applied biosystems

Axiom™ HLA Analysis v1.2

USER GUIDE

Publication Number 703338
Revision 3

For Research Use Only. Not for use in diagnostic procedures.

ThermoFisher SCIENTIFIC

Affymetrix, Inc.
3450 Central Expressway
Santa Clara, CA 95051

The information in this guide is subject to change without notice.

DISCLAIMER

TO THE EXTENT ALLOWED BY LAW, LIFE TECHNOLOGIES AND/OR ITS AFFILIATE(S) WILL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, INDIRECT, PUNITIVE, MULTIPLE, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING FROM THIS DOCUMENT, INCLUDING YOUR USE OF IT.

Revision history: Pub No. 703338

Revision	Date	Description
3	June 2019	Version 1.2 release. Added Windows 10 compatibility.
2	June 2015	Version 1.1 release
1	March 2015	Initial release.

Important Software Licensing Information

Legal entity

Affymetrix, Inc. | Santa Clara, CA 95051 USA | Toll Free in USA 1 800 955 6288

TRADEMARKS

Contents

Overview

Determining HLA genotypes is an important part of many studies to understand the genetic basis of immune response, disease associations, and transplant tolerance or rejection. The ability to determine HLA types from new or pre-existing genotyping data in parallel with other genetic analyses is a combination that will enable new scientific insights with greater efficiency.

Axiom HLA Analysis software uses a multi-population reference panel and the HLA type imputation model HLA*IMP:02[1] to statistically infer the HLA types of human samples from genotype data generated from Affymetrix genotyping arrays.

Through our on-going work with collaborators, Affymetrix continues to improve the multi-population reference panel to benefit the global community.

[1] = For more information on the HLA*IMP:02 algorithm, see Appendix: Multi-Population Classical HLA Imputation.

System requirements

Operating System	Speed	Memory (RAM)	Available Disk Space
Microsoft Windows® 10 (64 bit) Professional	2.83 GHz Intel Pentium Quad Core Processor	16 GB RAM	150 GB
Microsoft Windows® 7 (64 bit) Professional with Service Pack 1	2.83 GHz Intel Pentium Quad Core Processor	16 GB RAM	150 GB

Installation

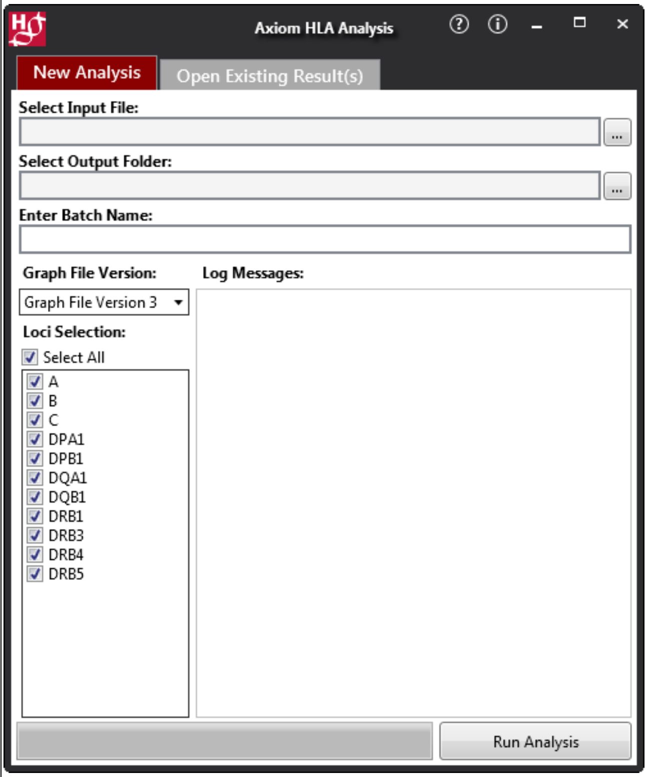
1. Download the Axiom HLA Analysis 1.2 zip package from the Affymetrix website.
2. Save the zipped package to a local (easily accessible) folder on your system.
3. Unzip the file as you normally would.
4. Locate the **Axiom HLA Analysis.exe** file, then double-click on it.
5. Follow the installer’s on-screen instructions.

Starting Axiom HLA Analysis

1. Click **Start** → **All Programs** → **Thermo Fisher Scientific** → **Axiom HLA Analysis** or double-click on the Desktop icon.

The main **Axiom HLA Analysis** window appears. (Figure 1)

Figure 1 Axiom HLA Analysis main window



Graph File Version: Graph File Version 3

Loci Selection:

☑ Select All

☑ A
☑ B
☑ C
☑ DPA1
☑ DPB1
☑ DQA1
☑ DQB1
☑ DRB1
☑ DRB3
☑ DRB4
☑ DRB5

Selecting an input file

IMPORTANT! Your input file must be a valid tab-delimited VCF (Variant Call Format) file. To create a VCF file, see "Ways to generate a VCF file" on page 16.

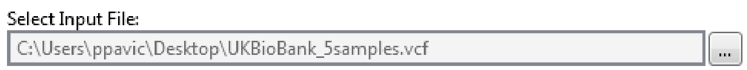
1. Click the Input File's **Browse** ... button.

   An Explorer window appears.

2. Navigate to your VCF file location, then click **Open**.

   Your selected file and its path now appears, as shown in Figure 2.

Figure 2 Input file path


Select Input File: C:\Users\ppavic\Desktop\UKBioBank_5samples.vcf

Note: It is highly recommended your VCF file contains 20 or more samples.

Assigning an output file location

1. Click the Output Folder path's **Browse** ... button.

   A **Please select folder** window appears.

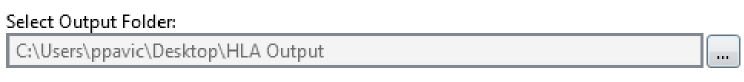
2. Navigate (as you normally would) to an easily accessible local output folder location, then click the window's New folder button.

   A new folder is created.

3. Enter a folder name (Example: **HLA_Output**), then click Select Folder.

   Your output folder and its path now appears, as shown in Figure 3.

Figure 3 Output folder path



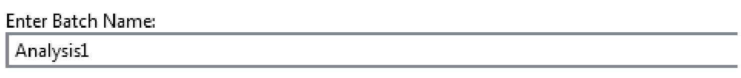
Select Output Folder: C:\Users\ppavic\Desktop\HLA Output

Assigning a batch name

1. Click inside the **Enter Batch Name** field (Figure 4), then enter a batch name.

   A warning message appears if you enter a batch name that contains a special character that is not supported. (Example: **"."**) Acknowledge the message, then enter a different batch name.

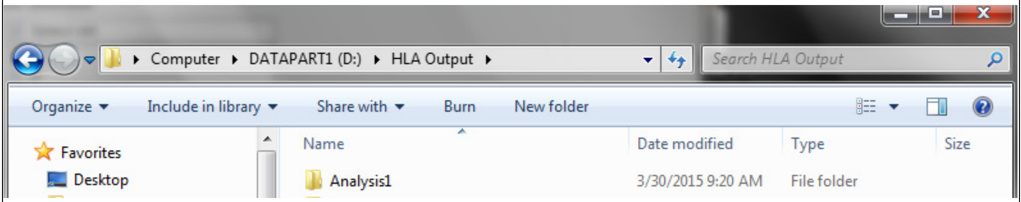
Figure 4 Enter Batch Name field



Enter Batch Name: Analysis1

After the analysis is complete, a sub-folder (inside your assigned output folder) is auto-generated and labeled using this batch name, as shown in Figure 5.

Figure 5 Example of an Output folder's Batch name/sub-folder



Name	Date modified	Type	Size
Analysis1	3/30/2015 9:20 AM	File folder	

Selecting a graph file version

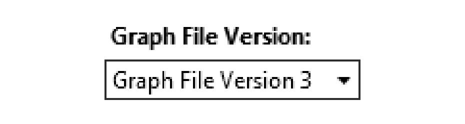
A select set of known genotypes and HLA types from a reference data set have been used to construct a haplotype graph of the extended MHC region of the genome for imputing the HLA type of an individual sample. These graphs are updated as more samples are included in the reference data set, therefore multiple versions of the graph file may be available on a given system.

For best results, it is recommended you use the latest available version of the graph file. Graph File version updates are available for download from **thermofisher.com** or email your local field support representative to request a version update.

Reference files should be updated at the beginning of a study. For consistency purposes, all samples in your study should be analyzed using the same Graph File version.

1. Click the **Graph File Version** drop-down menu to select the version you want. (Figure 6)

Figure 6 Graph File Version drop-down



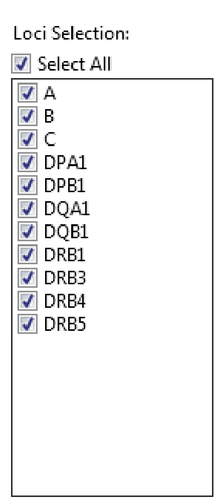
Graph File Version:

Graph File Version 3 ▾

Selecting Loci

By default, the **Select All** check box is selected and all Loci selections are auto-checked, as shown in Figure 7.

Figure 7 Loci Selection



Loci Selection:

☑ Select All

☑ A
☑ B
☑ C
☑ DPA1
☑ DPB1
☑ DQA1
☑ DQB1
☑ DRB1
☑ DRB3
☑ DRB4
☑ DRB5

To choose specific Loci from the selection list, click to uncheck the **Select All** check box, then click to check the specific Loci you want.

Running an analysis

1. Click Run Analysis.

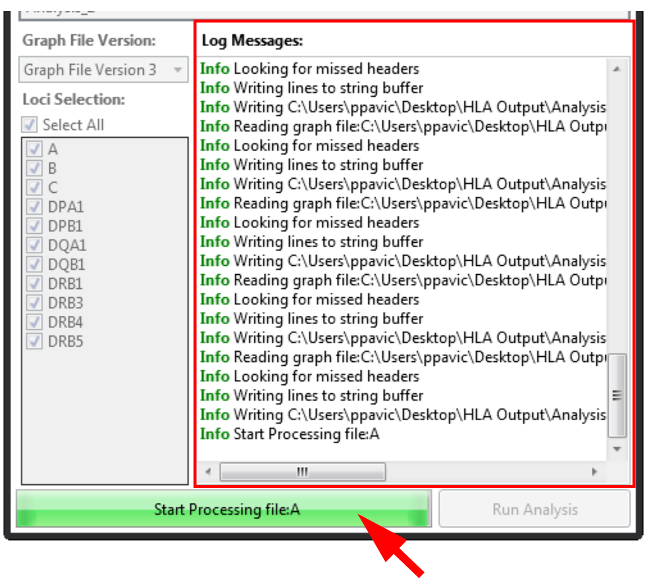
   The analysis process and report generation begins.

   Note: If duplicate IDs are detected within your Input (VCF) file, a warning message appears. Click Yes to remove the duplicate IDs. If you click No, the analysis cannot continue.

IMPORTANT! If your input and/or output location resides on a network drive, make sure the connection is reliable, as an intermittent or halted connection may cause the Viewer to inadvertently display results from a previous analysis.

The **Log Messages** pane and progress bar display analysis status, as shown in Figure 8.

Figure 8 Log Messages pane and progress bar



```
Info Looking for missed headers
Info Writing lines to string buffer
Info Writing C:\Users\ppavic\Desktop\HLA Output\Analysis
Info Reading graph file:C:\Users\ppavic\Desktop\HLA Outpu
Info Looking for missed headers
Info Writing lines to string buffer
Info Writing C:\Users\ppavic\Desktop\HLA Output\Analysis
Info Reading graph file:C:\Users\ppavic\Desktop\HLA Outpu
Info Looking for missed headers
Info Writing lines to string buffer
Info Writing C:\Users\ppavic\Desktop\HLA Output\Analysis
Info Reading graph file:C:\Users\ppavic\Desktop\HLA Outpu
Info Looking for missed headers
Info Writing lines to string buffer
Info Writing C:\Users\ppavic\Desktop\HLA Output\Analysis
Info Reading graph file:C:\Users\ppavic\Desktop\HLA Outpu
Info Looking for missed headers
Info Writing lines to string buffer
Info Writing C:\Users\ppavic\Desktop\HLA Output\Analysis
Info Start Processing file:A
```

Note: Processing times vary depending on the size of your VCF file and number of selected/checked Loci.

Viewing full reports

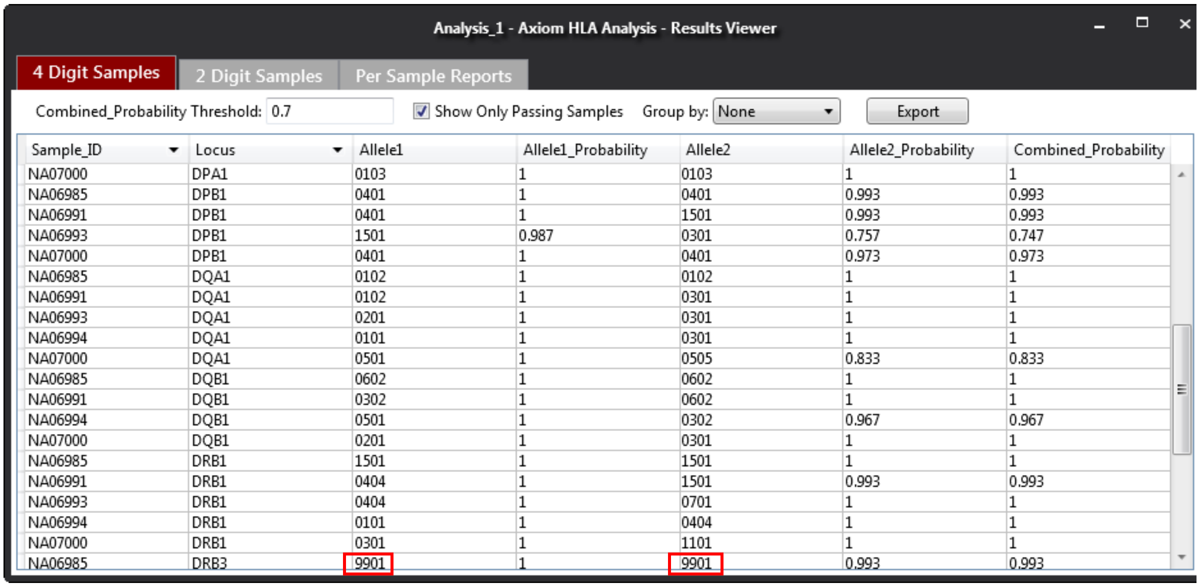
Immediately after a report has been successfully generated, the **Results Viewer** window appears and displays a **4 Digit Samples** report, as shown in Figure 9.

Each sample displays two allele calls (**Allele1** and **Allele2**) that represent the calls on the two copies of chromosome 6. The assignment of a call to one copy of the chromosome (or the other) is random and has no biological significance. The calls may switch from **Allele1** to **Allele2** between the 2 Digit and 4 Digit reports and/or between analysis runs on different workstations. This behavior is not significant.

Each call is assigned a probability score. Each score (**Allele1_Probability** and **Allele2 _Probability**) is the individual probability that the corresponding call is correct. The algorithm calls **Allele1** first, then based on that call, makes the call for **Allele2**. The **Combined_Probability** is the overall probability that both calls have been assigned correctly.

Note: In some instances, a particular loci may not be represented within the sample. If this is the case, the call is reported as **9901** (Figure 9) and its HLA type is not defined.

Figure 9 Report Options window



Combined_Probability Threshold: 0.7 ☑ Show Only Passing Samples Group by: None

Sample_ID	Locus	Allele1	Allele1_Probability	Allele2	Allele2_Probability	Combined_Probability
NA07000	DPA1	0103	1	0103	1	1
NA06985	DPB1	0401	1	0401	0.993	0.993
NA06991	DPB1	0401	1	1501	0.993	0.993
NA06993	DPB1	1501	0.987	0301	0.757	0.747
NA07000	DPB1	0401	1	0401	0.973	0.973
NA06985	DQA1	0102	1	0102	1	1
NA06991	DQA1	0102	1	0301	1	1
NA06993	DQA1	0201	1	0301	1	1
NA06994	DQA1	0101	1	0301	1	1
NA07000	DQA1	0501	1	0505	0.833	0.833
NA06985	DQB1	0602	1	0602	1	1
NA06991	DQB1	0302	1	0602	1	1
NA06994	DQB1	0501	1	0302	0.967	0.967
NA07000	DQB1	0201	1	0301	1	1
NA06985	DRB1	1501	1	1501	1	1
NA06991	DRB1	0404	1	1501	0.993	0.993
NA06993	DRB1	0404	1	0701	1	1
NA06994	DRB1	0101	1	0404	1	1
NA07000	DRB1	0301	1	1101	1	1
NA06985	DRB3	9901	1	9901	0.993	0.993

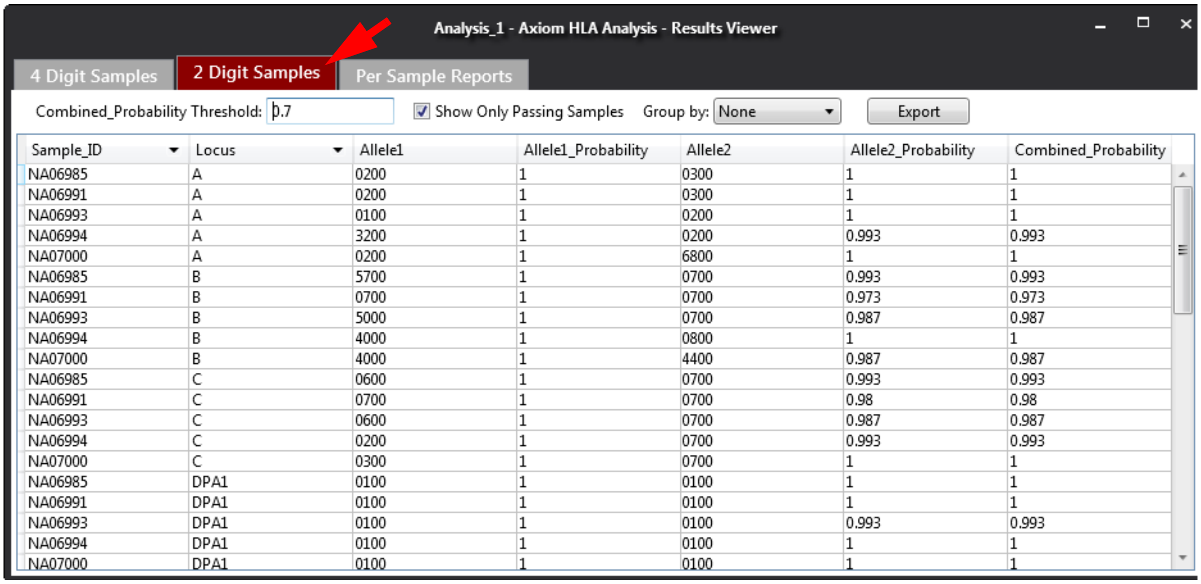
Using table data in Excel or Notepad

1. Single-click on a row or press Ctrl click, or Shift click to select multiple rows.
2. Press Ctrl c to copy your selected (highlighted) row(s).
3. Open MS Excel or MS Notepad, then press Ctrl v to paste your copied row(s).

Viewing a 2-Digit Samples report

1. Click the **2 Digit Samples** tab. (Figure 10)

Figure 10 Report Options window



Combined_Probability Threshold: 0.7 ☑ Show Only Passing Samples Group by: None

Sample_ID	Locus	Allele1	Allele1_Probability	Allele2	Allele2_Probability	Combined_Probability
NA06985	A	0200	1	0300	1	1
NA06991	A	0200	1	0300	1	1
NA06993	A	0100	1	0200	1	1
NA06994	A	3200	1	0200	0.993	0.993
NA07000	A	0200	1	6800	1	1
NA06985	B	5700	1	0700	0.993	0.993
NA06991	B	0700	1	0700	0.973	0.973
NA06993	B	5000	1	0700	0.987	0.987
NA06994	B	4000	1	0800	1	1
NA07000	B	4000	1	4400	0.987	0.987
NA06985	C	0600	1	0700	0.993	0.993
NA06991	C	0700	1	0700	0.98	0.98
NA06993	C	0600	1	0700	0.987	0.987
NA06994	C	0200	1	0700	0.993	0.993
NA07000	C	0300	1	0700	1	1
NA06985	DPA1	0100	1	0100	1	1
NA06991	DPA1	0100	1	0100	1	1
NA06993	DPA1	0100	1	0100	0.993	0.993
NA06994	DPA1	0100	1	0100	1	1
NA07000	DPA1	0100	1	0100	1	1

Grouping your full report samples

Click the **Group by** drop-down menu (Figure 11) to group your completed analysis data by either **Sample_ID** or **Locus**.

Figure 11 Group by drop-down

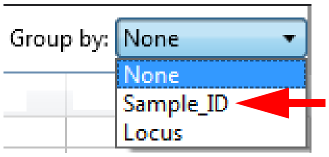
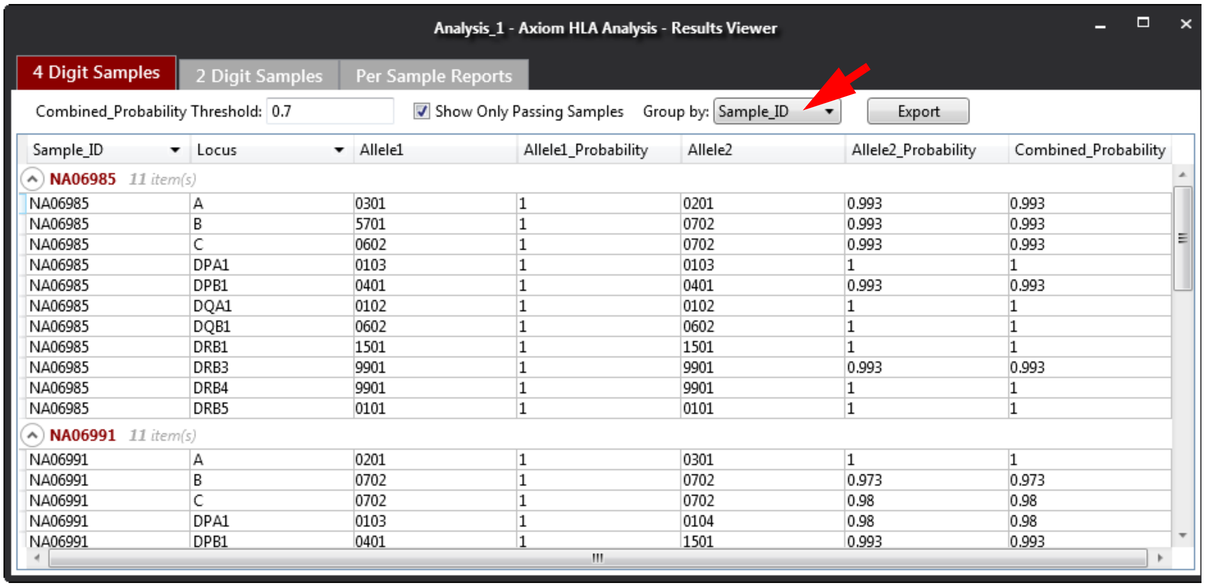


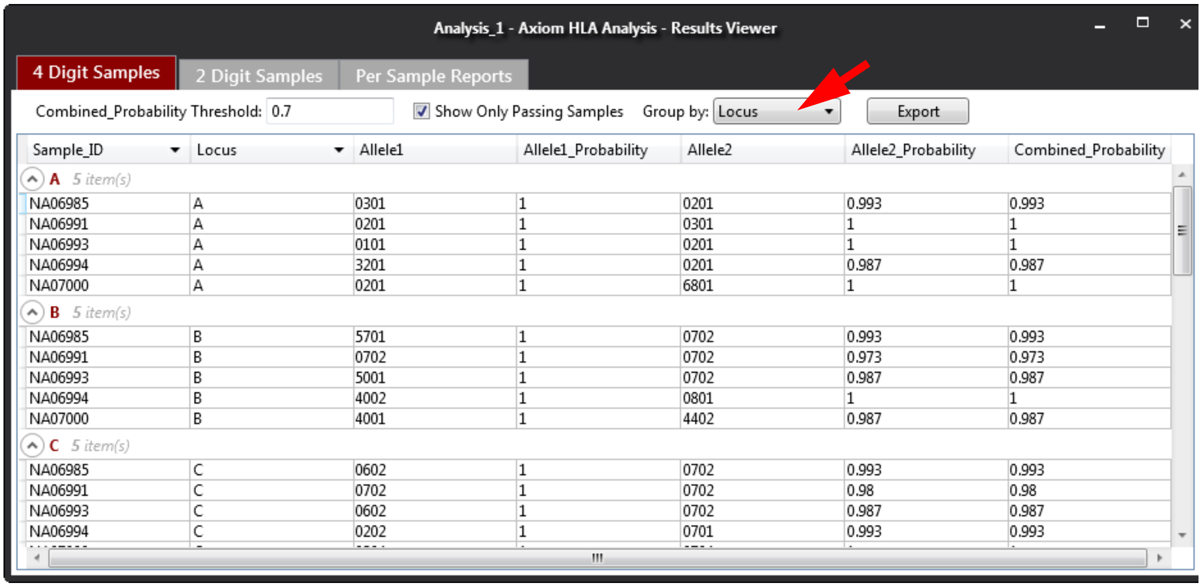
Figure 12 Group by Sample ID example



Combined_Probability Threshold: 0.7 ☑ Show Only Passing Samples Group by: Sample_ID

Sample_ID	Locus	Allele1	Allele1_Probability	Allele2	Allele2_Probability	Combined_Probability
NA06985 *11 item(s)*						
NA06985	A	0301	1	0201	0.993	0.993
NA06985	B	5701	1	0702	0.993	0.993
NA06985	C	0602	1	0702	0.993	0.993
NA06985	DPA1	0103	1	0103	1	1
NA06985	DPB1	0401	1	0401	0.993	0.993
NA06985	DQA1	0102	1	0102	1	1
NA06985	DQB1	0602	1	0602	1	1
NA06985	DRB1	1501	1	1501	1	1
NA06985	DRB3	9901	1	9901	0.993	0.993
NA06985	DRB4	9901	1	9901	1	1
NA06985	DRB5	0101	1	0101	1	1
NA06991 *11 item(s)*						
NA06991	A	0201	1	0301	1	1
NA06991	B	0702	1	0702	0.973	0.973
NA06991	C	0702	1	0702	0.98	0.98
NA06991	DPA1	0103	1	0104	0.98	0.98
NA06991	DPB1	0401	1	1501	0.993	0.993

Figure 13 Group by Locus example



Combined_Probability Threshold: 0.7 ☑ Show Only Passing Samples Group by: Locus

Sample_ID	Locus	Allele1	Allele1_Probability	Allele2	Allele2_Probability	Combined_Probability
A *5 item(s)*						
NA06985	A	0301	1	0201	0.993	0.993
NA06991	A	0201	1	0301	1	1
NA06993	A	0101	1	0201	1	1
NA06994	A	3201	1	0201	0.987	0.987
NA07000	A	0201	1	6801	1	1
B *5 item(s)*						
NA06985	B	5701	1	0702	0.993	0.993
NA06991	B	0702	1	0702	0.973	0.973
NA06993	B	5001	1	0702	0.987	0.987
NA06994	B	4002	1	0801	1	1
NA07000	B	4001	1	4402	0.987	0.987
C *5 item(s)*						
NA06985	C	0602	1	0702	0.993	0.993
NA06991	C	0702	1	0702	0.98	0.98
NA06993	C	0602	1	0702	0.987	0.987
NA06994	C	0202	1	0701	0.993	0.993

Note: The probability scores may vary slightly between different workstations. This is normal, as the imputation model uses a random seed in the inference step which may result in insignificant differences in the probability.

Changing the default combined probability threshold

1. Click inside the **Combined_Probability Threshold** text field to enter a different value.

   Your newly entered value is instantly reflected within the table. Calls that do not pass your entered threshold value are not displayed.

Showing only passing samples

By default, only the samples that have passed your filter criteria are displayed. Uncheck the **Show Only Passing Samples** check box to show all samples.

Exporting/saving a full report

1. Click .

   An Explorer window appears.
2. Navigate to an easily accessible save location, enter a filename, then click **Save**.

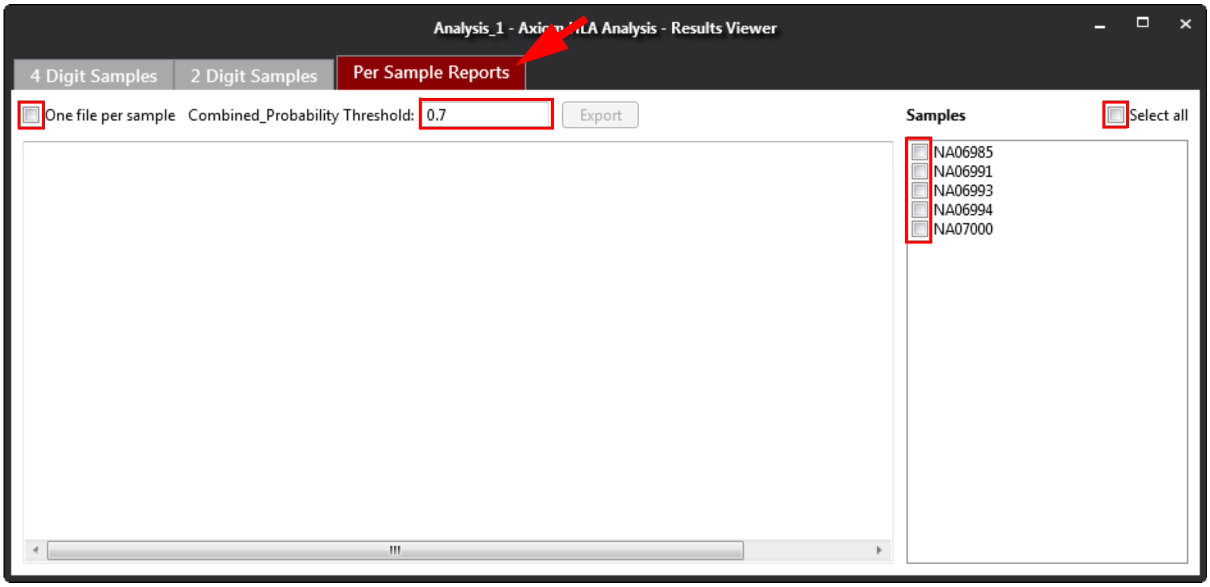
   An Information window appears confirming your full report has been saved successfully.
3. Click **OK** to acknowledge the message.
4. Use Windows Explorer to navigate to the location of your newly exported/saved report.
5. Use Excel, MS Word, WordPad, or Notepad to open the tab-delimited.txt formatted report file.

   Save or print the report as you normally would or click **File** → **Save As** to save it with a different file extension (for use in other applications).

Viewing per sample reports

1. Click the **Per Sample Reports** tab. (Figure 14)

Figure 14 Per Sample Reports window tab



□ One file per sample Combined_Probability Threshold: 0.7

□ Select all

2. Optional: Click the **One file per sample** check box to generate individual per sample reports. Leave this check box unchecked to generate a single (combined) per sample report.

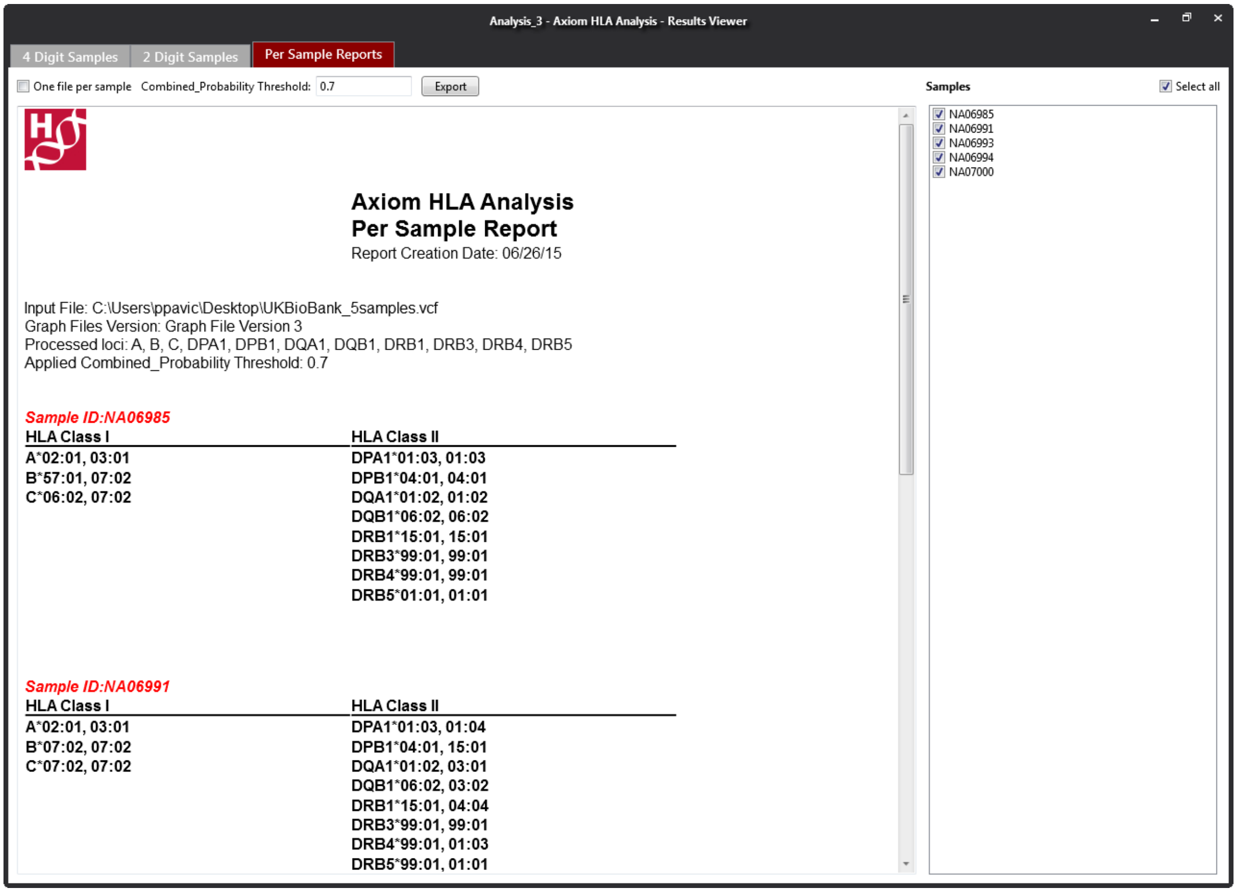
   Note: The When the **One file per sample** check box is checked, only the first sample is displayed in the Results Viewer, as shown in Figure 16 on page 13.

3. Optional: Click inside the **Combined_Probability Threshold** text field to enter a different value.

4. In the **Samples** window pane (Figure 14), click each check box next to the listed Sample you want to include in a per sample report or click the **Select all** check box to include all samples in the per sample report.

   Your **Per Sample Report(s)** appear within the viewer, as shown in Figure 15.

Figure 15 Per Sample Report Viewer example



□ One file per sample Combined_Probability Threshold: 0.7

☑ Select all

☑ NA06985
☑ NA06991
☑ NA06993
☑ NA06994
☑ NA07000

Axiom HLA Analysis
Per Sample Report
Report Creation Date: 06/26/15

Input File: C:\Users\ppavic\Desktop\UKBioBank_5samples.vcf
Graph Files Version: Graph File Version 3
Processed loci: A, B, C, DPA1, DPB1, DQA1, DQB1, DRB1, DRB3, DRB4, DRB5
Applied Combined_Probability Threshold: 0.7

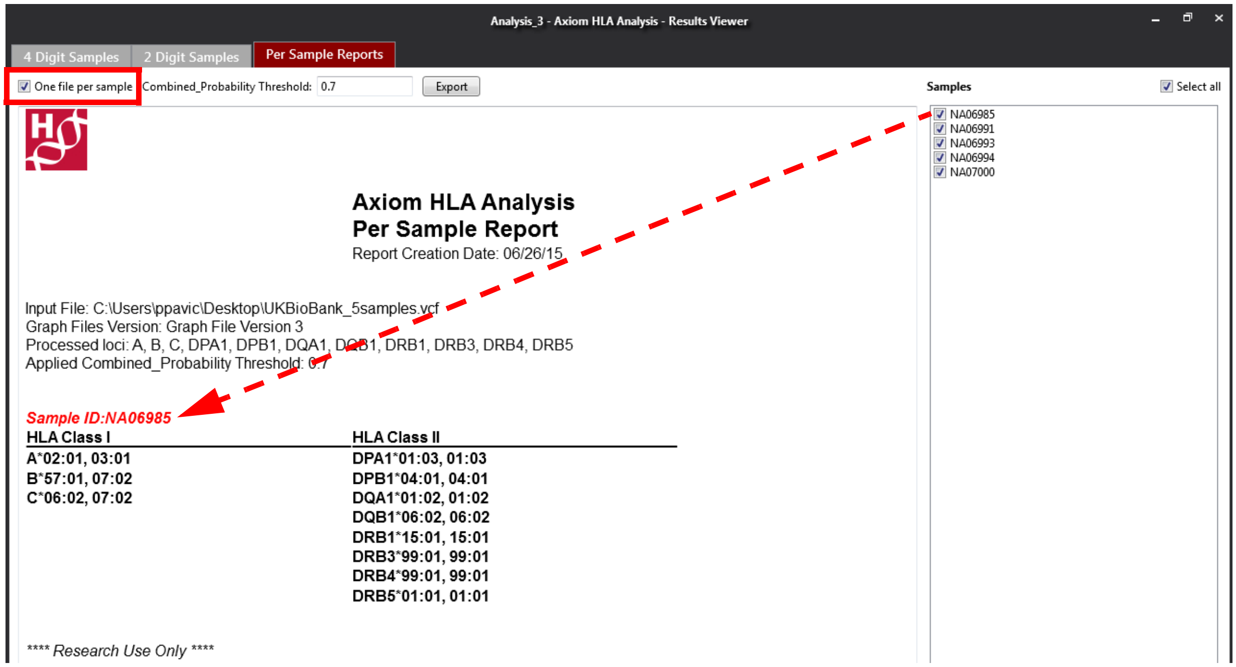
Sample ID:NA06985

HLA Class I	HLA Class II
A*02:01, 03:01	DPA1*01:03, 01:03
B*57:01, 07:02	DPB1*04:01, 04:01
C*06:02, 07:02	DQA1*01:02, 01:02
	DQB1*06:02, 06:02
	DRB1*15:01, 15:01
	DRB3*99:01, 99:01
	DRB4*99:01, 99:01
	DRB5*01:01, 01:01

Sample ID:NA06991

HLA Class I	HLA Class II
A*02:01, 03:01	DPA1*01:03, 01:04
B*07:02, 07:02	DPB1*04:01, 15:01
C*07:02, 07:02	DQA1*01:02, 03:01
	DQB1*06:02, 03:02
	DRB1*15:01, 04:04
	DRB3*99:01, 99:01
	DRB4*99:01, 01:03
	DRB5*99:01, 01:01

Figure 16 One file per sample Report Viewer example



☑ One file per sample Combined_Probability Threshold: 0.7

☑ Select all

☑ NA06985
☑ NA06991
☑ NA06993
☑ NA06994
☑ NA07000

Axiom HLA Analysis
Per Sample Report
Report Creation Date: 06/26/15

Input File: C:\Users\ppavic\Desktop\UKBioBank_5samples.vcf
Graph Files Version: Graph File Version 3
Processed loci: A, B, C, DPA1, DPB1, DQA1, DQB1, DRB1, DRB3, DRB4, DRB5
Applied Combined_Probability Threshold: 0.7

Sample ID:NA06985

HLA Class I	HLA Class II
A*02:01, 03:01	DPA1*01:03, 01:03
B*57:01, 07:02	DPB1*04:01, 04:01
C*06:02, 07:02	DQA1*01:02, 01:02
	DQB1*06:02, 06:02
	DRB1*15:01, 15:01
	DRB3*99:01, 99:01
	DRB4*99:01, 99:01
	DRB5*01:01, 01:01

***** Research Use Only *****

Exporting and viewing per sample reports

1. Click Export .

   A **Select Folder** window appears.
2. Click to highlight a folder or click the window's New folder button to create a new folder.
3. Click **Select Folder**.

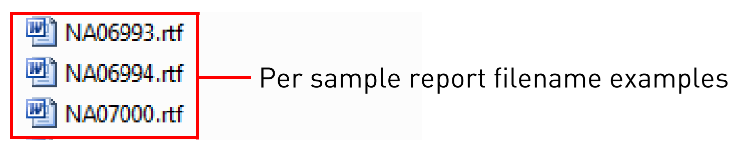
   A window appears displaying your RTF formatted report files.
4. Double-click on the RTF report file you want to view.

   The report opens.
5. Optional: Click **File** → **Save As** to save the report with a different file extension (for use in other applications).

Default Per Sample Report Filenames

Each per sample report is auto-assigned a Sample ID-based filename, as shown in Figure 17.

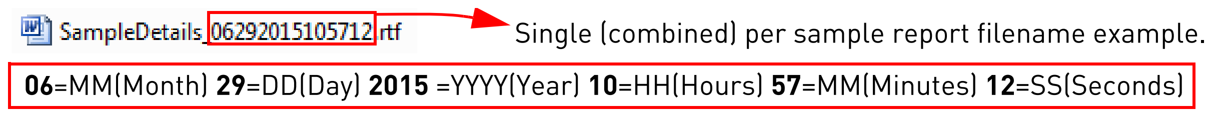
Figure 17 Report filename examples



NA06993.rtf
NA06994.rtf
NA07000.rtf

A single (multiple per sample) report is auto-assigned the filename **SampleDetails** and a time-stamp, as shown in Figure 18.

Figure 18 Single Per Sample Report filename example



SampleDetails_06292015105712.rtf

06=MM(Month) **29**=DD(Day) **2015** =YYYY(Year) **10**=HH(Hours) **57**=MM(Minutes) **12**=SS(Seconds)

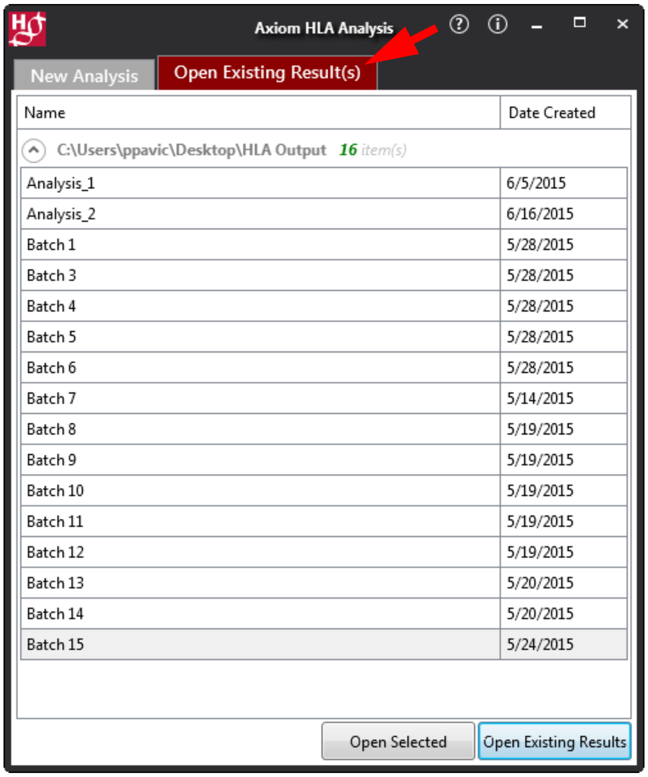
Viewing completed results

All completed results reside in the Open Existing Result(s) window tab.

1. Click the **Open Existing Result(s)** tab.

   The Open Existing Result(s) window tab appears. (Figure 19)

Figure 19 Open Existing Result window tab



Name	Date Created
C:\Users\ppavic\Desktop\HLA Output 16 item(s)	
Analysis_1	6/5/2015
Analysis_2	6/16/2015
Batch 1	5/28/2015
Batch 3	5/28/2015
Batch 4	5/28/2015
Batch 5	5/28/2015
Batch 6	5/28/2015
Batch 7	5/14/2015
Batch 8	5/19/2015
Batch 9	5/19/2015
Batch 10	5/19/2015
Batch 11	5/19/2015
Batch 12	5/19/2015
Batch 13	5/20/2015
Batch 14	5/20/2015
Batch 15	5/24/2015

Viewing a result

1. Single-click on a result you want to view to highlight it, then click Open Selected.

   Your selected result appears in the **Results Viewer** window.

   To use the viewer, see "Viewing full reports" on page 8 or "Viewing per sample reports" on page 12.

Viewing an existing result

Do the following to view an existing result not displayed on the Open Existing Result(s) window tab:

1. Click Open Existing Results.

   A **Please select folder** window appears.

2. Use Windows Explorer to navigate to the folder containing the existing result you want to view

3. Single-click to highlight its folder, then click Select Folder.

   The result is now listed on the **Open Existing Result(s)** window tab.

4. Single-click to highlight the result set you want to view, then click Open Selected.

   Your result appears in the **Results Viewer** window.

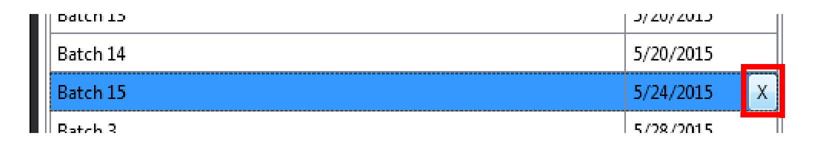
   To use the viewer, see "Viewing full reports" on page 8 or "Viewing per sample reports" on page 12.

Removing a listed result set

1. Click to highlight the result set you want to remove, then click X. (Figure 20)

   The result set is now removed from the **Open Existing Result(s)** window tab list, but is not deleted from the folder it resides in.

Figure 20 Removing a result set



Batch 13	5/20/2015	
Batch 14	5/20/2015	
Batch 15	5/24/2015	X
Batch 3	5/28/2015	

Ways to generate a VCF file

Using Axiom Analysis Suite

At the SNP Summary Table window tab, click **Export** → **Export Genotyping Data**, then refer to the Axiom Analysis Suite User Guide (P/N 703307) for instructions on how to generate a VCF file.

Using a command line

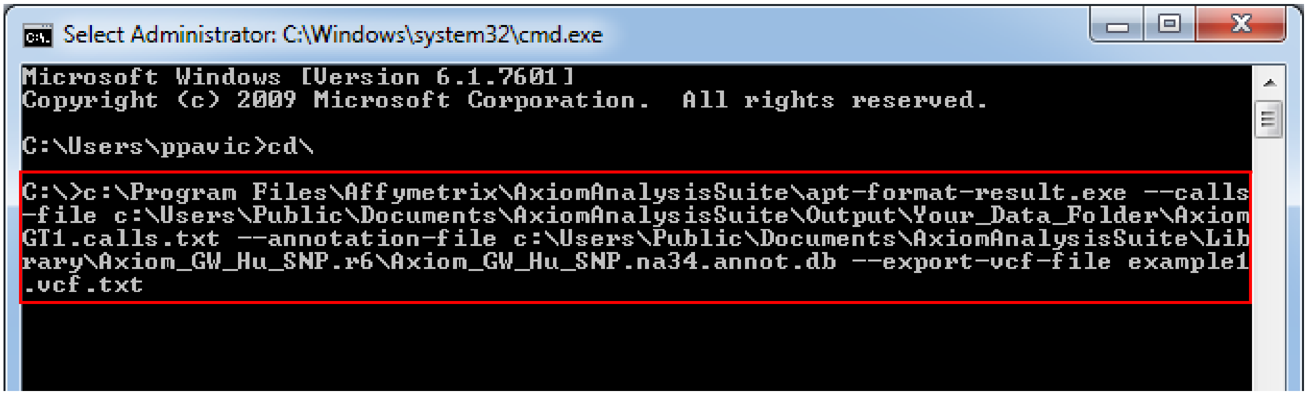
The command line application **apt-format-result.exe** included in Affymetrix Power Tools (APT) 1.17 can be used to generate a VCF file.

Make sure your command line is formatted, as follows:

apt_format-results.exe --calls-file <full path to your calls> -annotation file <full path to annot.db file>

Example command line: (Figure 21)

Figure 21 Per Sample example



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ppavic>cd\

C:\>c:\Program Files\Affymetrix\AxiomAnalysisSuite\apt-format-result.exe --calls
-file c:\Users\Public\Documents\AxiomAnalysisSuite\Output\Your_Data_Folder\Axiom
GT1.calls.txt --annotation-file c:\Users\Public\Documents\AxiomAnalysisSuite\Lib
rary\Axiom_GW_Hu_SNP.r6\Axiom_GW_Hu_SNP.na34.annot.db --export-vcf-file example1
.vcf.txt
```

Note: The calls.txt is produced by Genotyping Console, Axiom Analysis Suite, and Affymetrix Power Tools (apt-probeset-genotype.exe and apt-genotype-axiom.exe). Refer to the User Manuals of these applications for the specific location of the calls.txt file.

Supplemental information

The following pages are from external sources and have been added to this User Guide for your reference.

OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Multi-Population Classical HLA Type Imputation

Alexander Dilthey[1,2]☯*, Stephen Leslie[3]☯, Loukas Moutsianas[2], Judong Shen[4], Charles Cox[5], Matthew R. Nelson[4], Gil McVean[1,2]

1 Department of Statistics, University of Oxford, Oxford, United Kingdom, **2** Wellcome Trust Centre for Human Genetics, Oxford, United Kingdom, **3** Murdoch Childrens Research Institute, Royal Children's Hospital, Parkville, Victoria, Australia, **4** Quantitative Sciences, GlaxoSmithKline, Research Triangle Park, North Carolina, United States of America, **5** Quantitative Sciences, GlaxoSmithKline, Stevenage, United Kingdom

Abstract

Statistical imputation of classical HLA alleles in case-control studies has become established as a valuable tool for identifying and fine-mapping signals of disease association in the MHC. Imputation into diverse populations has, however, remained challenging, mainly because of the additional haplotypic heterogeneity introduced by combining reference panels of different sources. We present an HLA type imputation model, HLA*IMP:02, designed to operate on a multi-population reference panel. HLA*IMP:02 is based on a graphical representation of haplotype structure. We present a probabilistic algorithm to build such models for the HLA region, accommodating genotyping error, haplotypic heterogeneity and the need for maximum accuracy at the HLA loci, generalizing the work of Browning and Browning (2007) and Ron et al. (1998). HLA*IMP:02 achieves an average 4-digit imputation accuracy on diverse European panels of 97% (call rate 97%). On non-European samples, 2-digit performance is over 90% for most loci and ethnicities where data available. HLA*IMP:02 supports imputation of HLA-DPB1 and HLA-DRB3-5, is highly tolerant of missing data in the imputation panel and works on standard genotype data from popular genotyping chips. It is publicly available in source code and as a user-friendly web service framework.

Citation: Dilthey A, Leslie S, Moutsianas L, Shen J, Cox C, et al. (2013) Multi-Population Classical HLA Type Imputation. PLoS Comput Biol 9(2): e1002877. doi:10.1371/journal.pcbi.1002877

Editor: Sharon Browning, University of Washington, United States of America

Received July 27, 2012; **Accepted** November 16, 2012; **Published** February 14, 2013

Funding: We acknowledge funding from Studienstiftung des deutschen Volkes, EPSRC, The Leverhulme Trust, HFSP, The Wellcome Trust, and GlaxoSmithKline. We also acknowledge funding from the Wellcome Trust core grant. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: dilthey@well.ox.ac.uk

☯ These authors contributed equally to this work.

Introduction

Statistical imputation of classical human leukocyte antigen (HLA) alleles from SNP genotypes in case-control studies has become established as a valuable tool for identifying and fine-mapping signals of disease association in the MHC. Application of the HLA type imputation framework HLA*IMP [1,2] has, for example, helped to fine-map secondary HLA-based risk effects in multiple sclerosis [3], contributed to characterizing an HLA-related gene-gene interaction in psoriasis [4], and was essential in refuting a suspected strong HLA contribution to childhood B-cell precursor acute lymphoblastic leukaemia [5]. Classical HLA allele imputation has, in other settings, been used to identify particular amino acids within classical peptides contributing to disease risk [6].

Classical HLA allele imputation is complicated by hyperpolymorphism (*HLA-B* , for example, has dozens of common alleles and >2000 rare alleles) and the complex haplotype structure of the HLA region, justifying the development of specialized imputation machinery. Linkage disequilibrium (LD) between loci usually declines with distance, as LD is broken down by recombination. In the HLA, however, this is not always empirically true. Many comparatively distant SNPs carry information on the allelic state of the classical HLA genes [7]. Fully capturing this information is not trivial. For example, a commonly used model in statistical genetics, the Li and Stephens approximation [8], does not allow for explicit modelling of long-distance LD relationships due to its reliance on a first order Markov chain. HLA*IMP therefore uses a particular formulation of the Li and Stephens approximation that assigns equal weight to all selected SNPs irrespective of distance from the classical locus of interest [2]. We have since demonstrated (e.g., [1]) that this formulation leads to highly accurate HLA type imputations, at least when reference and imputation panel are derived from the same population.

For the increasingly important use case of multi-population studies (where the reference and analysis panels consist of samples taken from multiple, possibly diverse, populations), HLA type imputation has, however, remained challenging: Imputation accuracy is limited by the extent to which the reference panel captures the diversity of the target population and current methods typically rely on single-source reference panels of Northern European origin [1,9].

The obvious solution, successfully applied in SNP genotype imputation [10,11,12], is to make use of diverse multi-population reference panels. However, an additional challenge of multi-population classical HLA type imputation is that single HLA alleles can appear on multiple SNP haplotype backgrounds [7], a phenomenon we refer to as "haplotypic heterogeneity". Moreover, genetic data obtained from multiple data sets from different

Author Summary

The human leukocyte antigen (HLA) proteins influence how pathogens and components of body cells are presented to immune cells. It has long been known that they are highly variable and that this variation is associated with differential risk for autoimmune and infectious diseases. Variant frequencies differ substantially between and even within continents. Determining HLA genotypes is thus an important part of many studies to understand the genetic basis of disease risk. However, conventional methods for HLA typing (e.g. targeted sequencing, hybridisation, amplification) are typically laborious and expensive. We have developed a method for inferring an individual's HLA genotype based on evaluating genetic information from nearby variable sites that are more easily assayed, which aims to integrate heterogeneous data. We introduce two key innovations: we allow for single HLA types to appear on heterogeneous backgrounds of genetic information and we take into account the possibility of genotyping error, which is common within the HLA region. We show that the method is well-suited to deal with multi-population datasets: it enables integrated HLA type inference for individuals of differing ancestry and ethnicity. It will therefore prove useful particularly in international collaborations to better understand disease risks, where samples are drawn from multiple countries.

populations is likely to contain systematic genotyping artefacts. Here we present HLA*IMP:02, an HLA type imputation method that is particularly aimed at inference in multi-population and multi-ethnicity settings. That is, it is designed to accommodate both haplotypic heterogeneity and genotyping error.

Inference under HLA*IMP:02 is based on a graphical model of the haplotype structure of the MHC region. We motivate this choice by restating an observation made by Browning and Browning [13]: Graphical haplotype models are well-suited to model LD relationships spanning different scales of distance ("variable-length Markov chains"), which fits with the HLA region's empirically observed LD structure. We present an algorithm to build such models from a set of reference genotype data. The main design features of the algorithm are that it takes into account haplotype uncertainty introduced by potential genotyping error, that it allows for haplotypic heterogeneity and that it tailors the graphs to make them maximally informative about the allelic state of the HLA loci. Our algorithm can be viewed as a probabilistic generalization of the works of Browning and Browning [14]. Compared with HLA*IMP, HLA*IMP:02 also offers a couple of practical advantages: it is highly tolerant of missing data in the inference panel and supports imputation of HLA-DPB1 and HLA-DRB3-5.

It is instructive to explicitly consider how the design of HLA*IMP:02 leads to an improved ability to deal with heterogeneous data, as compared to HLA*IMP:

- Data representation: HLA*IMP:02 builds a combined locus-specific haplotype graph model of the whole dataset. In HLA*IMP, in contrast, reference genotype data is phased and separated by HLA alleles. All further steps are based on these allelic groups (one for each HLA allele in the reference panel). This design prevents HLA*IMP from sharing SNP haplotype information across haplotypes carrying different HLA alleles.
- Maximising imputation performance: HLA*IMP:02, uses all available SNPs in the HLA region. However, while building the haplotype graph no two internal haplotype states that exhibit different association patterns to HLA alleles are combined, thus maintaining accuracy specifically for HLA allele prediction. HLA*IMP, in contrast, carries out a process of SNP selection, identifying SNPs in the region that are informative for accurate prediction of HLA types. Finding a set of consistently informative SNPs becomes increasingly difficult as the degree of stratification in the reference panel increases.
- Inference model: In the haplotype-graph approach of HLA*IMP:02, haplotypes are not grouped in advance. If an allele appears on multiple SNP haplotypes, there will be multiple paths through the graph leading to the allele. Inference is based on comparing the likelihoods of all possible paths. Ambiguity therefore typically only arises if two or more alleles share the same SNP haplotypes, but not if one allele appears on more than one background. Additional heterogeneity in the reference panel (characterized by alleles appearing on more than one unique background) does not decrease the model's ability to correctly infer HLA genotypes. HLA*IMP, in contrast, appears to suffer decreased performance in both scenarios (one allele/multiple backgrounds, multiple alleles/one background). This is perhaps because inference under HLA*IMP is based on finding the most similar *group* of haplotypes (implemented through a particular formulation of the Li and Stephens [8] Hidden Markov Model, HMM). Additional heterogeneity in an allele's SNP background necessarily reduces group-wise average similarity and dilutes the model's ability to correctly infer HLA genotypes.

We carry out three experiments to investigate the performance of HLA*IMP:02 on reference panels of varying heterogeneity. In the first experiment, we apply HLA*IMP:02 to a homogeneous (predominately British) reference panel and show that it performs as well as HLA*IMP in this baseline scenario. In the second experiment, we demonstrate that HLA*IMP:02 achieves high imputation accuracy at 4-digit HLA type resolution (reflecting primary sequence of the HLA proteins) when applied to an integrated cross-European reference panel, clearly outperforming HLA*IMP. In the third experiment, we use a highly heterogeneous multi-ethnic reference panel to impute HLA genotypes of Asian, African-American, African, European and Hispanic individuals. We show that accuracy for the European individuals remains essentially unchanged by making the reference panel more heterogeneous and that the model achieves high imputation accuracy for the other ethnicities at 2-digit resolution, which reflects the serological properties of the HLA alleles (see Subsection "Validation" for a precise definition in our context).

Materials and Methods

HLA*IMP:02

We use an acyclic probabilistic finite automaton ("haplotype graph", see Figure 1) to represent haplotype structure in the HLA region [14,15]. The haplotype graph describes the haplotype structure of SNPs around the classical HLA loci. In Figure 1, each possible path through the graph also passes through an edge carrying an HLA allele, and therefore specifies a corresponding HLA genotype. The likelihood of any particular path depends on the branching structure of the graph (as specified by the probabilities on the edges in Figure 1) as well as on the observed SNP genotypes from an individual that we want to make inference for. For example, if we observe the SNP genotypes `TTA?TA` (the question mark stands for the unknown HLA allele, and we only consider the haploid case here for simplicity), the likelihood of the path passing through the bottom nodes (and the 1501 allele) is 0.2, and the likelihood of all others paths is 0 (not allowing for any deviations from the edge labels for the sake of this argument). For `ATA?GA`, the case is also clear: `0301` is the only possible allele. If we now change the second-last genotype to `T` (yielding `ATA?TA`), there are two possible paths. The one passing through `1501` has a likelihood of 0.012, and the other one passing through `0301` has a likelihood of 0.056. Conditional on the observed SNP genotypes, `1501` is therefore approximately twice as probable as `0301`. Changing the second and third genotypes would not influence this result (which relates back to our introductory comments on the variable length of captured LD relationships: the first position influences inference, the second and third do not).

In order to use haplotype graphs for imputation, there are two general problems to address: how to construct a haplotype graph from a set of reference data, and how to use an existing graph for imputing the genotype of an additional individual. Methods to construct and use haplotype graph-like objects from a set of reference data were discussed by Ron et al. [15] and introduced into the field of statistical genetics by Browning [13] and Browning and Browning [14]. The work we present here can be viewed as a probabilistic generalization of the works of Ron et al. [15] and Browning and Browning [14]. To use haplotype graph models specifically for HLA type inference, we have developed solutions to two related tasks: how to build a haplotype graph model from the reference panel allowing for errors in SNP genotype data and haplotypic heterogeneity and how to boost accuracy for HLA allele imputations. A full and formal description of the HLA*IMP:02 algorithm can be found in the Supporting Text S1. Here we provide outline of our algorithm and the inference process, highlighting where we generalized and extended previous approaches.

Constructing a haplotype graph from a set of reference data (including both SNP and HLA genotypes) is an iterative process, consisting, as in BEAGLE, of three main steps:

- Initialization: for each individual, populate the set H of current haplotype estimates by sampling from the uniform distribution over all genotype-consistent haplotype pairs. In contrast to BEAGLE, we preserve missing data in the generated haplotype pairs.
- Probabilistic graph construction: build a haplotype graph object from the set H of current haplotype estimates. Each element in H corresponds to one path through the graph which is going to be constructed. We define a probability distribution over possible paths for each element in H and probabilistically attach the elements in H to nodes in the graph. This enables us to allow for genotyping errors and missing data in H and puts some part of the probability mass of similar haplotypes on the same nodes, even if they differ in single positions (by setting the probability of genotyping error to 0, one obtains the deterministic BEAGLE/Ron et al. [15] mode of haplotype propagation through the graph). In the process of building the graph, we collapse similar nodes for reasons of parsimony and computational efficiency. In defining node similarity, we introduce criteria that relate to each node's pattern of association with the HLA loci along the graph, and prevent collapsing two nodes that exhibit differing patterns of LD with HLA alleles (by setting the set of the loci that these additional criteria apply to the empty set, one obtains the conventional similarity criterion from BEAGLE/Ron et al. [15]).
- Resampling: Construct the diploid HMM induced by the constructed haplotype graph and re-populate H. If a predefined number of iterations has not been exceeded, fit this HMM to the reference genotype data, re-populate H with haplotype samples from the HMM (imputing missing data) and go to step 2. Like Browning and Browning [16], we use an HMM that allows for genotyping error.

The HMM resulting from the final iteration is used to generate HLA type estimates for all following imputation operations (BEAGLE, in contrast, builds joint haplotype graphs of imputation and reference panels, and carries out imputation as part of this procedure, which requires special measures for assuring convergence if the joint set is dominated by samples from the imputation dataset).

Availability, Performance, Usability

Source code for HLA*IMP:02 is available from http://oxfordhla.well.ox.ac.uk (free for academic use). Compiling and

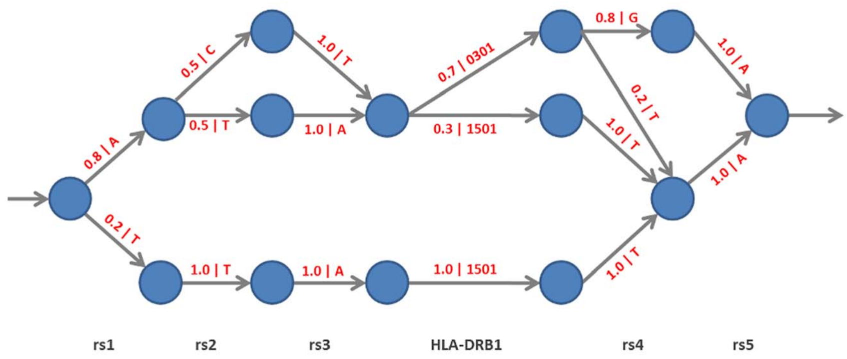


Figure 1. Features of haplotype graph models. Illustration of the features of haplotype graph models. Haplotype graphs are a subclass of connected directed graphs and belong to the class of acyclic probabilistic finite automata. Their most important properties are illustrated here: 1) They are leveled, i.e. each vertex v has an associated positive number 1, and all edges emanating from v at level l lead to a vertex at level $l+1$ and represent the same genetic locus. Vertices at level T are final vertices with no outgoing edges, and there is a path from every vertex in the graph to one of the final vertices. 2) Edges carry "emission symbols" which are emitted when an edge is traversed (in the figure: the symbols after the "|" character adjacent to the edges), and there are no two edges emanating from the same vertex which carry the same symbol. 3) Each vertex has an edge probability distribution over its attached edges (in the figure: the numbers in front of the "|" character adjacent to the edges), according to which an edge is selected conditional in being at that vertex.
doi:10.1371/journal.pcbi.1002877.g001

running the program requires a standard UNIX server environment (ideally with multiple CPU cores and ≥ 64 GB RAM).

To give an idea of the expected runtime, producing the graph for *HLA-A* for the first experiment presented in the "Results" section took approximately 137 CPU hours (user plus system time for a single CPU; the program supports parallelization via openMP, so that the actual runtime on modern multi-CPU systems is much lower); carrying out inference for a single individual required approximately 4 CPU seconds (user plus system time).

Like HLA*IMP, HLA*IMP:02 is also available as a front-end/back-end web service that integrates data preparation, QC and imputation. Figure 2 shows the steps typically required to produce HLA type imputations, starting from SNP genotypes (for example in PLINK [17], CHIAMO [18] or VCF formats). The system supports virtually all currently employed genotyping platforms, including genotyping arrays from Affymetrix, Illumina, and the Immunochip. The front-end converts genotype data into the format used by HLA*IMP:02, carries out quality control based on data completeness and aligns SNP genotypes to the positive strand (as defined in HapMap). All output data from the front-end can be directly uploaded to the HLA*IMP:02 server. Run in standard mode, the HLA*IMP:02 back-end will also produce allele- and locus-specific cross-validation estimates of accuracy, specific to the SNPs available in the user dataset. To ensure data protection and security, sample identifiers have to be anonymized prior to submission. The server stores all user data in a specially protected area, with no read access for the normal web server processes. Upon completion of an imputation job, the server generates a secondary access key, which is directly sent to the user; only the combination of access key and user account password will enable access to the imputation results.

HLA*IMP:01

We compare the performance of HLA*IMP:02 to HLA*IMP, which we refer to as "HLA*IMP:01" for clarity. HLA*IMP:01 has been described elsewhere [1,2]. Windows of 400 SNPs around the classical HLA loci and population prior frequencies, estimated from the reference panel, for classical HLA alleles were found to give good results, and these settings are identical to those used by the Internet implementation of HLA*IMP (http://oxfordhla.well.ox.ac.uk) and those used for recent genome-wide association studies [3,4,19].

Validation

We validate HLA type imputations at the genotype level in a locus-specific manner, i.e. compare two unordered sets with two elements each for each individual and locus, one set (I) representing the imputation results and the other (L) containing the lab-derived types. We only consider individuals who carry two HLA alleles typed at 4-digit resolution at the locus under validation or one allele at 4-digit resolution and one missing allele. For 2-digit (serological properties of the HLA alleles) validation, we consider the same individuals, but we set the last 2 digits of each HLA allele to '00' (this will lead to an underestimate of accuracy in some cases, as there are some serologically defined 2-digit allele groups that map to more than one pair of leading two digits). We may or may not apply a posterior probability call threshold T on the per-allele level (see Section "HLA type inference" of the Supporting Text S1 for a description of how we calculate allele-specific posterior probabilities) to our imputations before validating.

If there is no missing data in L, there are three possible cases:

- 0 imputations left after thresholding: we count 0 correctly imputed alleles out of 0.
- 1 imputation (I_1) left after thresholding: we count 1 correctly imputed alleles out of 1 if $I_1 \in L$, otherwise 0 out of 1.
- 2 imputations left after thresholding: we count 0 correct imputations out of 2 if $(I_1 \notin L) \wedge (I_2 \notin L)$, 1 out of 2 if $(I_1 \in L) \veebar (I_2 \in L)$, 2 out of 2 otherwise. ($\veebar$ is the "exclusive OR" operator, which is true if and only if exactly one of the arguments is true).

If $L = \{\text{missing}, A\}$ (i.e. only one allele has been typed), there are also three possible cases:

- 0 imputations left after thresholding: we count 0 correctly imputed alleles out of 0.
- 1 imputation (I_1) left after thresholding: we count 1 correctly imputed alleles out of 1 if $I_1 = A$, otherwise 0 out of 1.
- 2 imputations left after thresholding: we count 1 correct imputations out of 1 if $I_1 = A$ or $I_2 = A$ or both.

In terms of thresholding strategies, we use either no threshold; or a threshold of T = 0.7 for both models; or a threshold of T = 0.7 for HLA*IMP:01 (as recommended in Dilthey et al. [1]) and a threshold matched to obtain equal call rates for HLA*IMP:02. The last strategy is only employed to ensure comparability of

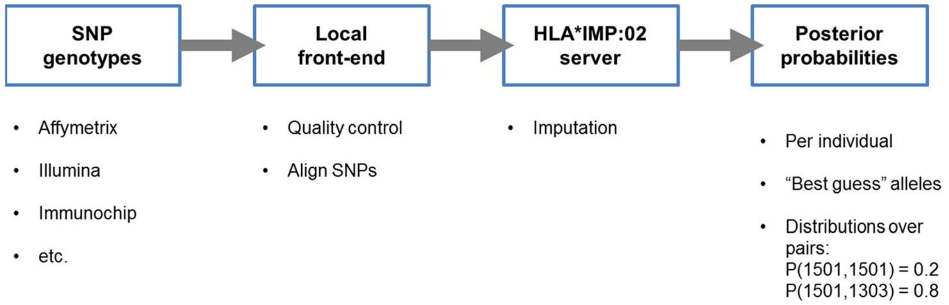


Figure 2. Standard workflow for HLA*IMP:02. Standard workflow for HLA*IMP:02: standard output data from popular genotyping platforms, for example current Illumina or Affymetrix chips, are converted into the HLA*IMP format using the locally installed front-end program. The front-end also carries out necessary steps of quality control, such as aligning SNP strandedness. The output files from the front-end are submitted to the HLA*IMP:02 server, which processes the data and produces imputations (posterior probabilities over pairs of alleles as well as a "best guess" pair of two alleles with associated quality scores).

doi:10.1371/journal.pcbi.1002877.g002

results for the first baseline experiment (see next section), in which we compare the performance of HLA*IMP:01 and HLA*IMP:02 on a homogeneous dataset.

At the per-locus level, we use concordance (which is, at the per-locus level, identical to PPV) as a measure of accuracy. We also provide more detailed statistics at the allele level (see below).

Data

The experiments presented in this paper are based on different combinations of two datasets.

The first set, denoted "Golden Set" (GS), has been described elsewhere [1] and comprises 2512 individuals from the 1958 Birth Cohort (http://www.b58cgene.sgul.ac.uk/), the HapMap CEU [20] and the CEPH CEU+ [7] cohorts. Genotyping of the GS was carried out on the Illumina 1.2M and Affymetrix Genome-Wide Human SNP Array 6.0 chips. HLA typing methods vary according to the original cohort. Protocols for 1958 BC HLA genotyping are described online (https://www-gene.cimr.cam.ac.uk/public_data/HLA/HLA.shtml). CEU and CEU+ were typed using exon-sequencing methods.

The second set, denoted "HLARES_ALL", has been provided by GlaxoSmithKline and comprises (post quality control, as described in Dilthey et al. [1]) 1460 individuals from diverse, though mainly European or European-ancestry, populations (see Supporting Table S2). The individuals in HLARES_ALL were drawn from several clinical trials and typed on the Illumina 1 M SNP genotyping platform, and classical HLA type information (derived by exon sequencing) is available for many of them (see Table 1 for details). Genome-wide principal components analysis (PCA) of the samples in HLARES_ALL was carried out using the program EIGENSTRAT [21].

We resolve ambiguous HLA type information by using the maximum population frequency call. Besides that, we treat all HLA genotypes "as is"; that is, we make no attempt to control, for example, for changes of HLA nomenclature or allele databases. This might lead to slight underestimates of accuracy (in the worst case, we do not recognize identical alleles as identical).

In the first experiment (homogeneous reference), we evaluate the performance of statistical HLA type imputation (*HLA-A* , *-B* , *-C* , *-DQA1* , *-DQB1* and *-DRB1*) on cross-European samples, based on a mainly British reference panel. We use the GS as reference panel to impute classical HLA types of those samples in HLARES_ALL with self-declared European ancestry (HLARES_EU) and measure concordance with lab-derived HLA type information where available. Supporting Table S2 describes the distribution of countries the individuals in HLARES_EU were sampled from. There are 6056 SNPs in the extended MHC region (xMHC, here defined as the chromosomal region on chromosome 6 from position 25,921,129 to position 33,535,328, build 36; see Horton et al. [22]) in the intersection of the GS and HLARES_EU datasets. To mirror the context in which HLA*IMP:01 was applied in recent genome-wide association studies [3,4,19], we further restrict the available SNP set to those also present in one of them [3], resulting in 2020 SNPs.

In the second experiment (medium heterogeneity), we evaluate the performance of statistical HLA type imputation on European samples, based on a cross-European reference panel. To obtain a cross-European reference panel (GS&HLARES_EU), we merge the GS and HLARES_EU datasets, keeping only SNPs in the intersection of the two panels (6056 xMHC SNPs). We randomly split GS&HLARES_EU into two panels, and use the first one (GS&HLARES_EU 2/3, containing approximately 2/3 of the original data) as reference, and the second one (GS&HLARES_EU 1/3, approximately 1/3 of the original data) as validation panel. Referring to the increased population structure in GS&HLARES_EU 2/3 as compared to GS, we call GS&HLARES_EU 2/3 a heterogeneous reference panel. We measure concordance with experimentally-derived HLA type information where available. We use the data on additional loci present in GS&HLARES_EU (*HLA-DPB1* , *-DRB3* , *-DRB4* , *-DRB5*) to evaluate how well their allelic states can be imputed. Also, in a variation of the second experiment, we modify the SNP density in the xMHC region to investigate to what extent performance will depend on the selected SNP genotyping platform and data missingness profiles.

In the third experiment (high heterogeneity), we evaluate the performance of statistical HLA type imputation on multi-ethnic samples, based on a multi-ethnic reference panel. To obtain a multi-ethnic reference panel (GS&HLARES_ALL), we merge the GS and HLARES_ALL datasets. We also include the HapMap YRI cohort [20], as individuals self-reporting as of African ancestry constitute a subset of HLARES_ALL. We keep all available SNP genotypes from the intersection of GS and YRI (7733 SNPs from GS of which 7632 xMHC SNPs are also present in YRI), and combine them with the SNP genotypes from HLARES_ALL (6050 SNPs, setting the remaining 1582 SNP genotypes to "missing"). The resulting set GS&HLARES_ALL has 7632 xMHC SNPs. We randomly split GS&HLARES_ALL in two panels, and use the first one (GS&HLARES_ALL 2/3, containing approximately 2/3 of the original data) as reference, and the second one as (GS&HLARES_ALL 1/3, approximately 1/3 of the original data) as validation panel. We call GS&HLARES_ALL 1/3 a "highly heterogeneous" reference panel. Note that GS&HLARES_ALL is still dominated by samples of European origin. We measure concordance with experimentally-derived HLA type information where available.

Table 1 provides a summary of the number of individuals and HLA alleles present in all reference and validation panels.

Results

We have repeated some of the initial HapMap-based experiments from Dilthey et al. [1] to investigate the effects of the methodological innovations proposed in this paper (see Supporting Table S1 and Section "Properties of the presented model and parameter inference" in the Supporting Text S1). We find that allowing for path uncertainty has a positive effect across all examined loci. The additional localization criteria, though theoretically appealing, do not consistently improve accuracy across loci (see Supporting Text S1, Section "Properties of the presented model and parameter inference"). Based on our initial experiments, localization is not used for *HLA-B* and *HLA-DRB1* .

On a homogeneous reference panel (first experiment, GS), HLA*IMP:02 achieves the same level of performance as HLA*IMP (see Table 2). Measured at six classical HLA loci (*HLA-A* , *-B* , *-C* , *-DQA1* , *-DQB1* and *-DRB1*), HLA*IMP:02 achieves an average 4-digit resolution accuracy of 94% at an average call rate of 97%, vs. 93% accuracy at a call rate of 97% for HLA*IMP:01 (call threshold T = 0.7 for HLA*IMP:01 and matched to obtain equal or higher call rates for HLA*IMP:02). Locus-specific performance is very similar for both models. We observe the lowest accuracy at *HLA-DQA1* (88%) and the lowest call rate at *HLA-DRB1* (90%).

On a heterogeneous reference panel (second experiment, GS&HLARES_EU 2/3), HLA*IMP:02 achieves an average accuracy of 97% at an average call rate of 97% (see Table 3). HLA*IMP:01, in contrast, achieves an average accuracy of 93% at an average call rate of 93% (using a call threshold of T = 0.7 for

Table 1. Dataset characteristics summary.

Number of individuals with at least one allele typed at 4-digit resolution (2-digit for DRB3, DRB4, DRB5)

	SNPs	HLA-A	HLA-B	HLA-C	HLA-DQA1	HLA-DQB1	HLA-DRB1	HLA-DPB1	HLA-DRB3	HLA-DRB4	HLA-DRB5
GS	7733	1556	1570	1153	87	1585	1517	0	0	0	0
HLARES_EU	7568	308	1060	349	279	446	897	74	282	282	282
GS&HLARES_EU	6056	1864	2630	1502	366	2031	2414	74	282	282	282
GS&HLARES_EU 2/3	6056	1253	1758	1017	250	1359	1592	50	187	187	187
GS&HLARES_EU 1/3	6056	611	872	485	116	672	822	24	95	95	95
GS&HLARES_ALL	7632	2028	3063	1675	521	2223	2809	112	353	353	353
GS&HLARES_ALL 2/3	7632	1356	2055	1129	354	1495	1853	77	242	242	242
GS&HLARES_ALL 1/3	7632	672	1008	546	167	728	956	35	111	111	111

Number of 4-digit HLA alleles (2-digit for DRB3, DRB4, DRB5)

	HLA-A	HLA-B	HLA-C	HLA-DQA1	HLA-DQB1	HLA-DRB1	HLA-DPB1	HLA-DRB3	HLA-DRB4	HLA-DRB5
GS	27	46	21	7	18	34	0	0	0	0
HLARES_EU	34	60	28	14	17	43	18	5	3	4
GS&HLARES_EU	38	65	30	14	19	47	18	5	3	4
GS&HLARES_EU 2/3	33	60	27	13	19	42	15	5	3	3
GS&HLARES_EU 1/3	28	52	24	11	17	37	13	5	3	4
GS&HLARES_ALL	58	110	39	15	21	62	21	5	3	4
GS&HLARES_ALL 2/3	48	99	34	14	20	56	17	5	3	4
GS&HLARES_ALL 1/3	45	85	30	13	20	49	17	5	3	3

The upper part of this table shows the number of individuals that are available for building the HLA*IMP:02 graphs in a locus-specific manner. For HLA*IMP, the number of available haplotypes is approximately double the individual number. The bottom part of the table shows allelic diversity for all reference and validation datasets used in our study. Note that the allelic diversity in the HLARES and in the GS&HLARES 2/3 datasets is bigger than in the GS.

doi:10.1371/journal.pcbi.1002877.t001

Table 2. Baseline validation on a homogeneous reference panel.

Threshold	Locus	# Validated	HLA*IMP:02			HLA:IMP:01		
			Call Rate	Accuracy	T	Call Rate	Accuracy	T
T = 0.00	*HLA-A*	574	1.00	0.96		1.00	0.90	
	HLA-B	2002	1.00	0.90		1.00	0.93	
	HLA-C	596	1.00	0.96		1.00	0.96	
	HLA-DQA1	446	1.00	0.87		1.00	0.87	
	HLA-DQB1	758	1.00	0.98		1.00	0.97	
	HLA-DRB1	1730	1.00	0.88		1.00	0.89	
T = Matched	*HLA-A*	574	0.96	0.96	0.55	0.94	0.91	0.700
	HLA-B	2002	0.98	0.92	0.40	0.98	0.94	0.700
	HLA-C	596	0.99	0.96	0.60	0.99	0.97	0.700
	HLA-DQA1	446	0.99	0.88	0.40	0.99	0.88	0.700
	HLA-DQB1	758	0.99	0.98	0.60	0.99	0.97	0.700
	HLA-DRB1	1730	0.90	0.93	0.60	0.90	0.93	0.700

Non-thresholded and thresholded HLARES validation results for HLA*IMP:02 and HLA*IMP:01: the complete GS is used to impute HLARES_EU samples. Accuracy (PPV) is measured at 4-digit resolution. "# Validated" refers to the number of validated alleleles (pre-thresholding). Note that the call threshold for HLA*IMP:02 was matched to obtain equal or higher call rates than with HLA*IMP:01.

doi:10.1371/journal.pcbi.1002877.t002

both models). The most problematic locus for HLA*IMP:02 is *HLA-DRB1* , with an achieved accuracy/call rate of 95%/91%. Even without call threshold, HLA*IMP:02 achieves an all-loci average accuracy of 96% (vs. 90% for HLA*IMP:01). At T = 0.00, HLA*IMP:02 outperforms HLA*IMP:01 at every locus, by 6% on average. Applied to *HLA-DPB1* and the allelic state of the *DRB* paralogs (see Supporting Table S3), HLA*IMP:02 achieves an accuracy of 90% on *DPB1* without any call threshold. Due to the limitations of the data set, we can only evaluate the performance at the *DRB* paralogous loci at 2-digit resolution, including one pseudo-allele for absence from a haplotype. We find the imputations to be correct in ≥94% of cases (T = 0.00, very similar results obtained for HLA*IMP:01, data not shown). HLA*IMP:02 produces well-calibrated imputations (see Supporting Figure S1).

Table 3. Multi-population European validation results.

Threshold	Locus	# Validated	HLA*IMP:02		HLA:IMP:01	
			Call Rate	Accuracy	Call Rate	Accuracy
T = 0.00	*HLA-A*	808	1.00	0.97	1.00	0.91
	HLA-B	1646	1.00	0.95	1.00	0.89
	HLA-C	752	1.00	0.96	1.00	0.91
	HLA-DQA1	194	1.00	0.97	1.00	0.87
	HLA-DQB1	934	1.00	0.98	1.00	0.92
	HLA-DRB1	1358	1.00	0.91	1.00	0.87
T = 0.70	*HLA-A*	808	0.98	0.97	0.94	0.94
	HLA-B	1646	0.96	0.97	0.93	0.92
	HLA-C	752	0.99	0.97	0.94	0.94
	HLA-DQA1	194	0.96	0.98	0.93	0.90
	HLA-DQB1	934	0.99	0.98	0.94	0.94
	HLA-DRB1	1358	0.91	0.95	0.89	0.92

Medium heterogeneity non-thresholded and thresholded cross-validation results for HLA*IMP:02 and HLA*IMP:01: GS&HLARES_EU 2/3 is used to impute GS&HLARES_EU 1/3. Accuracy (PPV) is measured at 4-digit resolution. "# Validated" refers to the number of validated alleleles (pre-thresholding).

doi:10.1371/journal.pcbi.1002877.t003

By analyzing allele- and locus-specific error profiles, we can identify factors influencing the imputation accuracy of HLA*IMP:02 (see Figure 3). First, we note that most alleles are imputed reliably at 4-digit resolution, in particular those with higher frequencies in the reference panel. Alleles that exhibit problems at 4-digit imputation are typically correctly imputed at 2-digit resolution. Second, we can distinguish between at least three classes of problems. Some alleles, for example HLA-A*33:01, are not present in the reference dataset at all. They can therefore not be correctly imputed. Other alleles, for example HLA-B*27:02, are present in the reference dataset, but at low frequencies. Non-calls and 4-digit errors accumulate for these alleles. Third, some alleles, for example DRB1*01:01, are better represented in the reference panel, but there are still some problems with imputing them correctly. We note that these error modes are also seen in HLA*IMP:01 and that the identified classes of error also apply to the homogeneous reference experiment (see Supporting Figures S2, S3, S4). Finally, there is another abundant type of error, seen only in HLA*IMP:01 and not observed in the low heterogeneity case, which drives the observed drop in performance difference relative to HLA*IMP:02: classification problems for well-represented alleles. It seems likely that this is due to within-Europe population structure and heterogeneity in haplotype backgrounds, which the model of HLA*IMP:01 cannot take into account appropriately. We provide allele-specific measures of sensitivity, specificity, PPV and r^2, based on HLA*IMP:02, for the first two experiments in Supporting Tables S4 and S5.

To investigate how strong an effect the utilized SNP genotyping array and missing data in the imputation dataset will have on expected accuracy, we carry out a variation of the second experiment. Instead of separately evaluating a range of genotyping platforms and missingness profiles, we present two generic experiments, focusing on SNP density in the xMHC region: we

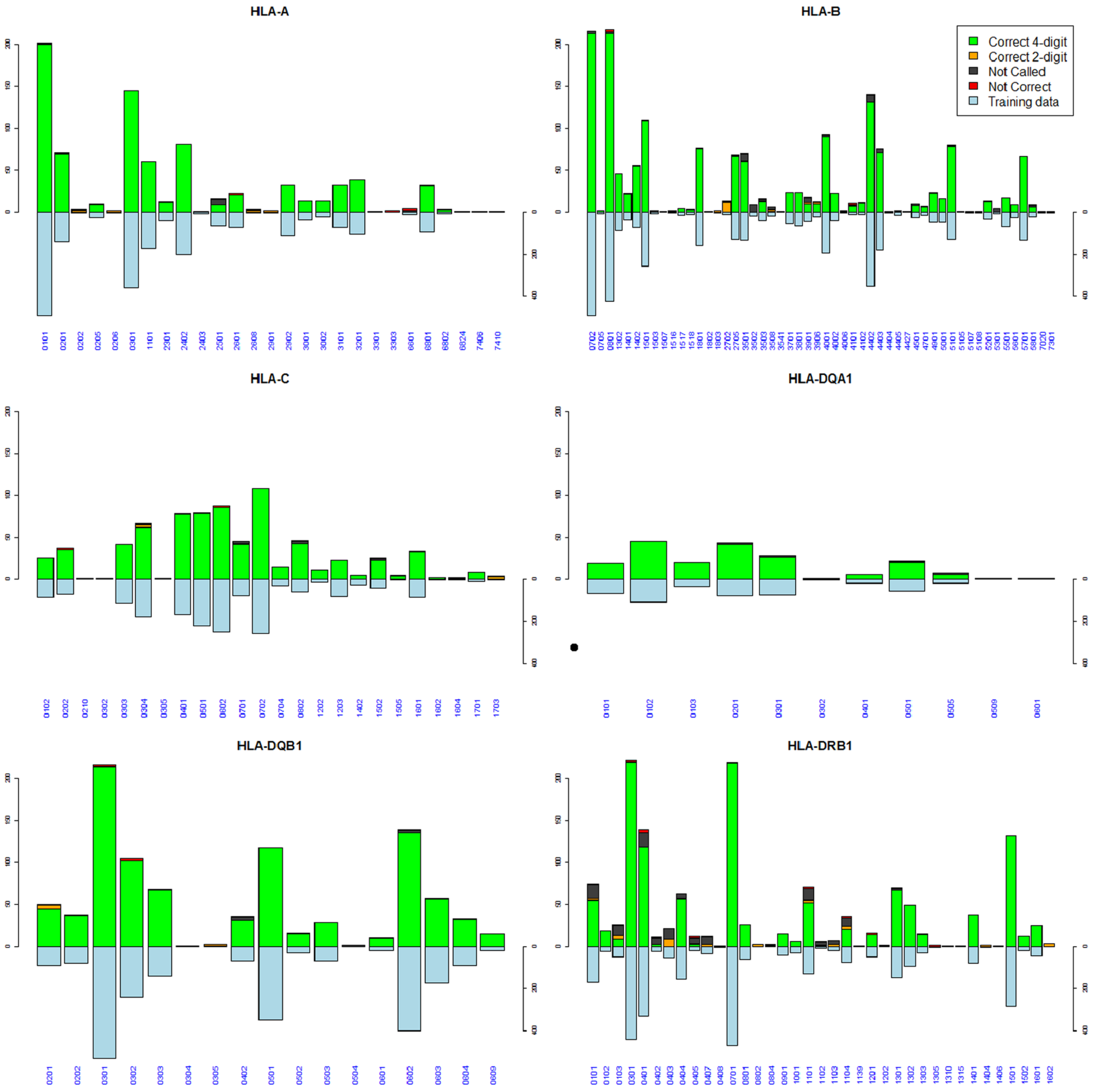


Figure 3. Per-allele accuracies on a diverse European reference panel (HLA*IMP:02). Per-allele analysis of HLA*IMP:02 imputation accuracy for six classical loci in the GS&HLARES_EU cross-European validation experiment at a call threshold of T = 0.70. The x-axis represents the different HLA alleles in the validation panel. The downward blue bars indicate how often each allele appears in the reference panel (the GS&HLARES_EU 2/3 dataset). Imputation success is indicated by the upward stack plots: green indicates correct imputations at 4-digit HLA type resolution; orange indicates correct imputations at 2-digit resolution; black indicates alleles below the call threshold; red indicates incorrect imputations. Non-calls and imputations which are only correct at 2-digit resolution accumulate in the alleles which are rare or not present at all in the reference panel.
doi:10.1371/journal.pcbi.1002877.g003

randomly delete 70% and 90% of the SNP genotypes from the inference panel (independently for each individual, to minimize sampling effects), while the graph from the second experiment remains unchanged. In the 70% scenario, each individual remains with >1500 SNPs in the xMHC region, which is comparable to the SNP density of many 500 K arrays. In the 90% scenario, approximately 600 SNPs in the xMHC region remain, which is substantially less than the number provided by older ~300 K genotyping arrays. We observe that even with low SNP densities, the observed performance of HLA*IMP:02 is relatively stable: Setting 70% of the SNP genotypes in the inference panel (GS&HLARES_EU 1/3) to "missing", the drop in achieved per-locus accuracy is ≥1% (at a call threshold of T = 0.00, see Table 4). Setting 90% of the SNP genotypes in the inference panel to "missing", the maximum loss in accuracy is 5% for all loci but *DQA1* (probably related to the smaller amount of reference data

Table 4. Missing data in the inference panel.

Locus	# Validated	70% missing	90% missing
HLA-A	808	0.96	0.94
HLA-B	1646	0.95	0.93
HLA-C	752	0.95	0.94
HLA-DQA1	194	0.96	0.90
HLA-DQB1	934	0.97	0.95
HLA-DRB1	1358	0.90	0.86

4-digit resolution accuracies (PPV) when 70% and 90% of the inference panel SNP genotypes (GS&HLARES_EU 1/3) in the second experiment are randomly set to "missing". No call threshold is employed. "# Validated" refers to the number of validated alleleles.

doi:10.1371/journal.pcbi.1002877.t004

for this locus, see Table 1), where it is 7%. Of note, many reference panel SNPs are present on the Immunochip platform; repeating the second experiment constrained to the Immunochip SNP set for the imputation panel shows virtually the same results as the 70% experiment (data not shown).

Increasing the heterogeneity in the reference panel (third experiment, GS&HLARES_ALL 2/3) by including individuals of other ethnicities (African-ancestry, Asian, Hispanic, third experiment) only slightly decreases the achieved performance on the European validation samples for HLA*IMP:02 (see Table 5), yielding an average accuracy of 97% and an average call rate of 95% (T = 0.70). At 4-digit resolution, performance on the non-European samples is markedly lower, with an average accuracy and an average call rate of 87% (T = 0.70). Imputation accuracy is lowest for the Asian samples (average accuracy 76% at T = 0.00) and comparable for African-ancestry and Hispanic samples (84% and 85% respectively, at T = 0.00). There are more pronounced locus-specific differences in the non-European validation data: In the African-ancestry samples and at T = 0.00, for example, accuracy at *HLA-DRB1* is at 71%, whereas it is at 97% at *HLA-C* . At 2-digit resolution, alleles are imputed more reliably: average accuracy at T = 0.00 is 90% for Asian samples (ranging from 78% at *HLA-B* to 98% at *HLA-DRB1*); 93% for samples of African ancestry (ranging from 82% at *HLA-B* to 100% at *HLA-C/DQA1/DQB1*); and 99% for Hispanic samples (ranging from 97% at *HLA-B* to 100% at all other loci);

Discussion

Better imputation of classical HLA alleles is an important goal in enabling association studies to understand the genetic risk of many complex and infectious diseases. We have developed HLA*IMP:02, a statistical model for the imputation of classical HLA types, which attempts to address problems arising in performing imputation from multiple heterogeneous (both in experimental origin and ethnicity) data sets. We have shown that HLA*IMP:01 (our previous method; [1,2]) and HLA*IMP:02 achieve similar levels of performance on homogeneous reference panels, but that HLA*IMP:02 clearly outperforms HLA*IMP:01 on heterogeneous European reference panels, yielding accuracies and call rates ≥95% at 4-digit resolution in nearly all European scenarios. Using HLA*IMP:02 instead of HLA*IMP:01 can therefore be expected to increase power and accuracy in cross-European genome-wide association studies.

The improved performance of HLA*IMP:02 (when compared with HLA*IMP:01) is likely due to the path-based approach that allows for HLA alleles to appear on multiple haplotype backgrounds, a known consequence of population stratification in the HLA region. To further investigate this hypothesis, we have examined the local haplotype structure around the HLA-A*02:01 allele in GS&HLARES_EU, as inferred (and used) by HLA*IMP:01 (Supporting Figure S5, part B). From visual inspection of the figure, it is clear that there are at least three major haplotypic backgrounds for 02:01 (when inspecting the corresponding figure for the GS, we find two major haplotypic backgrounds; Supporting Figure S5, part A). What is more, when comparing the haplotypes that HLA*IMP:01 correctly imputes with those that it doesn't, we find that there are features which appear virtually exclusively in the second group (marked in S11 part B). Interestingly, these features are also present in the group of haplotypes that serve as reference panel, but the model does not seem to utilize this information in the right way. This is consistent with our interpretation that the model of HLA*IMP:01 does not cope well with haplotypic heterogeneity. HLA*IMP:02, on the other hand, can accommodate haplotypic heterogeneity and imputes A*02:01 nearly perfectly in the same experiment.

The observed performance of HLA*IMP:02 is relatively stable under high levels of missing data in the inference panel. This property represents an important improvement upon HLA*IMP:01, which offered no conceptually consistent way (except for repeating the computationally intensive process of SNP selection) towards dealing with missing SNPs in the inference panel. Of note, the HLA*IMP:02 back end web service will automatically carry out the SNP density experiment presented here, constraining the set of available SNPs to those found in the user dataset. The results from this experiment (including average per-locus accuracies and PPV, sensitivity and specificity for each allele) are included in the archive file which contains the main imputations.

The model of HLA*IMP:02 could handle pre-phased data in a straightforward way. There is no evidence to suggest that recent encouraging results from SNP genotype imputation [23] do not apply to pre-phasing with the aim of HLA type imputation. However, in light of the complex regional haplotype structure and high levels of diversity, we believe that the effect of pre-phasing on HLA type imputation accuracy needs to be studied in more detail.

At 2-digit resolution, HLA*IMP:02 achieves average accuracies ≥90% for all tested ethnicities using a multi-ethnic reference panel. These results suggest that the model's ability to deal with heterogeneity in the reference set extends to highly diverse panels. Moreover, extensions of the reference panel in a way that matches imputation study panels can be expected to furthermore increase (4-digit) performance, in particular for samples that are not well-represented by the current reference. We illustrate this effect in Figure 4 for *HLA-DRB1* , one of the more challenging loci for HLA type imputation. The figure displays samples from HLARES_ALL 1/3 stratified by the samples' first two principal components (it is well-known that PCA can be used to control for population stratification [24] and is informative of relatedness [25]). In one experiment, we use an exclusively European reference to impute the samples (left-hand panel). In the other experiment, we make use of the full reference panel GS&HLARES ALL 2/3 (right-hand panel). Particularly samples in the periphery of PC space benefit from improving reference panel size and match with the imputation panel, whereas samples in the proximity of European data are hardly affected. Averaged over all loci, accuracy for the non-European samples increases by 8% when including the non-European reference data (data not shown). These observations are consistent with results from SNP genotype imputation, where using matched and diverse reference panels is also known to have a positive effect on accuracy [11,12,26,27].

Table 5. Multi-ethnic validation results.

Threshold	Population	Locus	# Validated	Call Rate	Accuracy 4-digit	Accuracy 2-digit
T = 0.00	African-American/African	*HLA-A*	30	1.00	0.73	0.83
		HLA-B	44	1.00	0.73	0.82
		HLA-C	30	1.00	0.97	1.00
		HLA-DQA1	28	1.00	1.00	1.00
		HLA-DQB1	30	1.00	0.87	1.00
		HLA-DRB1	34	1.00	0.71	0.91
	Asian	*HLA-A*	28	1.00	0.79	0.96
		HLA-B	110	1.00	0.68	0.78
		HLA-C	28	1.00	0.82	0.89
		HLA-DQA1	22	1.00	0.73	0.91
		HLA-DQB1	36	1.00	0.83	0.89
		HLA-DRB1	102	1.00	0.72	0.98
	European	*HLA-A*	824	1.00	0.96	0.97
		HLA-B	1662	1.00	0.95	0.98
		HLA-C	752	1.00	0.97	0.99
		HLA-DQA1	206	1.00	0.96	0.99
		HLA-DQB1	924	1.00	0.97	0.99
		HLA-DRB1	1356	1.00	0.90	0.99
	Hispanic	*HLA-A*	28	1.00	0.82	1.00
		HLA-B	126	1.00	0.63	0.97
		HLA-C	36	1.00	0.92	1.00
		HLA-DQA1	28	1.00	0.93	1.00
		HLA-DQB1	40	1.00	0.97	1.00
		HLA-DRB1	128	1.00	0.80	0.98
T = 0.70	African-American/African	*HLA-A*	30	0.93	0.79	0.89
		HLA-B	44	0.89	0.79	0.85
		HLA-C	30	1.00	0.97	1.00
		HLA-DQA1	28	1.00	1.00	1.00
		HLA-DQB1	30	0.93	0.89	1.00
		HLA-DRB1	34	0.59	1.00	1.00
	Asian	*HLA-A*	28	0.96	0.81	1.00
		HLA-B	110	0.71	0.85	0.91
		HLA-C	28	0.86	0.79	0.88
		HLA-DQA1	22	0.82	0.78	0.94
		HLA-DQB1	36	0.83	0.90	0.93
		HLA-DRB1	102	0.74	0.83	1.00
	European	*HLA-A*	824	0.95	0.97	0.98
		HLA-B	1662	0.95	0.97	0.99
		HLA-C	752	0.99	0.97	0.99
		HLA-DQA1	206	0.97	0.97	0.99
		HLA-DQB1	924	0.99	0.98	0.99
		HLA-DRB1	1356	0.87	0.95	0.99
	Hispanic	*HLA-A*	28	1.00	0.82	1.00
		HLA-B	126	0.75	0.73	0.97
		HLA-C	36	0.94	0.97	1.00
		HLA-DQA1	28	0.86	0.96	1.00
		HLA-DQB1	40	0.95	1.00	1.00
		HLA-DRB1	128	0.73	0.88	1.00

High heterogeneity non-thresholded and thresholded cross-validation results for HLA*IMP:02, stratified by ethnicity of the imputed samples. GS&HLARES_ALL 2/3 is used to impute GS&HLARES_ALL 1/3. Accuracy (PPV) is measured at 4-digit resolution and at 2-digit resolution. "# Validated" refers to the number of validated alleleles (pre-thresholding).

doi:10.1371/journal.pcbi.1002877.t005

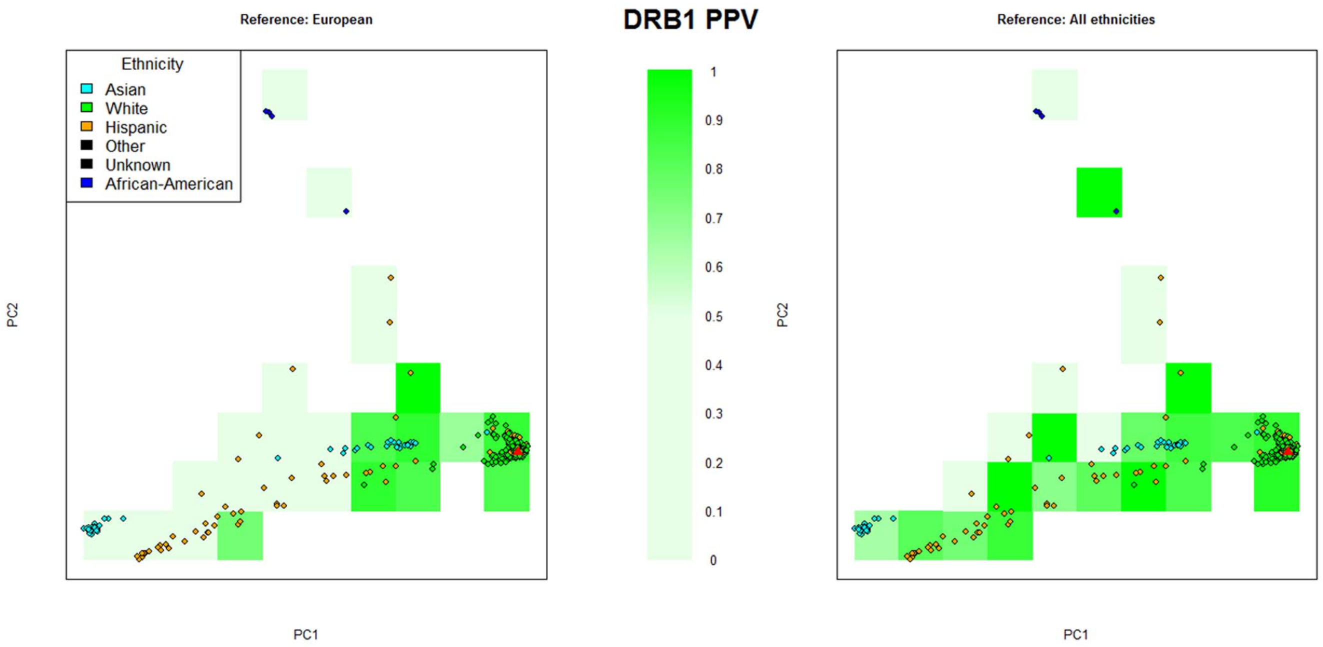


Figure 4. Accuracy comparison between complete and European-restricted reference panels. PCA-stratified accuracy comparison (*HLA-DRB1*) between the complete reference panel (GS&HLARES_ALL, right plot) and a European-restricted reference panel (left side) for the high heterogeneity scenario (imputing GS&HLARES_ALL 1/3, only samples from HLARES displayed). In each quadrant, mean accuracy (PPV) is indicated by color. The red triangle indicates the (approximate) centre of the European reference data. Note that incorporating the non-European reference data increases accuracy in particular for the non-European samples.
doi:10.1371/journal.pcbi.1002877.g004

In summary, the model of HLA*IMP:02 contributes to solving the important challenge of making HLA type inference from combined multi-population reference panels. Raising the accuracy of 4-digit imputation accuracy for non-European populations to the level currently observed for European samples is an important future goal that will require collection of reference data from other populations. However, the framework developed here should enable such integration to happen without compromising accuracy in European-ancestry populations.

Supporting Information

Figure S1 Calibration HLA*IMP:02. Calibration plot HLA*IMP:02, second experiment, medium heterogeneity. The red points show expected (x-axis) and achieved mean accuracies (y-axis) in each bin of step size 0.1, and the blue line is a plot of x = y. Note that the first four data points (bins 0–3) are only based on 37 individuals.
(TIF)

Figure S2 Per-allele analysis for HLA*IMP:02/ HLARES. Per-allele analysis of HLA*IMP:02 imputation accuracy for six classical loci in the HLARES validation experiment (first experiment, homogeneous reference) at a call threshold of T = 0.70. The x-axis represents the different HLA alleles in the validation panel. The downward blue bars indicate how often each allele appears in the reference panel (the GS dataset). Imputation success is indicated by the upward stack plots: green indicates correct imputations at 4-digit HLA type resolution; orange indicates correct imputations at 2-digit resolution; black indicates alleles below the call threshold; red indicates incorrect imputations.
(TIF)

Figure S3 Per-allele analysis for HLA*IMP:01/ HLARES. Per-allele analysis of HLA*IMP:01 imputation accuracy for six classical loci in the HLARES validation experiment (first experiment, homogeneous reference) at a call threshold of T = 0.70. The x-axis represents the different HLA alleles in the validation panel. The downward blue bars indicate how often each allele appears in the reference panel (the GS dataset). Imputation success is indicated by the upward stack plots: green indicates correct imputations at 4-digit HLA type resolution; orange indicates correct imputations at 2-digit resolution; black indicates alleles below the call threshold; red indicates incorrect imputations.
(TIF)

Figure S4 Per-allele analysis for HLA*IMP:01/ GS&HLARES_EU. Per-allele analysis of HLA*IMP:01 imputation accuracy for six classical loci in the GS&HLARES_EU validation experiment (second experiment, medium heterogeneity reference) at a call threshold of T = 0.70. The x-axis represents the different HLA alleles in the validation panel. The downward blue bars indicate how often each allele appears in the reference panel (the GS dataset). Imputation success is indicated by the upward stack plots: green indicates correct imputations at 4-digit HLA type resolution; orange indicates correct imputations at 2-digit resolution; black indicates alleles below the call threshold; red indicates incorrect imputations.
(TIF)

Figure S5 Barcode plot for HLA-A*02:01 in in GS and GS&HLARES_EU. This plot shows the inferred haplotype structure ("barcode plot") for HLA-A*02:01 in the first (based on GS, part A) and second experiment (based on GS&HLARES_EU, part B). Each row represents one haplotype, and each SNP is depicted as a little square. The colouring of the boxes indicates whether the haplotype carries the major SNP allele (bright box) or a minor allele (dark box). The black/white rows

represent the haplotypes carrying the 02:01 allele in the reference panel. Red and green rows represent haplotypes carrying 02:01 in the validation panel, with green indicating successful imputation and red indicating misimputation. We only show SNPs selected by HLA*IMP:01 in the process of SNP selection, and the inferred haplotypes are taken from the phased reference panel for HLA*IMP:01. The inferred haplotype structure in the second experiment in more complex than in the first experiment. Comparing correctly and incorrectly imputed haplotypes in the second experiment, it is clear that there are features (highlighted) which appear virtually exclusively in incorrectly imputed haplotypes (although they are present in the reference panel). Note that A*02:01 is imputed virtually perfectly by HLA*IMP:02 in this experiment, consistent with our hypothesis that HLA*IMP:02 is more tolerant of heterogeneous haplotype structures.
(TIF)

Table S1 HapMap-based BC58 validation accuracy. Accuracies (PPV) for the HapMap-based BC58 validation, as described in Leslie et al. [2] and Dilthey et al. [1]. No call threshold is employed. The column “HLA*IMP:02” refers to the full model with error parameters $!=0$ and localization (other parameters set to accommodate the much reduced sample size). In column I, the error probabilities for sampling from the graph (m_S) and for building the graph m_B are set to 0 (all other parameters equal to the column “HLA*IMP:02”). In column II, the error probability for building the graph is set to 0, and in column III, the error probability for sampling from the graph is set to 0. In column IV, localization is deactivated.
(DOCX)

Table S2 Countries and ethnicities in HLARES. Country and ethnicity of samples in the HLARES_EU and HLARES_ALL datasets.
(DOCX)

Table S3 HLA-DPB1 and DRB3-5. HLARES_EU cross validation for additional loci and structural variation (second experiment, medium heterogeneity): 2/3 of the HLARES_EU dataset are used as reference to impute the remaining 1/3. No call threshold is employed. Accuracy (PPV) for HLA-DPB1 measured at 4-digit resolution, at 2-digit resolution (including one pseudo-allele for absence) for DRB orthologs.
(DOCX)

Table S4 HLA-DPB1 and DRB3-5. Allele-specific sensitivity, specificity, PPV and r^2 for the first experiment (HLA*IMP:02, GS $->$ HLARES_EU). “NValidation” specifies how often an allele appears in the validation data (according to classical typing results, which we treat as the truth in this experiment). “NImputation” specifies how often an allele appears in the imputations for the validation data. The following columns specify sensitivity, specificity, PPV and r2 for each allele. All numbers are based on “best-guess” called alleles.
(DOCX)

Table S5 HLA-DPB1 and DRB3-5. Allele-specific sensitivity, specificity, PPV and r^2 for the second experiment (HLA*IMP:02, GS&HLARES_EU 2/3 $->$ GS&HLARES_EU 1/3). “NValidation” specifies how often an allele appears in the validation data (according to classical typing results, which we treat as the truth in this experiment). “NImputation” specifies how often an allele appears in the imputations for the validation data. The following columns specify sensitivity, specificity, PPV and r2 for each allele. All numbers are based on “best-guess” called alleles.
(DOCX)

Text S1 The HLA*IMP:02 model and algorithms. Mathematical and algorithmic characterization of the haplotype graph model of HLA*IMP:02, allowing for integrating over path uncertainty and localization.
(PDF)

Acknowledgments

We acknowledge the Oxford Supercomputing Centre (OSC). We also acknowledge use of genotype data from the British 1958 Birth Cohort DNA collection, funded by the Medical Research Council grant G0000934 and the Wellcome Trust grant 068545/Z/02. We also thank three anonymous reviewers for their comments.

Author Contributions

Conceived and designed the experiments: AD SL MRN GM. Performed the experiments: AD SL LM JS. Analyzed the data: AD SL. Contributed reagents/materials/analysis tools: CC. Wrote the paper: AD SL MRN GM.

References

1. Dilthey AT, Moutsianas L, Leslie S, McVean G (2011) HLA*IMP – an integrated framework for imputing classical HLA alleles from SNP genotypes. Bioinformatics 27: 968–72.
2. Leslie S, Donnelly P, McVean G (2008) A statistical method for predicting classical HLA alleles from SNP data. Am J Hum Genet 82: 48–56.
3. Sawcer S, Hellenthal G, Pirinen M, Spencer CC, Patsopoulos NA, et al. (2011) Genetic risk and a primary role for cell-mediated immune mechanisms in multiple sclerosis. Nature 476: 214–9.
4. Strange A, Capon F, Spencer CC, Knight J, Weale ME, et al. (2010) A genomewide association study identifies new psoriasis susceptibility loci and an interaction between HLA-C and ERAP1. Nat Genet 43: 761–7.
5. Hosking FJ, Leslie S, Dilthey A, Moutsianas L,Wang Y, et al. (2011) MHC variation and risk of childhood B-cell precursor acute lymphoblastic leukemia. Blood 117: 1633–40.
6. Raychaudhuri S, Sandor C, Stahl EA, Freudenberg J, Lee HS, et al. (2012) Five amino acids in three HLA proteins explain most of the association between MHC and seropositive rheumatoid arthritis. Nat Genet 44: 291–6.
7. de Bakker PI, McVean G, Sabeti PC, Miretti MM, Green T, et al. (2006) A highresolution HLA and SNP haplotype map for disease association studies in the extended human MHC. Nat Genet 38: 1166–72.
8. Li N, Stephens M (2003) Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. Genetics 165: 2213–33.
9. International H I V Controllers Study, Pereyra F, Jia X, McLaren PJ, Telenti A, et al. (2010) The major genetic determinants of HIV-1 control affect HLA class I peptide presentation. Science 330: 1551–7.
10. Huang L, Li Y, Singleton AB, Hardy JA, Abecasis G, et al. (2009) Genotypeimputation accuracy across worldwide human populations. Am J Hum Genet 84: 235–50.
11. Jostins L, Morley KI, Barrett JC (2011) Imputation of low-frequency variants using the HapMap3 benefits from large, diverse reference sets. Eur J Hum Genet 19: 662–6.
12. Howie B, Marchini J, Stephens M (2011) Genotype imputation with thousands of genomes. G3 (Bethesda) 1: 457–70.
13. Browning SR (2006) Multilocus association mapping using variable-length Markov chains. Am J Hum Genet 78: 903–13.
14. Browning SR, Browning BL (2007) Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. Am J Hum Genet 81: 1084–97.
15. Ron D, Singer Y, Tishby N (1998) On the learnability and usage of acyclic probabilistic finite automata. Journal of Computer and System Sciences 56: 133–152.
16. Browning BL, Yu Z (2009) Simultaneous genotype calling and haplotype phasing improves genotype accuracy and reduces false-positive associations for genome-wide association studies. Am J Hum Genet 85: 847–61.
17. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, et al. (2007) PLINK: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet 81: 559–75.
18. Marchini J, Howie B, Myers S, McVean G, Donnelly P (2007) A new multipoint method for genome-wide association studies by imputation of genotypes. Nat Genet 39: 906–13.
19. Evans DM, Spencer CC, Pointon JJ, Su Z, Harvey D, et al. (2011) Interaction between ERAP1 and HLA-B27 in ankylosing spondylitis implicates peptide

handling in the mechanism for HLA-B27 in disease susceptibility. Nat Genet 43: 761–767.

20. International HapMap Consortium, Frazer KA, Ballinger DG, Cox DR, Hinds DA, et al. (2007) A second generation human haplotype map of over 3.1 million SNPs. Nature 449: 851–61.
21. Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA, et al. (2006) Principal components analysis corrects for stratification in genome-wide association studies. Nat Genet 38: 904–9.
22. Horton R, Gibson R, Coggill P, Miretti M, Allcock RJ, et al. (2008) Variation analysis and gene annotation of eight MHC haplotypes: the MHC Haplotype Project. Immunogenetics 60: 1–18.
23. Howie B, Fuchsberger C, Stephens M, Marchini J, Abecasis GR (2012) Fast and accurate genotype imputation in genome-wide association studies through prephasing. Nat Genet 44: 955–9.
24. Novembre J, Johnson T, Bryc K, Kutalik Z, Boyko AR, et al. (2008) Genes mirror geography within Europe. Nature 456: 98–101.
25. McVean G (2009) A genealogical interpretation of principal components analysis. PLoS Genet 5: e1000686.
26. Marchini J, Howie B (2010) Genotype imputation for genome-wide association studies. Nat Rev Genet 11: 499–511.
27. Li Y, Willer C, Sanna S, Abecasis G (2009) Genotype imputation. Annu Rev Genomics Hum Genet 10: 387–406.

Multi-population classical HLA type imputation
Supporting Text S1

Alexander Dilthey * [1,2,†], Stephen Leslie[3,†], Loukas Moutsianas[2], Judong Shen[4], Charles Cox[5], Matthew R. Nelson[4], Gil McVean[1,2]

[1]Department of Statistics, University of Oxford, 1 South Parks Road, Oxford, OX1 3TG, United Kingdom
[2]Wellcome Trust Centre for Human Genetics, Roosevelt Drive, Oxford OX3 7BN, United Kingdom
[3]Murdoch Childrens Research Institute, Royal Children's Hospital, Flemington Road, Parkville, Victoria 3052 Australia
[4]Quantitative Sciences, GlaxoSmithKline, Research Triangle Park, NC 27709, USA
[5]Quantitative Sciences, GlaxoSmithKline, Stevenage, UK
[†]contributed equally
* dilthey@well.ox.ac.uk

Before describing the algorithmic details of HLA*IMP:02, we give a high-level overview. In genotype imputation, a reference panel with high marker density and a statistical model of population haplotype structure are used to impute missing markers in imputation panels with lower marker density [1]. The situation we deal with in this paper can be described as follows: suppose there is a reference panel R, consisting of the genotype data G_R of N_R individuals, typed at a (non-empty) set of loci L on the same chromosome. Also, there is an imputation panel I, consisting of the genotype data G_I of N_I individuals, typed at a (non-empty) set of loci L' ($L' \subseteq L$). The algorithm we present here then

1. uses R to construct a model of the haplotype structure of the haplotypes present in R. We denote this model as M, and it will, like many other models in population genetics, assign a likelihood to every possible haplotype over the L loci. M belongs to the class of haplotype graph models (formally described below).

2. treats each individual in I independently, and makes inference on the loci of interest $\{L \setminus L'\}$ by integrating over possible underlying haplotypes. That is, for the i-th individual in I, we evaluate the conditional probability $\mathrm{P}((h_1, h_2) \,|\, G_{I,i}, M)$, where M is our model of haplotype structure, $G_{I,i}$ the genotype data of the i-th individual I_i in I and (h_1, h_2) a pair of haplotypes for I_i at the specified L loci. Each (h_1, h_2) implies a genotype at all loci we may want to consider, and we average over these according to $\mathrm{P}((h_1, h_2) \,|\, G_{I,i}, M)$. Note that no information from I is used to build M.

In the following, we will therefore describe two separate tasks: how to infer a haplotype graph model M from a set of genotyped individuals and how to use this model to infer the missing genotypes of an individual with genotype $G_{I,i}$. Note that, although we use a haplotype structure model, haplotype inference (phasing) is not our aim in this paper: we are only interested in the correctness of the resulting genotype imputations. Note also that we will only make explicit reference to HLA types during later stages – for now, we assume that a locus may have an arbitrary number of alleles, and do therefore not fundamentally differentiate between a SNP and a classical HLA locus.

Note that Figures 1, 3 and 2 in this document provide a compressed view of the algorithm's most important features.

Haplotype graph models and their induced HMMs

We give formal definitions of haplotypes, haplotype graph models and how they relate to HMMs.

We define haplotypes to be strings of symbols of length $T = |L|$ (therefore $T > 0$), where at any given position $p \in \{0 .., (T-1)\}$ the symbols come from some predefined set A_p, the *model alphabet.* In the context of genetics the model alphabet may be, for example, the set of possible nucleotides (A, C, G and T).

Let the directed connected graph M consist of the directed edges E and the vertices V, i.e. $M = (V, E)$. If $v_a, v_b \in V$ then we define (v_a, v_b) to be the edge in E running from v_a to v_b, provided such an edge exists, and we say $(v_a, v_b) \in E$. For haplotype graphs, each vertex $v \in V$ has an associated well-defined level function $l(v)$, according to the following definition. There is exactly one vertex v_0 with no incoming edges and $l(v_0) = 0$. This vertex is called the root vertex. For every $(v_a, v_b) \in E$, we define $l(v_b) = l(v_a) + 1$. All vertices with no outgoing edges are called "final vertices", and in the case of haplotype graphs all such vertices have the same level, T. Each level can be thought of as a genetic locus. At each level p there is a set of possible emission symbols, the "model alphabet" A_p. Each edge (v_a, v_b) at level p (i.e. $l(v_a) = p$) has an associated emission symbol from A_p. There are no two edges with the same emission symbol originating from the same vertex. Each edge has an associated transition probability, and the transition probabilities of all edges emanating from a non-final vertex add up to 1.

Haplotype graphs probabilistically generate strings of the same length (haplotypes). We give a description of the algorithm that produces haplotypes from a haplotype graph model. Begin at vertex v_0. If at vertex v_a, select one outgoing edge $e = (v_a, v_b)$ according to the probabilities attached to the edges emanating from v_a. Emit the symbol attached to e. Move to the "target vertex" v_b. Continue this procedure at v_b until v_b is a final vertex with no outgoing edges. This may be thought of as defining a *path* through the graph, i.e. a connected sequence of vertices from v_0 to a final vertex at level T. Note that all so-generated haplotypes have the same length.

Browning and Browning [2] note that haplotype graph models as described above are Hidden Markov Models (HMMs). Each edge corresponds to a state, and the transition probabilities between any two edges (v_a, v_b) and (v_c, v_d) are 0 unless $v_c = v_b$. If $v_c = v_b$ then the transition probabilities are defined by the edge probability distribution at v_b. In a basic model, for a given state the emitted symbol (e.g. nucleotide) is just the symbol associated with the corresponding edge.

The induced HMM so-described provides a haploid model for genetic data. The generalization for diploid data, based on two connected haploid HMMs and their combined emission probabilities follows immediately, as described by Browning and Browning [2]. Informally, if the haploid model has n states mapping to level p in the haplotype graph, the diploid model has n^2 states at the same level, mapped to the set of ordered pairs (k, r) $(k \in \{1..n\},\ r \in \{1..n\})$. To be explicit, state (k, r) refers to the first of the two connected haploid models being in state k, and the other one being in state r. In a basic emission probability distribution, (k, r) emits the unordered 2-tuple (genotype) of the emission symbols associated with the haploid states k and r, and more complex emission probability distributions follow from combining the respective haploid emission probability distributions. The transition probability of (k, r) at level p to (k', r') at level $p+1$ is equal to the product of the haploid transition probabilities from k to k' and r to r'.

Note that at this point we can sample diploid (i.e., sequences g of T ordered genotypes) and haploid (i.e., haplotypes h) data from the graph-equivalent HMMs: $\mathrm{P}(h \,|\, M \to \mathrm{HMM}_1)$ and $\mathrm{P}(g \,|\, M \to \mathrm{HMM}_2)$ are both well-defined ($M \to \mathrm{HMM}_1$ denotes the M-equivalent haploid HMM, and $M \to \mathrm{HMM}_2$ denotes the M-equivalent diploid HMM). If we have observed haploid data h, we can use standard statistical techniques [3] to sample a haplotype h' (formed from concatenating the emission symbols of the edges associated with the traversed states) from $\mathrm{P}(h' \,|\, h, M \to \mathrm{HMM}_1)$. Both h and h' are haplotypes of length T, so that this expression does not seem to be very useful. Now, suppose that h was actually only typed at the loci specified by L', and that all other loci carry a “missing data” symbol E_M. If we now modify the emission probability structure of $M \to \mathrm{HMM}_1$ accordingly, for example by assigning each state the same probability to emit E_M (we say that this is an “agnostic” way to deal with missing data), we can use samples from $\mathrm{P}(h' \,|\, h, M \to \mathrm{HMM}_1)$ to estimate symbols for the positions carrying an E_M in h. This property immediately translates to $M \to \mathrm{HMM}_2$, the M-equivalent diploid HMM. That is, conditional on some genotype data g for an individual, with some of the positions potentially being missing data, we can sample from $\mathrm{P}((h_1, h_2) \mid g, M \rightarrow \mathrm{HMM}_2)$ ((h_1, h_2) is a pair of two haplotypes, formed from concatenating the emission symbols of the pairs of edges associated with the traversed states). If desired, by marginalizing over the respective elements in (h_1, h_2), it is now possible to independently estimate the underlying genotype of any of the loci in L loci conditional on M and g, including of course the loci $L \setminus L'$.

We have now described a (well-known, see [2]) solution to the second task: inference of missing genotypes for an additional individual, conditional on some observed genotype data and a haplotype graph M. By specifying the emission probability structure of $M \rightarrow \mathrm{HMM}_1$ in a way that allows for emitting other symbols than E_M or the state's underlying emission symbol, we introduce a mutation- or error-like effect (like [4]). For all following applications, we define a *graph sampling error* parameter m_S for the emission probability structure of haplotype graph-induced HMMs: conditional on not emitting E_M, the state's underlying emission symbol (coming from the associated edge in M) is emitted with probability $1 - m_S$. With probability m_S, one of the other members of the locus-specific model alphabet is uniformly selected and emitted.

Constructing a haplotype graph

We now describe our algorithm to construct a haplotype graph model M from a set R of individuals genotyped at T loci.

Note that the following description is conceptual – when actually implementing the algorithm, we employ a couple of heuristics to reduce the computational effort (see Section "Computational efficiency" for details).

Following [2], we employ an iterative strategy, with Z_{stop} (the number of iterations) usually set to 12:

1. Define a set H of temporary haplotypes and populate H with a number of samples for each individual i in R. The samples are generated by drawing N_S times from a uniform probability distribution over all pairs of haplotypes which are compatible (ignoring read error or mutation) with an individual's genotypes $G_{R,i}$, and each haplotype pair is broken up into two separate haplotypes before insertion into H.

Missing data in the genotypes is carried over to the haplotypes. Set $Z := 1$.

2. Construct a haplotype graph model M based on H, as described below.
3. Set $H = \{\}$. For each reference individual i, draw N_S samples from $\mathrm{P}((h_1, h_2) \,|\, G_{R,i}, M \to \mathrm{HMM}_2)$, and add h_1 and h_2 to H.
4. Invert each haplotype in H.
5. Set $Z := Z + 1$, terminate if $Z > Z_{\text{stop}}$. Otherwise, go to step 2.

Our haplotype graph construction algorithm is a probabilistic generalization of the works of Browning and Browning [2], which allows for uncertainty and missing data in the set of estimated haplotypes H and a tailoring of the graph according to prior knowledge on regional LD structure ("localization"). The aim is to infer an accurate and computationally convenient haplotype graph model from the set of haplotypes H.

Suppose for a given number of levels T (corresponding to the lengths of the haplotypes in the set H) we have the most general possible haplotype graph topology, i.e. a tree for which every possible emission symbol has an edge at every vertex in the graph. Note that at this stage there are no probabilities assigned. Each $h \in H$ with no missing data corresponds to a unique path through the graph topology, and we say that h is *attached* to all vertices that the path passes through. However, we want to allow for missing data in H, and we also want to take into account the possibility that an error process may have modified the elements in H prior to observation. In the context of genetics, it is easy to see why this makes sense. For example, a SNP genotyping error may lead to a haplotype being present in H which does really not exist in the population.

We define a simple error process for the elements in H. We assume that this error process acts independently on each character position and that, if an error occurs (with probability m_B), a new observed value is drawn from a uniform distribution over possible alternative alleles at the affected position (this could, if desired, be easily generalized to more complex error models). If we observe string h_1 of length T, the likelihood that string

h_2 is the true underlying string is

$$\prod_{p=0\,..\,T-1} \left[I_{h_1(p)==h_2(p)} \times (1-m_B) + (1-I_{h_1(p)==h_2(p)}) \times \frac{m_B}{|A_p|-1}\right],$$

where we define $I_{h_1(p)==h_2(p)}$ to be 1 if the p-th symbol of h_1 is equal to the p-th symbol of h_2 and 0 otherwise. For simplicity, although m_B may capture other effects than error, we refer to m_B as the *graph building error* probability. $|A_p|$ is the number of available symbols at haplotype position p (the size of the model alphabet at p).

If we observe missing data, we want to treat it in an agnostic way, i.e. assume equal probabilities for each symbol in the model alphabet at the corresponding position.

We now probabilistically attach the haplotypes in H to the most general possible haplotype graph topology for T levels (in our implementation, we actually prune the tree as we move along the haplotypes – see Section "Computational efficiency" for details). For each vertex, we introduce a list of probability-weighted potentially attached haplotypes. At each level of the graph, the sum of attachment probabilities has to be 1 for each haplotype. All haplotypes are attached to the root vertex with probability 1 by defining the attachment probability $\mathrm{P}_H(v_0, h) := 1$ for all $h \in H$; they are then distributed along the graph according to our error model. That is, if haplotype h is attached to v_a at level $l(v_a)$ with probability y, and if the next observed haplotype symbol is $s \neq E_M$, we have the following attachment probabilities for the children v_b of v_a at level $l(v_a)+1$: if the edge (v_a, v_b) carries the attached symbol s, the attachment probability of h at v_b is $y \times (1-m_B)$, i.e. we define $\mathrm{P}_H(v_b, h) := \mathrm{P}_H(v_a, h) \times (1-m_B)$. Otherwise, the attachment probability is $y \times \frac{m_B}{|A_{l(v_a)}|-1}$, and we define $\mathrm{P}_H(v_b, h) := \mathrm{P}_H(v_a, h) \times \frac{m_B}{|A_{l(v_a)}|-1}$. If $s = E_M$, we attach h in an agnostic manner, i.e. we define $\mathrm{P}_H(v_b, h) := \mathrm{P}_H(v_a, h) \times \frac{1}{|A_{l(v_a)}|}$.

For notational convenience, let $\texttt{attached}(v)$ denote the set of haplotypes attached to v with attachment probability $\mathrm{P}_H(v, h) > 0$. To examine the structure of the graph topology with attached haplotypes, for each vertex v, we define a function $\texttt{count}(v, x)$. If x is the empty string $''$, $\texttt{count}(v, x)$ returns the expected number of haplotypes in H attached to v:

$$\texttt{count}(v,'') = \sum_{h \in \texttt{attached}(v)} \mathrm{P}_H(v,h)$$

If x is a string of length ≥ 1, $\texttt{count}(v,x)$ returns the expected number of haplotypes that continue with a specified suffix x of length $\texttt{len}(x) \geq 1$ to the right-hand side of v. x can be a partial or complete suffix, i.e. of length $1..[T - l(v)]$:

$$\texttt{count}(v,x) = \sum_{h \in \texttt{attached}(v)} \big(\mathrm{P}_H(v,h) \times \prod_{p=[l(v)]..[l(v)+\texttt{len}(x)-1]} \big[I_{h(p)==x(p)} \times (1-m_B)+ (1-I_{h(p)==x(p)}) \times \frac{m_B}{A_p - 1}\big]\big)$$

We complete the definition of a haplotype graph by specifying edge transition probabilities. Define $\mathrm{P}(e|v)$ as the probability to follow edge e conditional on being at vertex v, and let s denote the symbol that is attached to e. Then we set

$$\mathrm{P}(e|v) := \texttt{count}(v,s)/\texttt{count}(v,'')\,,$$

Figure 1 in this document illustrates the effect of the described algorithm: instead of taking one specified path through the graph topology, a haplotype's probability "flows" through the graph.

However, we observe i) that the resulting haplotype graph exhibits a considerable topological complexity, if built from a reasonably sized set H, possibly leading to computational difficulties in later stages and ii) that the topology of the graph is still the most general one. If we assume that H was actually sampled from a haplotype graph, and if we want to recover the original graph's underlying structure, we have to take into account the possibility that the original graph's structure may have been simpler, i.e. that one vertex in the original graph corresponds to more than one vertex on the same level in the current graph. Introducing a criterion of similarity that is based on comparing vertices' conditional suffix distributions addresses both points. Informally speaking, vertices with very similar suffix output distributions can be merged into one vertex to reduce computa-

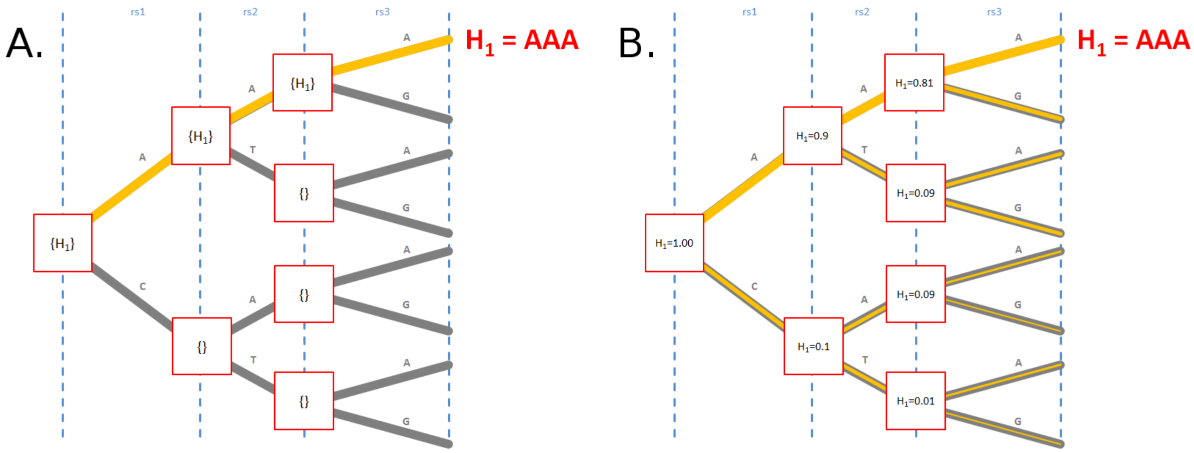


Figure 1: A: a non-probabilistic haplotype graph construction algorithm. Each haplotype in the set H follows one defined path (orange) through the graph's possible topology (orange and gray branches), here depicted for $H_1 = \mathtt{AAA}$. Each node (red squares) carries a list of attached haplotypes. B: the probabilistic haplotype graph construction algorithm presented in this chapter. Each haplotype in the set H induces a probability distribution over possible paths through the graph, here pictured as orange lines. The width of the lines indicates how probable a path is according to the path probability distribution (not drawn to scale). At each node, the path follows the edge carrying the haplotype's next symbol with probability $1 - m_B$, and the remaining probability mass is split over the remaining available edges. Each node carries a list of attached haplotypes with the respective attachment probability. The figure is based on a path distribution for "AAA", with the graph-building error probability m_B set to 0.1.

tional demands without substantially changing the model's haplotype frequencies. Also, if two vertices were actually identical or not distinguishable in an original haplotype graph model, we would expect their suffix output distributions to be similar (see [5] for a formal treatment).

We formalize the notion of similar suffix distributions following Ron et al. [5] and Browning and Browning [2] by defining the function $\mathtt{similar}(v_a, v_b)$ as the maximum difference between the two conditional suffix probability distributions of v_a and v_b:

$$\mathtt{similar}(v_a, v_b) := \max_{x \in S_{v_a, v_b}} \left| \mathrm{P}_{\text{Suffix}}(v_a, x) - \mathrm{P}_{\text{Suffix}}(v_b, x) \right| ,$$

where we define

$$\mathrm{P}_{\text{Suffix}}(v, x) := \mathtt{count}(v, x) / \mathtt{count}(v, '') \, .$$

S_{v_a, v_b} is defined as the set of possible suffixes originating from v_a or v_b (partial or complete). In order to accommodate the complex haplotype structure of the MHC, we include the edge label leading to a node as the first character of all suffixes.

We apply `similar` to all pairs of vertices (v_a, v_b) at all levels to identify pairs of vertices that can be merged. If $\texttt{similar}(v_a, v_b) < \epsilon$, two vertices are merged. We follow Browning and Browning [2] in using a variance-based threshold:

$$\epsilon := D \times \sqrt{N_S/2} \times (\texttt{count}(v_a,'') ^{-1} + \texttt{count}(v_b,'')^{-1})^{1/2} \,,$$

where D is a scale parameter (usually 0.8 here, determined by initial experiments) and N_S is the number of haplotype pair samples from each individual.

To merge v_a and v_b,

1. create a new vertex v_c at the same level as v_a and v_b

2. redirect all incoming edges of v_a and v_b to v_c, and for all $h \in \{\texttt{attached}(v_b) \cup \texttt{attached}(v_a)\}$, set $\mathrm{P}_H(v_c, h) := \mathrm{P}_H(v_b, h) + \mathrm{P}_H(v_a, h)$

3. attach all outgoing edges of v_b and v_a to v_c, and delete v_b and v_a.

4. note that step 2 will result in a structure violating the haplotype graph assumptions, as it will result in two edges (v_c, v_d) , $(v_c, v_{d'})$ with the same attached symbol. Merge v_d and $v_{d'}$ as described for all such cases (i.e. recursively from step 1, if necessary), and delete one of the two resulting edges leading to the new node replacing d and d'.

5. finally, update $\mathrm{P}(e|v)$ for all modified vertices and compute the similar function for v_c and all other vertices on the same level.

Figure 2 in this document illustrates the process of merging nodes. For notational convenience, we have assumed a fixed graph building error probability m_B here for all loci, but it is easy to see that this is not necessary.

Finally, we describe how to localize the graph construction process. Localization aims at incorporating prior knowledge on patterns of long-range LD into the graph-building process. Consider the following example to see why localization can be sensible. Suppose that two haplotypes from H are attached to v_a: '00A' and '11A' (the allele identifiers are arbitrary). Suppose further that a node v_b on the same level has also two attached

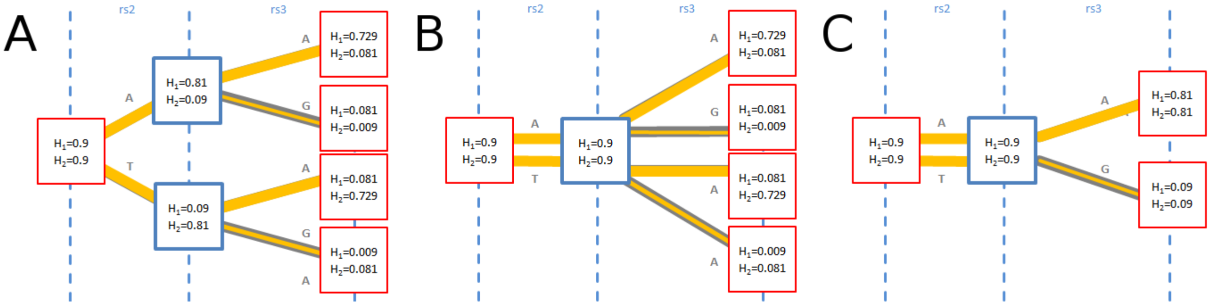


Figure 2: The essential steps of merging nodes in the probabilistic framework described here. A: two haplotypes (AAA and ATA) have been attached to the topology shown in Figure 1 in this document (the graph's first level is not shown) with $m_B = 0.1$. The conditional suffix distributions of two nodes (pictured as blue squares) are identical and the nodes will be merged. B: all outgoing edges from the two nodes have been attached to one newly created joint node (blue square). The resulting structure is no haplotype graph, because two edges emanating from the new node carry the same symbols as two other edges emanating from the same node. C: The nodes that the conflicting edges lead two are recursively merged, resulting in a haplotype graph structure.

haplotypes, '00B' and '11B'. If we compare the conditional suffix distributions, we find no difference for suffixes of length up to 2. For suffixes of length 3 and a small m_B, we find that the maximum difference is just below 0.5 (because none of the 3-character suffixes present in one vertex is present in the other one). Depending on our choice of ϵ, we may decide to merge the two vertices. The problem here is that the vertices actually exhibit quite different patterns of LD to the third position – the maximum conditional probability difference is almost 1. The localization element extends the function `similar` to take into account such situations for a set S_L of levels of predefined loci and could therefore prevent merging the two vertices.

Define the indicator function $I_{h(p)==s}$ to be 1 if haplotype h carries allele s at position p, and 0 otherwise. We define

$$\mathrm{P}_{\mathrm{LOCALIZE}}(v, s, p) :=$$
$$\frac{\sum_{h \in \texttt{attached}(v)} \mathrm{P}_H(v,h) \times (I_{h(p)==s} \times (1-m_B) + (1 - I_{h(p)==s}) \times \frac{m_B}{|A_p|-1})}{\texttt{count}(v,'')}.$$

We note that this conditional probability integrates over the uncertainty in the intermediate SNP genotypes and redefine the `similar` function to include all loci specified in S_L:

$$\texttt{similar}(v_a, v_b) := \max\{$$

$$\max_{x \in S_{v_a, v_b}} |\mathrm{P}(v_a, x) - \mathrm{P}(v_b, x)|,$$

$$\max_{\{p \in S_L, s \in A_p\}} |\mathrm{P}_{\mathrm{LOCALIZE}}(v_a, s, p) - \mathrm{P}_{\mathrm{LOCALIZE}}(v_b, s, p)|$$

$$\}\,.$$

Figure 3 in this document illustrates the localization feature.

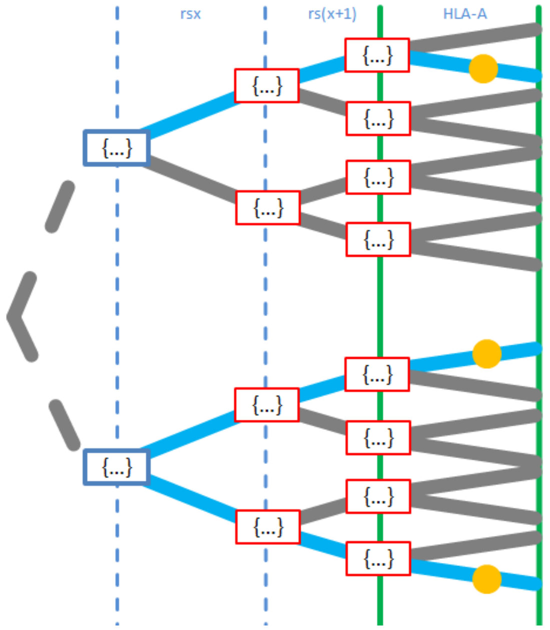


Figure 3: Localization at the example of an HLA locus. When comparing the conditional HLA allele probabilities for two nodes (blue squares) for a particular *HLA-A* allele (marked with an orange circle in the graph), the probabilities of all paths leading to this allele are added up (separately for each node). Note that the two blue paths for the lower node would count as two distinct suffixes without localization.

Computational efficiency

The algorithm we have described to build localized haplotype graphs from an uncertain set of haplotypes requires substantial computational resources: to calculate the conditional suffix distributions for each vertex, it is necessary to sum over all attached haplotypes with attachment probability > 0. Haplotype attachment distributions of single haplotypes are typically (depending on the uncertainty model and the number of alleles at the involved loci) very skewed: a few vertices at any level usually account for most of the available probability. Therefore, a threshold t is introduced: if $\mathrm{P}_H(v, h) < t$, $\mathrm{P}_H(v, h)$ is set to 0 and the removed probability mass is spread proportionally over all vertices with $\mathrm{P}_H(v, h) \geq t$. Also, when computing $\texttt{similar}(v_a, v_b)$ for two vertices, only such $x \in S_{v_a,v_b}$ that are present in at least one of the haplotypes attached to v_a or v_b are evaluated – suffixes x merely induced by error processes on both vertices will carry smaller probabilities than the original strings and therefore lead to a smaller absolute difference in probability.

HLA type inference

HLA loci are treated as multi-allelic SNPs, i.e. the observed HLA types are part of the haplotype strings H and appear as edges in the haplotype graph. SNP- and HLA-genotyped individuals can be used as input for the "iterative refinement" algorithm, without prior phasing. Only individuals with at least one genotyped 4-digit HLA allele are used for constructing the haplotype graph model, and 4- and 2-digit alleles are treated in the same way, i.e. as unrelated, separate entities (4-digit resolution specifies the primary structure of the classical HLA proteins, whereas 2-digit resolution refers to more general serological properties of the alleles).

To account for the long-range LD structure of the MHC region, the graph building algorithm is localized for all classical HLA loci but *B* and *DRB1* (see below). Usually, we set $m_B = m_S$.

Note that the results from step 3 of the model building algorithm can be used to quantitatively assess whether a lab-based HLA typing result in the reference dataset is consistent with the graph or not; the posterior probabilities follow from summing over the haplotype samples. To minimize the impact of mis-typed HLA alleles in the reference panel, after a specific number of graph-building and sampling iterations (usually 8), the number of sampled haplotypes for a specific individual is weighted by the internally estimated probability that the individual's lab-based HLA type is consistent with the graph.

We build locus-specific HLA haplotype graphs for windows of a specified size each side of the locus; 300 SNPs each side have been found to give good results. HLA type inference is carried out by sampling haplotype pairs from the diploid HMM, conditional on the observed genotypes $G_{I,i}$ for each individual i in the inference dataset and the haplotype graph: $\mathrm{P}((h_1, h_2) \,|\, G_{I,i}, M \rightarrow \mathrm{HMM}_2)$. This leads to posterior distributions over possible pairs of HLA types that can be processed in an uncertainty-aware way or thresholded. To call alleles, we first determine the most likely single allele for each individual and then the most likely second allele, conditional on that individual carrying the first allele. We use the marginal probability to observe the first allele (i.e. summed over all samples from the haplotype pair distribution) as quality score ("allele-specific posterior probability") for the first allele and the joint probability for the first and the second allele as quality score for the second allele.

Properties of the presented model and parameter inference

We have presented a generalized haplotype graph construction algorithm, related to the BEAGLE algorithm [2] and earlier work in computational linguistics [5], which probabilistically attaches haplotypes to vertices while building the graph. We have introduced two additional parameters: a graph building error parameter m_B and the set of localization loci S_L that can be used to adapt graph construction to complex patterns of LD. Our algorithm also allows for missing data.

We briefly discuss some properties of the generalized model:

- The error model we have introduced leads to a relative decline of the importance of long-range haplotype differences in terms of collapsing vertices: $|P_{\mathrm{Suffix}}(v_1, x) - P_{\mathrm{Suffix}}(v_2, x)|$ is decreased for x with large differences. This depends on d, the scaling parameter in the collapsing criterion, and m_B, the error probability.

- We expect the generalized model to be potentially useful in other applications than the one considered here. For example, if a haplotype graph is to be constructed for a set of experimentally determined haplotypes (from single chromosome sequencing, say), the uncertainty model for the graph-building step we have introduced can be used to model read errors.

- The described algorithm can deal with missing data in the set of haplotypes in a straightforward way by defining a probability distribution on missing characters, e.g. a uniform distribution. This property allows us not having to guess genotypes for the first iteration of graph-building. Although the algorithm as described here imputes missing genotypes in the reference panel during the first sampling process, the missing data status could as well be preserved in the sampled haplotypes and could be carried over to later stages. As the reference panels we are dealing with are consistently typed on dense sets of markers, we have decided against this possibility here. However, under other circumstances, for example when SNP coverage in the reference panel varies strongly, not imputing missing SNP data may turn out to be beneficial [6].

- Treating HLA alleles as multiallelic SNPs leads to a couple of useful properties in learning and inference settings. The graph itself can reflect patterns of long-range linkage disequilibrium between HLA alleles – HLA and SNP genotypes are used to infer the graph structure in a combined manner, and there is no requirement that all individuals be typed at the same set of HLA loci. Consider, for example, an inference dataset with *HLA-DRB1*-typed individuals, but lacking information for *HLA-DQB1*. Providing the *DRB1* genotypes as well as the SNP genotypes enables the model to use partial HLA type information in inferring missing bits of the complete HLA type (depending on the particular structure of the graph used for inference, of course).

Choosing optimal parameters for building haplotype graphs and for inference is an important direction for further research. For the experiments presented in the main text, we have used: $m_B = m_S = 0.002, \quad t = 0.001, \quad N_S = 50, \quad D = 0.8\,.$

Although standard statistical techniques like Maximum Likelihood and Markov Chain Monte Carlo could be applied in theory, the computational costs to do so seem prohibitive at the moment. In the context of this paper, our main purpose is statistical HLA type imputation, and we measure the fit of model and parameterization by the validation experiments presented in the main text. In order to justify the introduction of additional parameters, we have repeated some of the experiments presented in Leslie et al. [7] and Dilthey et al. [8]. We have used CEU HapMap data as a reference panel, constructed haplotype graphs and imputed HLA types into a subset of the BC58 (all data exactly as described in our earlier papers). The results are summarized in Table S1. The column "HLA*IMP:02" refers to the full model (with parameters adapted to accommodate the much reduced panel size). In column I, the error probabilities for sampling from the graph and for building the graph are set to 0 (all other parameters equal to the full model). In column II, the error probability for building the graph is set to 0, and in column III, the error probability for sampling from the graph is set to 0. We find that the full model outperforms each of the reduced versions. In column IV, we have deactivated HLA localization. Interestingly, the full model only yields better results at *A* and *DQB1*, whereas the results at *B* and *DRB1* are worse. This may relate to classical typing problems, potentially associated with hypervariability (*B*) and nearby structural variation (*DRB1*), or it may indicate that localization does not improve imputation accuracy. Until further investigation, we deactivate localization for *B* and *DRB1*.

References

1. Marchini J, Howie B (2010) Genotype imputation for genome-wide association studies. Nat Rev Genet 11: 499-511.

2. Browning SR, Browning BL (2007) Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. Am J Hum Genet 81: 1084-97.

3. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the Ieee 77: 257-286.

4. Browning BL, Yu Z (2009) Simultaneous genotype calling and haplotype phasing improves genotype accuracy and reduces false-positive associations for genome-wide association studies. Am J Hum Genet 85: 847-61.

5. Ron D, Singer Y, Tishby N (1998) On the learnability and usage of acyclic probabilistic finite automata. Journal of Computer and System Sciences 56: 133-152.

6. Howie BN, Donnelly P, Marchini J (2009) A flexible and accurate genotype imputa-

tion method for the next generation of genome-wide association studies. PLoS Genet 5: e1000529.

7. Leslie S, Donnelly P, McVean G (2008) A statistical method for predicting classical HLA alleles from SNP data. Am J Hum Genet 82: 48-56.

8. Dilthey AT, Moutsianas L, Leslie S, McVean G (2011) HLA*IMP – an integrated framework for imputing classical HLA alleles from SNP genotypes. Bioinformatics 27: 968-72.

For support visit **thermofisher.com/support** or email **techsupport@lifetech.com**
thermofisher.com

3 June 2019

ThermoFisher
SCIENTIFIC