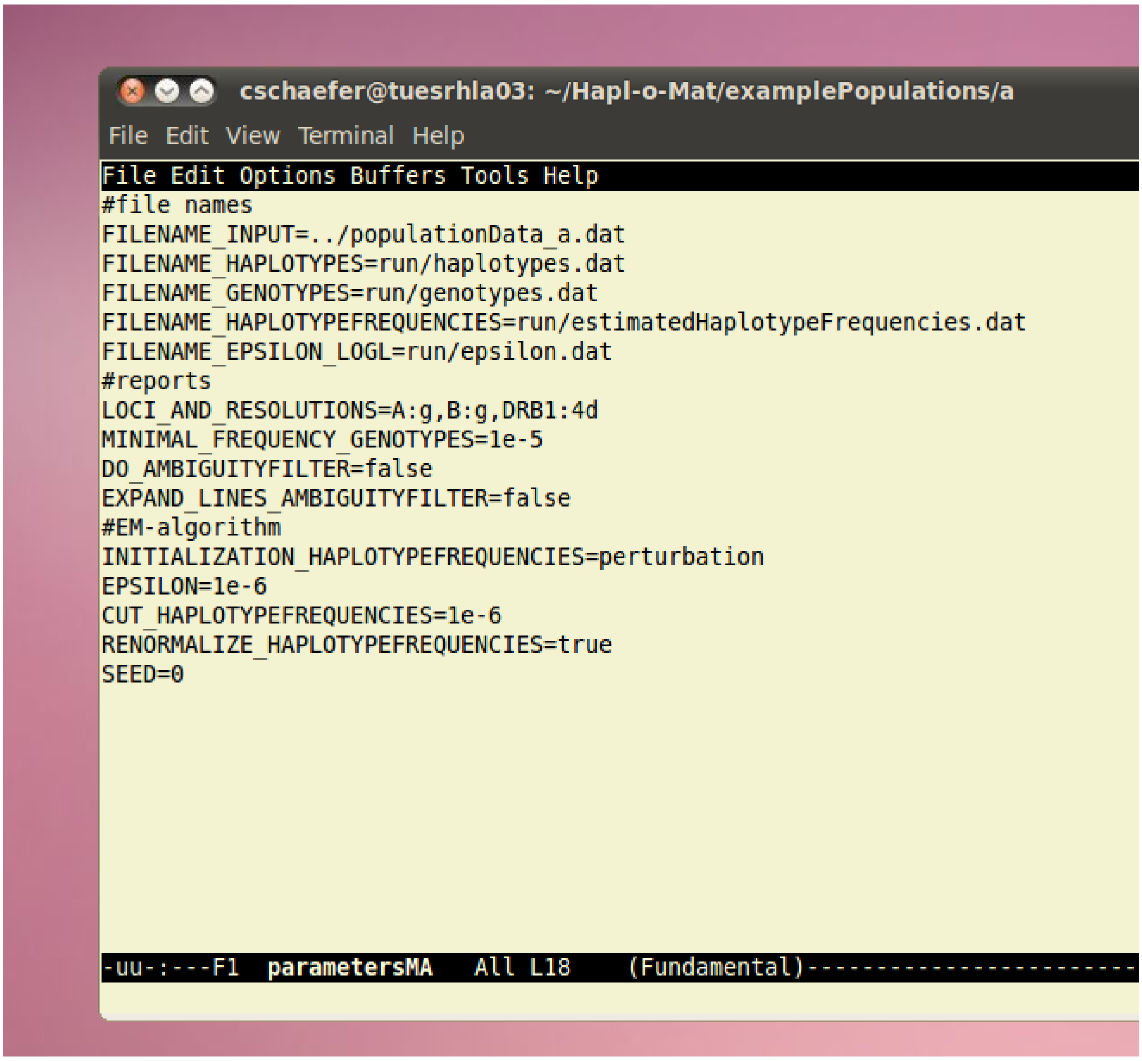
cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/a
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~/Hapl-o-Mat$ ls
COPYING  examplePopulations  haplomat  InputData  parametersGL   parametersMA    prepareData  src
data     GETTINGSTARTED      include   Makefile   parametersGLC  parametersREAD  README
cschaefer@tuesrhla03:~/Hapl-o-Mat$ cd examplePopulations/
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ mkdir a
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ cd a
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/a$ cp -r ../../data/ .
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/a$ cp ../../haplomat ../../parametersMA .
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/a$ ls
data  haplomat  parametersMA
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/a$
```

Parameters

According to the format of the input genotype data we use the parameter file “parametersMA”. Open it in a text editor of your choice and set the following values:

```
#file names
FILENAME_INPUT=../populationData_a.dat
FILENAME_HAPLOTYPES=run/haplotypes.dat
FILENAME_GENOTYPES=run/genotypes.dat
FILENAME_HAPLOTYPEFREQUENCIES=run/hfs.dat
FILENAME_EPSILON_LOGL=run/epsilon.dat
#reports
LOCI_AND_RESOLUTIONS=A:g,B:g,DRB1:4d
MINIMAL_FREQUENCY_GENOTYPES=1e-5
DO_AMBIGUITYFILTER=false
EXPAND_LINES_AMBIGUITYFILTER=false
#EM-algorithm
INITIALIZATION_HAPLOTYPEFREQUENCIES=perturbation
EPSILON=1e-6
CUT_HAPLOTYPEFREQUENCIES=1e-6
RENORMALIZE_HAPLOTYPEFREQUENCIES=true
SEED=1000
```


cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/a
File Edit View Terminal Help
File Edit Options Buffers Tools Help

```
#file names
FILENAME_INPUT=../populationData_a.dat
FILENAME_HAPLOTYPES=run/haplotypes.dat
FILENAME_GENOTYPES=run/genotypes.dat
FILENAME_HAPLOTYPEFREQUENCIES=run/estimatedHaplotypeFrequencies.dat
FILENAME_EPSILON_LOGL=run/epsilon.dat
#reports
LOCI_AND_RESOLUTIONS=A:g,B:g,DRB1:4d
MINIMAL_FREQUENCY_GENOTYPES=1e-5
DO_AMBIGUITYFILTER=false
EXPAND_LINES_AMBIGUITYFILTER=false
#EM-algorithm
INITIALIZATION_HAPLOTYPEFREQUENCIES=perturbation
EPSILON=1e-6
CUT_HAPLOTYPEFREQUENCIES=1e-6
RENORMALIZE_HAPLOTYPEFREQUENCIES=true
SEED=0
```

-uu-:---F1 parametersMA   All L18   (Fundamental)-----------------------

Do not forget to create the folder “run” by typing “mkdir run”.

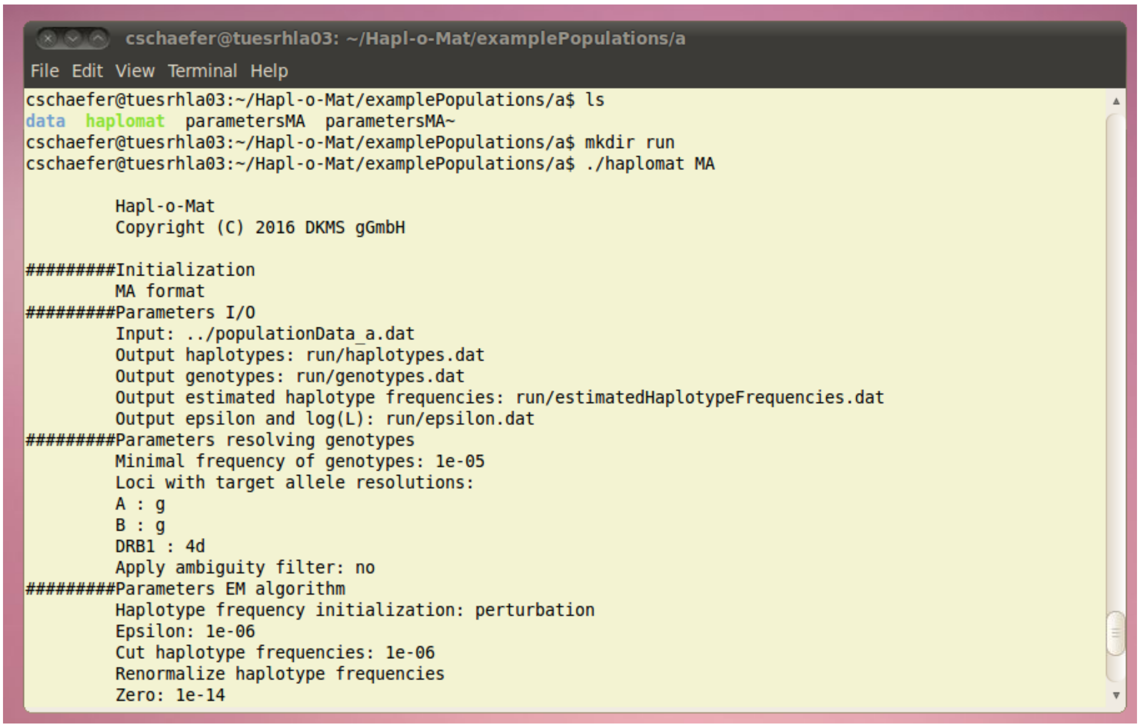
Run Hapl-o-Mat

Compute haplotype frequencies from the genotype input data by running Hapl-o-Mat. If you are not already there, go to folder “a” and run Hapl-o-Mat via

```
./haplomat MA
```

It produces some output on the screen including your chosen parameters, statistics on the resolved genotype data and the expectation-maximization algorithm, and the run time. You can easily write this output to an extra file by starting Hapl-o-Mat with

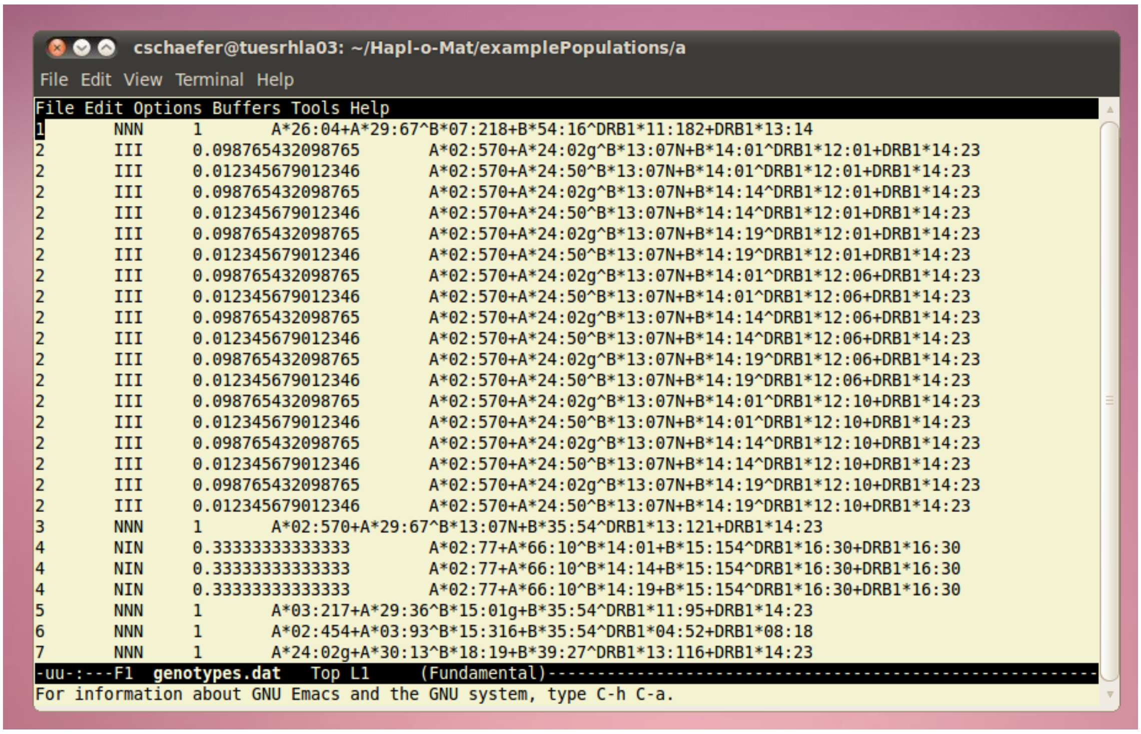
```
./haplomat MA > Log.dat
```


cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/a
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/a$ ls
data  haplomat  parametersMA  parametersMA~
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/a$ mkdir run
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/a$ ./haplomat MA

         Hapl-o-Mat
         Copyright (C) 2016 DKMS gGmbH

#########Initialization
         MA format
#########Parameters I/O
         Input: ../populationData_a.dat
         Output haplotypes: run/haplotypes.dat
         Output genotypes: run/genotypes.dat
         Output estimated haplotype frequencies: run/estimatedHaplotypeFrequencies.dat
         Output epsilon and log(L): run/epsilon.dat
#########Parameters resolving genotypes
         Minimal frequency of genotypes: 1e-05
         Loci with target allele resolutions:
         A : g
         B : g
         DRB1 : 4d
         Apply ambiguity filter: no
#########Parameters EM algorithm
         Haplotype frequency initialization: perturbation
         Epsilon: 1e-06
         Cut haplotype frequencies: 1e-06
         Renormalize haplotype frequencies
         Zero: 1e-14
```

Results

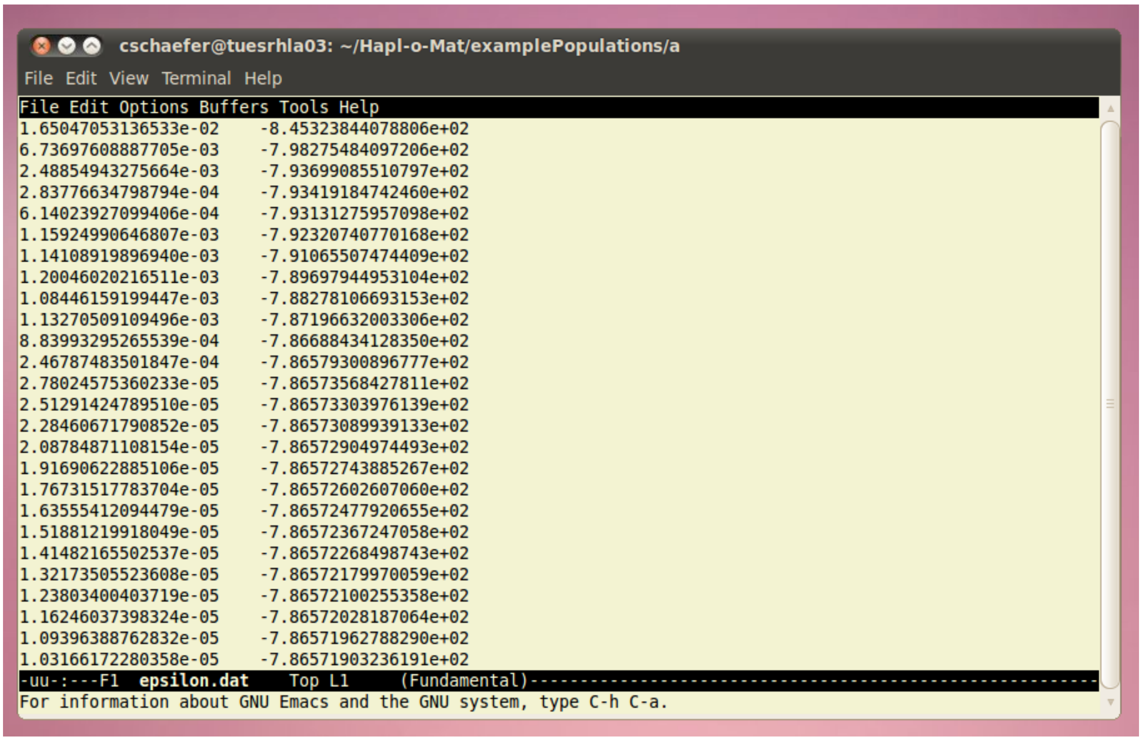
Now let's examine the results produced by Hapl-o-Mat. We first look into the file with the resolved genotypes, "run/genotypes.dat".


cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/a
File Edit View Terminal Help
File Edit Options Buffers Tools Help

```
1       NNN     1       A*26:04+A*29:67^B*07:218+B*54:16^DRB1*11:182+DRB1*13:14
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:01^DRB1*12:01+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:01^DRB1*12:01+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:14^DRB1*12:01+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:14^DRB1*12:01+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:19^DRB1*12:01+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:19^DRB1*12:01+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:01^DRB1*12:06+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:01^DRB1*12:06+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:14^DRB1*12:06+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:14^DRB1*12:06+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:19^DRB1*12:06+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:19^DRB1*12:06+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:01^DRB1*12:10+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:01^DRB1*12:10+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:14^DRB1*12:10+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:14^DRB1*12:10+DRB1*14:23
2       III     0.098765432098765       A*02:570+A*24:02g^B*13:07N+B*14:19^DRB1*12:10+DRB1*14:23
2       III     0.012345679012346       A*02:570+A*24:50^B*13:07N+B*14:19^DRB1*12:10+DRB1*14:23
3       NNN     1       A*02:570+A*29:67^B*13:07N+B*35:54^DRB1*13:121+DRB1*14:23
4       NIN     0.33333333333333        A*02:77+A*66:10^B*14:01+B*15:154^DRB1*16:30+DRB1*16:30
4       NIN     0.33333333333333        A*02:77+A*66:10^B*14:14+B*15:154^DRB1*16:30+DRB1*16:30
4       NIN     0.33333333333333        A*02:77+A*66:10^B*14:19+B*15:154^DRB1*16:30+DRB1*16:30
5       NNN     1       A*03:217+A*29:36^B*15:01g+B*35:54^DRB1*11:95+DRB1*14:23
6       NNN     1       A*02:454+A*03:93^B*15:316+B*35:54^DRB1*04:52+DRB1*08:18
7       NNN     1       A*24:02g+A*30:13^B*18:19+B*39:27^DRB1*13:116+DRB1*14:23
-uu-:---F1  genotypes.dat   Top L1     (Fundamental)-----------------------------------------------
For information about GNU Emacs and the GNU system, type C-h C-a.
```

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 GL 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.

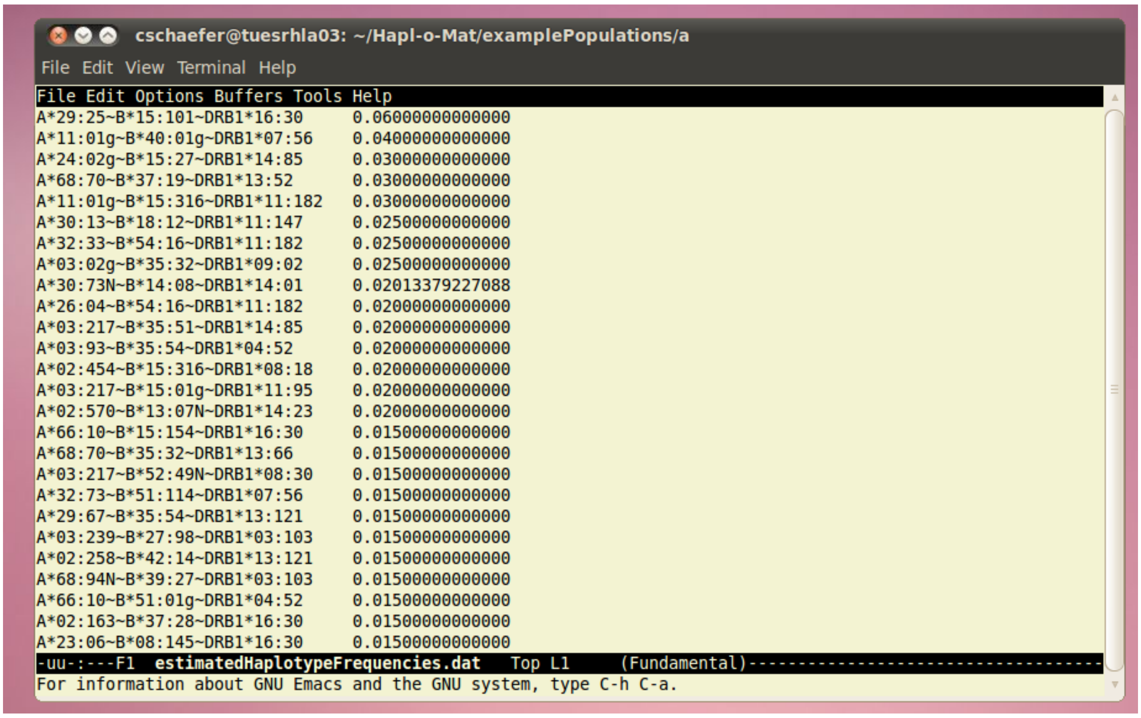
The evolution of the stopping criterion and log-likelihood while iterating expectation and maximization steps is written to "run/epsilon.dat". The first column is the stopping criterion and the second one the not normalized log-likelihood.



cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/a
File Edit View Terminal Help

```
File Edit Options Buffers Tools Help
1.65047053136533e-02    -8.45323844078806e+02
6.73697608887705e-03    -7.98275484097206e+02
2.48854943275664e-03    -7.93699085510797e+02
2.83776634798794e-04    -7.93419184742460e+02
6.14023927099406e-04    -7.93131275957098e+02
1.15924990646807e-03    -7.92320740770168e+02
1.14108919896940e-03    -7.91065507474409e+02
1.20046020216511e-03    -7.89697944953104e+02
1.08446159199447e-03    -7.88278106693153e+02
1.13270509109496e-03    -7.87196632003306e+02
8.83993295265539e-04    -7.86688434128350e+02
2.46787483501847e-04    -7.86579300896777e+02
2.78024575360233e-05    -7.86573568427811e+02
2.51291424789510e-05    -7.86573303976139e+02
2.28460671790852e-05    -7.86573089939133e+02
2.08784871108154e-05    -7.86572904974493e+02
1.91690622885106e-05    -7.86572743885267e+02
1.76731517783704e-05    -7.86572602607060e+02
1.63555412094479e-05    -7.86572477920655e+02
1.51881219918049e-05    -7.86572367247058e+02
1.41482165502537e-05    -7.86572268498743e+02
1.32173505523608e-05    -7.86572179970059e+02
1.23803400403719e-05    -7.86572100255358e+02
1.16246037398324e-05    -7.86572028187064e+02
1.09396388762832e-05    -7.86571962788290e+02
1.03166172280358e-05    -7.86571903236191e+02
-uu-:---F1  epsilon.dat    Top L1     (Fundamental)---------------------------------------------------------
For information about GNU Emacs and the GNU system, type C-h C-a.
```

The inferred haplotypes including estimated frequencies are listed in "run/hfs.dat". Haplotypes are saved in the GL format. This is the file you were aiming at. It is sorted by descending frequency and already normalized if you activated the corresponding option (we did in this tutorial).

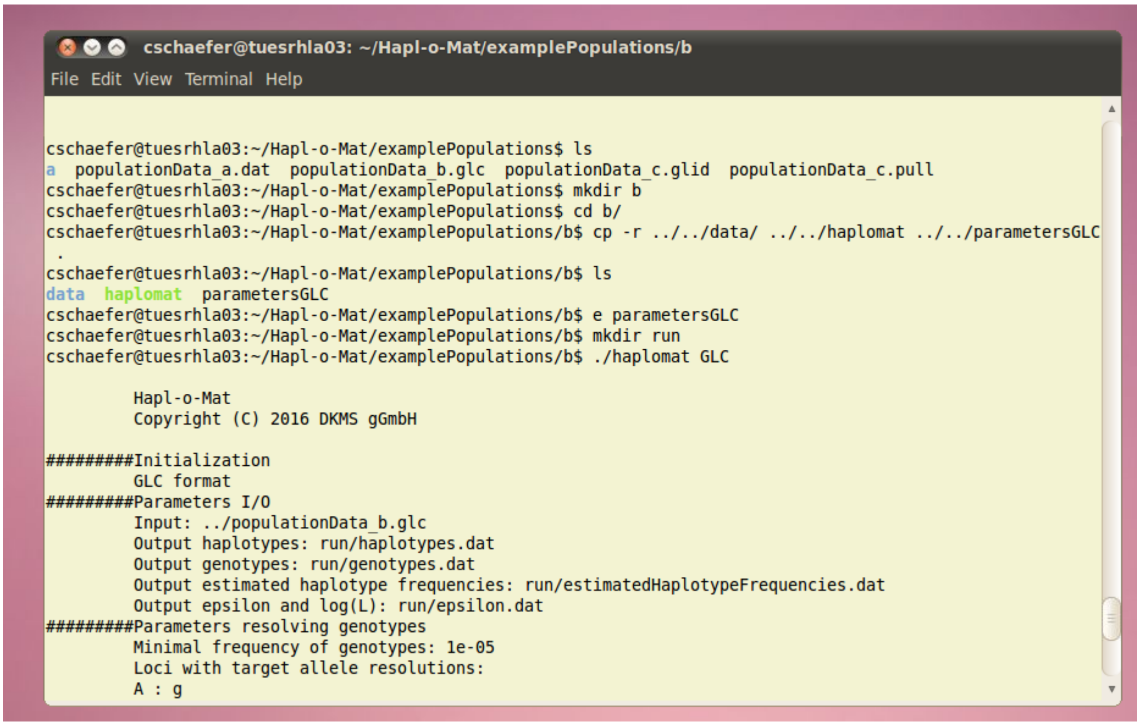

cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/a
File Edit View Terminal Help
File Edit Options Buffers Tools Help

```
A*29:25~B*15:101~DRB1*16:30     0.06000000000000
A*11:01g~B*40:01g~DRB1*07:56    0.04000000000000
A*24:02g~B*15:27~DRB1*14:85     0.03000000000000
A*68:70~B*37:19~DRB1*13:52      0.03000000000000
A*11:01g~B*15:316~DRB1*11:182   0.03000000000000
A*30:13~B*18:12~DRB1*11:147     0.02500000000000
A*32:33~B*54:16~DRB1*11:182     0.02500000000000
A*03:02g~B*35:32~DRB1*09:02     0.02500000000000
A*30:73N~B*14:08~DRB1*14:01     0.02013379227088
A*26:04~B*54:16~DRB1*11:182     0.02000000000000
A*03:217~B*35:51~DRB1*14:85     0.02000000000000
A*03:93~B*35:54~DRB1*04:52      0.02000000000000
A*02:454~B*15:316~DRB1*08:18    0.02000000000000
A*03:217~B*15:01g~DRB1*11:95    0.02000000000000
A*02:570~B*13:07N~DRB1*14:23    0.02000000000000
A*66:10~B*15:154~DRB1*16:30     0.01500000000000
A*68:70~B*35:32~DRB1*13:66      0.01500000000000
A*03:217~B*52:49N~DRB1*08:30    0.01500000000000
A*32:73~B*51:114~DRB1*07:56     0.01500000000000
A*29:67~B*35:54~DRB1*13:121     0.01500000000000
A*03:239~B*27:98~DRB1*03:103    0.01500000000000
A*02:258~B*42:14~DRB1*13:121    0.01500000000000
A*68:94N~B*39:27~DRB1*03:103    0.01500000000000
A*66:10~B*51:01g~DRB1*04:52     0.01500000000000
A*02:163~B*37:28~DRB1*16:30     0.01500000000000
A*23:06~B*08:145~DRB1*16:30     0.01500000000000
-uu-:---F1  estimatedHaplotypeFrequencies.dat   Top L1     (Fundamental)------------------------------------
For information about GNU Emacs and the GNU system, type C-h C-a.
```

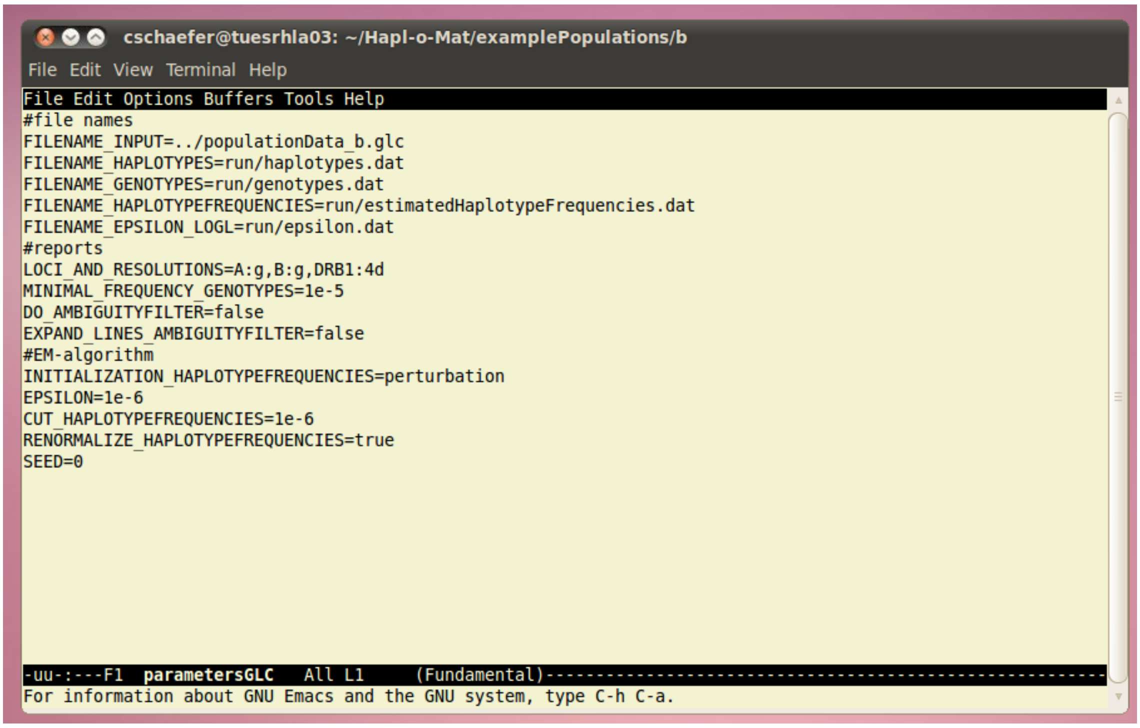
Input Format GLC

This time ambiguities in the genotypic population data are recorded via genotype list strings. The file with the population data is called "populationData_b.glc". As all the information is in one file, the input format is GLC. Running Hapl-o-Mat works exactly as in the first tutorial. You just use the parameter file "parametersGLC" instead of "parametersMA" and make the appropriate changes. Run Hapl-o-Mat in folder "b" with

```
./haplomat GLC
```


cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/b
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ ls
a  populationData_a.dat  populationData_b.glc  populationData_c.glid  populationData_c.pull
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ mkdir b
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ cd b/
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/b$ cp -r ../../data/ ../../haplomat ../../parametersGLC
 .
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/b$ ls
data  haplomat  parametersGLC
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/b$ e parametersGLC
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/b$ mkdir run
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/b$ ./haplomat GLC

         Hapl-o-Mat
         Copyright (C) 2016 DKMS gGmbH

#########Initialization
         GLC format
#########Parameters I/O
         Input: ../populationData_b.glc
         Output haplotypes: run/haplotypes.dat
         Output genotypes: run/genotypes.dat
         Output estimated haplotype frequencies: run/estimatedHaplotypeFrequencies.dat
         Output epsilon and log(L): run/epsilon.dat
#########Parameters resolving genotypes
         Minimal frequency of genotypes: 1e-05
         Loci with target allele resolutions:
         A : g
```



cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/b
File Edit View Terminal Help
File Edit Options Buffers Tools Help

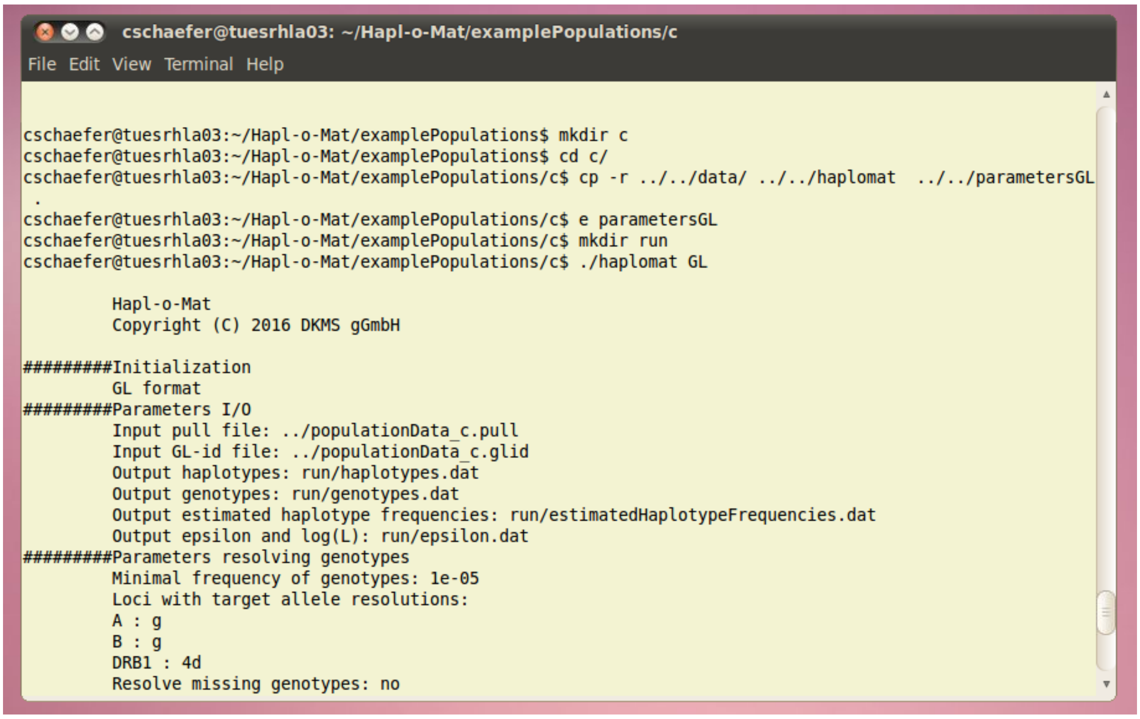
```
#file names
FILENAME_INPUT=../populationData_b.glc
FILENAME_HAPLOTYPES=run/haplotypes.dat
FILENAME_GENOTYPES=run/genotypes.dat
FILENAME_HAPLOTYPEFREQUENCIES=run/estimatedHaplotypeFrequencies.dat
FILENAME_EPSILON_LOGL=run/epsilon.dat
#reports
LOCI_AND_RESOLUTIONS=A:g,B:g,DRB1:4d
MINIMAL_FREQUENCY_GENOTYPES=1e-5
DO_AMBIGUITYFILTER=false
EXPAND_LINES_AMBIGUITYFILTER=false
#EM-algorithm
INITIALIZATION_HAPLOTYPEFREQUENCIES=perturbation
EPSILON=1e-6
CUT_HAPLOTYPEFREQUENCIES=1e-6
RENORMALIZE_HAPLOTYPEFREQUENCIES=true
SEED=0
```

-uu-:---F1 parametersGLC   All L1   (Fundamental)------------------------------------
For information about GNU Emacs and the GNU system, type C-h C-a.

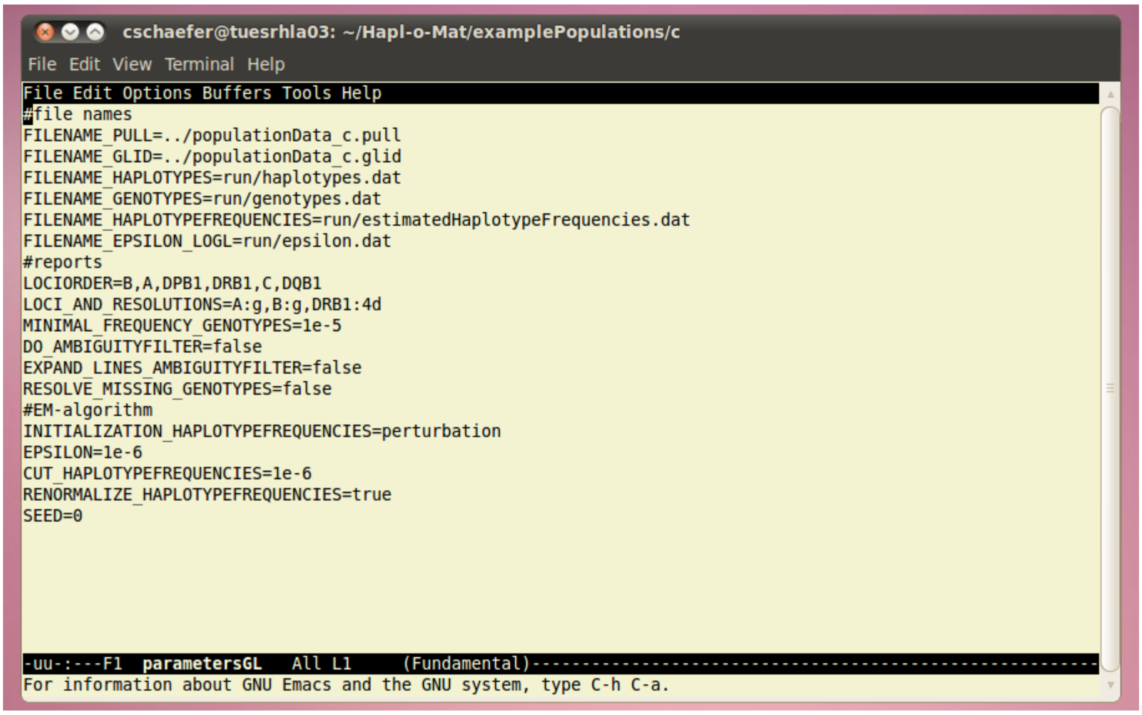
Input Format GL

Again, ambiguities in the genotypic population data are recorded via genotype list strings. Since the data is saved in two different files, the input format is GL. Follow the steps from tutorial a), but use the parameter file "parametersGL". The file names for the population data are populationData_c.pull" and "populationData_c.glid". I guess, you can figure out the matching positions in the parameter file. GL input format requires the order of loci as input, which can be obtained by looking in the pull- and glid-file. The first individual from "populationData_c.pull" has GL-ids 1, 2, 3, 4, 5, and 6. We know from "populationData_c.pull" that they correspond to loci B, A, DPB1, DRB1, C, and DQB1, respectively. Because of that we set LOCIORDER=B,A,DPB1,DRB1,C,DQB1". Finally, set the additional option RESOLVE_MISSING_GENOTYPE to "false". Run Hapl-o-Mat in folder "c" with

```
./haplomat GL
```


cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/c
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ mkdir c
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ cd c/
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/c$ cp -r ../../data/ ../../haplomat  ../../parametersGL
 .
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/c$ e parametersGL
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/c$ mkdir run
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/c$ ./haplomat GL

         Hapl-o-Mat
         Copyright (C) 2016 DKMS gGmbH

#########Initialization
         GL format
#########Parameters I/O
         Input pull file: ../populationData_c.pull
         Input GL-id file: ../populationData_c.glid
         Output haplotypes: run/haplotypes.dat
         Output genotypes: run/genotypes.dat
         Output estimated haplotype frequencies: run/estimatedHaplotypeFrequencies.dat
         Output epsilon and log(L): run/epsilon.dat
#########Parameters resolving genotypes
         Minimal frequency of genotypes: 1e-05
         Loci with target allele resolutions:
         A : g
         B : g
         DRB1 : 4d
         Resolve missing genotypes: no
```

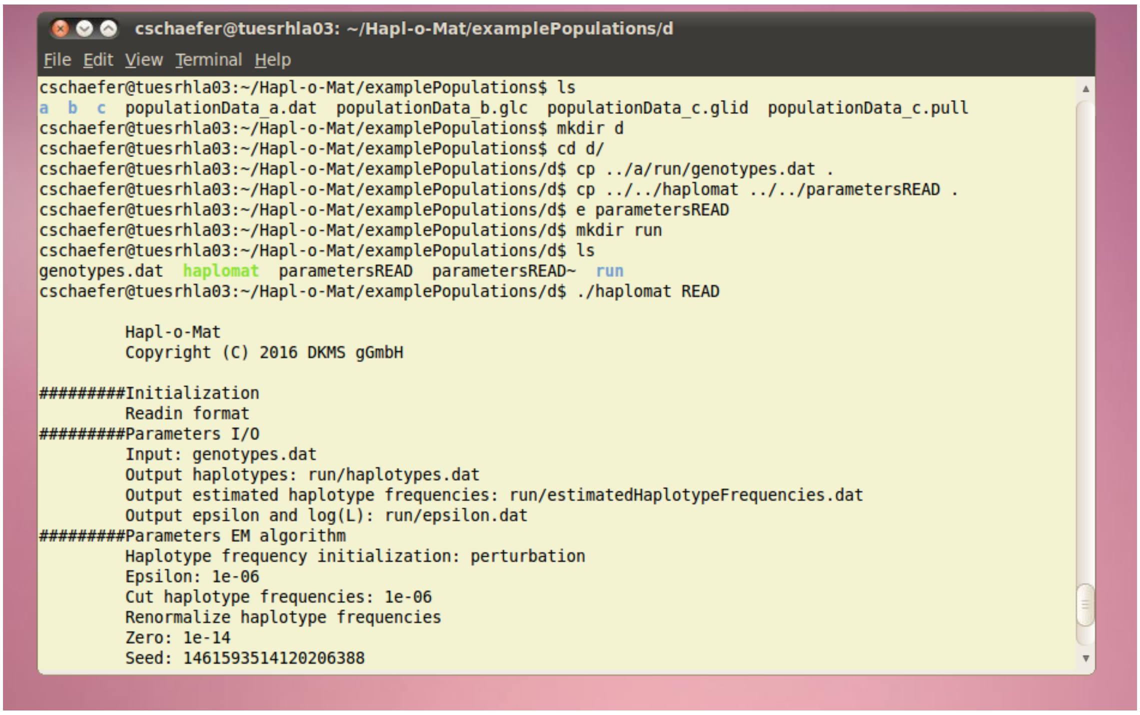

cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/c
File Edit View Terminal Help

```
File Edit Options Buffers Tools Help
#file names
FILENAME_PULL=../populationData_c.pull
FILENAME_GLID=../populationData_c.glid
FILENAME_HAPLOTYPES=run/haplotypes.dat
FILENAME_GENOTYPES=run/genotypes.dat
FILENAME_HAPLOTYPEFREQUENCIES=run/estimatedHaplotypeFrequencies.dat
FILENAME_EPSILON_LOGL=run/epsilon.dat
#reports
LOCIORDER=B,A,DPB1,DRB1,C,DQB1
LOCI_AND_RESOLUTIONS=A:g,B:g,DRB1:4d
MINIMAL_FREQUENCY_GENOTYPES=1e-5
DO_AMBIGUITYFILTER=false
EXPAND_LINES_AMBIGUITYFILTER=false
RESOLVE_MISSING_GENOTYPES=false
#EM-algorithm
INITIALIZATION_HAPLOTYPEFREQUENCIES=perturbation
EPSILON=1e-6
CUT_HAPLOTYPEFREQUENCIES=1e-6
RENORMALIZE_HAPLOTYPEFREQUENCIES=true
SEED=0






-uu-:---F1  parametersGL   All L1     (Fundamental)----------------------------------------------------
For information about GNU Emacs and the GNU system, type C-h C-a.
```

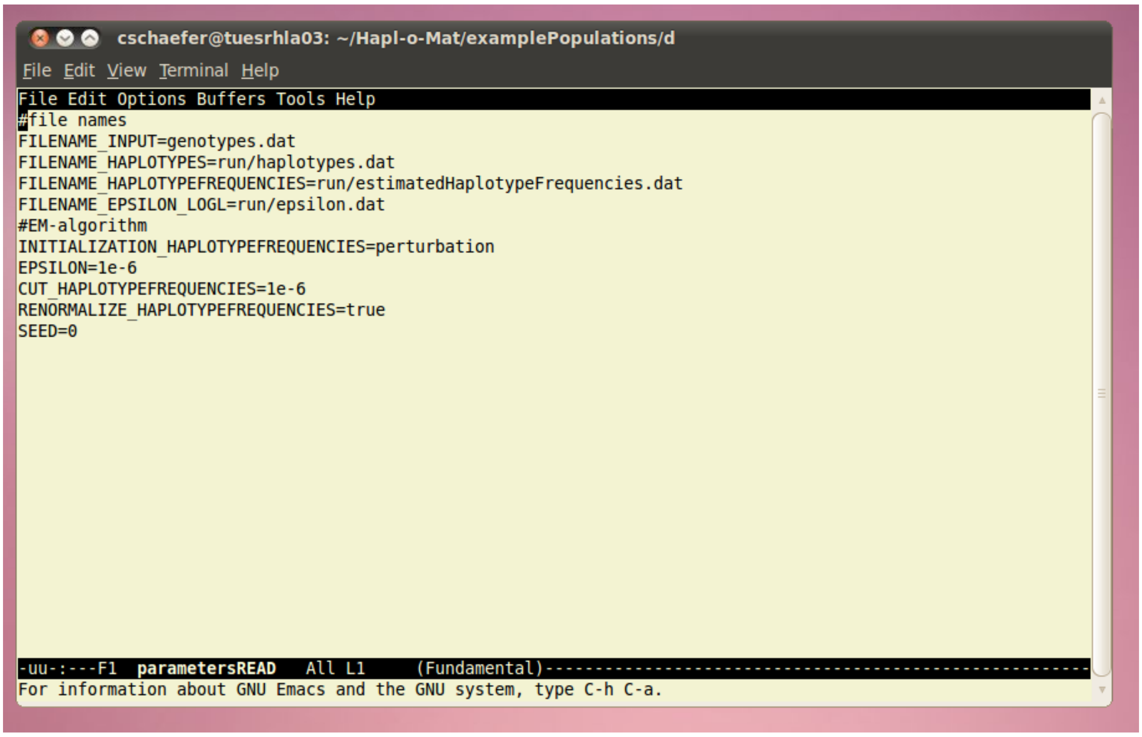
Input Format READ

Finally, we test the input format READ. Create a folder "d" and copy one file with resolved genotypes, say "a/run/genotypes.dat" there. Add "haplomat" and "parametersREAD" to this folder. Using the input format READ, Hapl-o-Mat does not resolve ambiguities or translates alleles, but reads in already resolved genotype data. Because of that the folder “data” is not required and the parameter file “parameterREADS” misses some options. Just adjust the file names and set parameters for the haplotype frequency estimation. Run Hapl-o-Mat in folder "d" via

./haplomat READ


cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/d
File Edit View Terminal Help

```
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ ls
a  b  c  populationData_a.dat  populationData_b.glc  populationData_c.glid  populationData_c.pull
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ mkdir d
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations$ cd d/
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/d$ cp ../a/run/genotypes.dat .
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/d$ cp ../../haplomat ../../parametersREAD .
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/d$ e parametersREAD
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/d$ mkdir run
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/d$ ls
genotypes.dat  haplomat  parametersREAD  parametersREAD~  run
cschaefer@tuesrhla03:~/Hapl-o-Mat/examplePopulations/d$ ./haplomat READ

         Hapl-o-Mat
         Copyright (C) 2016 DKMS gGmbH

#########Initialization
         Readin format
#########Parameters I/O
         Input: genotypes.dat
         Output haplotypes: run/haplotypes.dat
         Output estimated haplotype frequencies: run/estimatedHaplotypeFrequencies.dat
         Output epsilon and log(L): run/epsilon.dat
#########Parameters EM algorithm
         Haplotype frequency initialization: perturbation
         Epsilon: 1e-06
         Cut haplotype frequencies: 1e-06
         Renormalize haplotype frequencies
         Zero: 1e-14
         Seed: 1461593514120206388
```


cschaefer@tuesrhla03: ~/Hapl-o-Mat/examplePopulations/d
File Edit View Terminal Help
File Edit Options Buffers Tools Help

```
#file names
FILENAME_INPUT=genotypes.dat
FILENAME_HAPLOTYPES=run/haplotypes.dat
FILENAME_HAPLOTYPEFREQUENCIES=run/estimatedHaplotypeFrequencies.dat
FILENAME_EPSILON_LOGL=run/epsilon.dat
#EM-algorithm
INITIALIZATION_HAPLOTYPEFREQUENCIES=perturbation
EPSILON=1e-6
CUT_HAPLOTYPEFREQUENCIES=1e-6
RENORMALIZE_HAPLOTYPEFREQUENCIES=true
SEED=0
```

```
-uu-:---F1  parametersREAD    All L1     (Fundamental)----------------------------------------------
For information about GNU Emacs and the GNU system, type C-h C-a.
```

Quick Guide

The following overview gives you a small reminder on how to use Hapl-o-Mat:

1) Build the executable “haplomat” via make.

2) Update the data comprising information on the HLA nomenclature following the instructions under Data Preparation or in the file “prepareData/README”.

3) Prepare the genotype population data you want to study. Identify how genotyping ambiguities are recorded (NMDP codes or GL strings) and choose the input format

accordingly. Adapt the format of your data, e.g. include the header line or make alleles TAB separated.

4) Set the parameters in the parameter file corresponding to your input format.

5) Copy the executable “haplomat”, the folder “data”, the parameter file, and your input population data into one folder. Create any folders you specified in the parameter file. You do not need all the other files to run Hapl-o-Mat. Run Hapl-o-Mat.