# HIBAG – an R Package for HLA Genotype Imputation with Attribute Bagging

Xiuwen Zheng

Apr 10, 2015

## Contents



## 1 Overview

The human leukocyte antigen (HLA) system, located in the major histocompatibility complex (MHC) on chromosome 6p21.3, is highly polymorphic. This region has been shown to be important in human disease, adverse drug reactions and organ transplantation [1]. HLA genes play a role in the immune system and autoimmunity as they are central to the presentation of antigens for recognition by T cells. Since they have to provide defense against a great diversity of environmental microbes, HLA genes must be able to present a wide range of peptides. Evolutionary pressure at these loci have given rise to a great deal of functional diversity. For example, the *HLA–B* locus has 1,898 four-digit alleles listed in the April 2012 release of the IMGT-HLA Database [2] (`http://www.ebi.ac.uk/imgt/hla/`).

Classical HLA genotyping methodologies have been predominantly developed for tissue typing purposes, with sequence based typing (SBT) approaches currently considered the gold standard. While there is widespread availability of vendors offering HLA genotyping services, the complexities involved in performing this to the standard required for diagnostic purposes make using a SBT approach time-consuming and cost-prohibitive for most research studies wishing to look in detail at the involvement of classical HLA genes in disease.

Here we introduce a new prediction method for **H**LA **I**mputation using attribute **BAG**ging, HIBAG, that is highly accurate, computationally tractable, and can be used with published parameter estimates, eliminating the need to access large training samples [3]. It relies on a training set with known HLA and SNP genotypes, and combines the concepts of attribute bagging with haplotype inference from unphased SNPs and HLA types. Attribute bagging is a technique for improving the accuracy and stability of classifier ensembles using bootstrap aggregating and random variable selection [4, 5, 6]. In this case, individual classifiers are created which utilize a subset of SNPs to predict HLA types and haplotype frequencies estimated from a training data set of SNPs and HLA types. Each of the classifiers employs a variable selection algorithm with a random component to select a subset of the SNPs. HLA type predictions are determined by maximizing the average posterior probabilities from all classifiers.

## 2 Features

1. HIBAG can be used by researchers with published parameter estimates (http://www.biostat.washington.edu/~bsweir/HIBAG/) instead of requiring access to large training sample datasets.
2. A typical HIBAG parameter file contains only haplotype frequencies at different SNP subsets rather than individual training genotypes.
3. SNPs within the xMHC region (chromosome 6) are used for imputation.
4. HIBAG employs unphased genotypes of unrelated individuals as a training set.
5. HIBAG supports parallel computing with R.

## 3 Examples

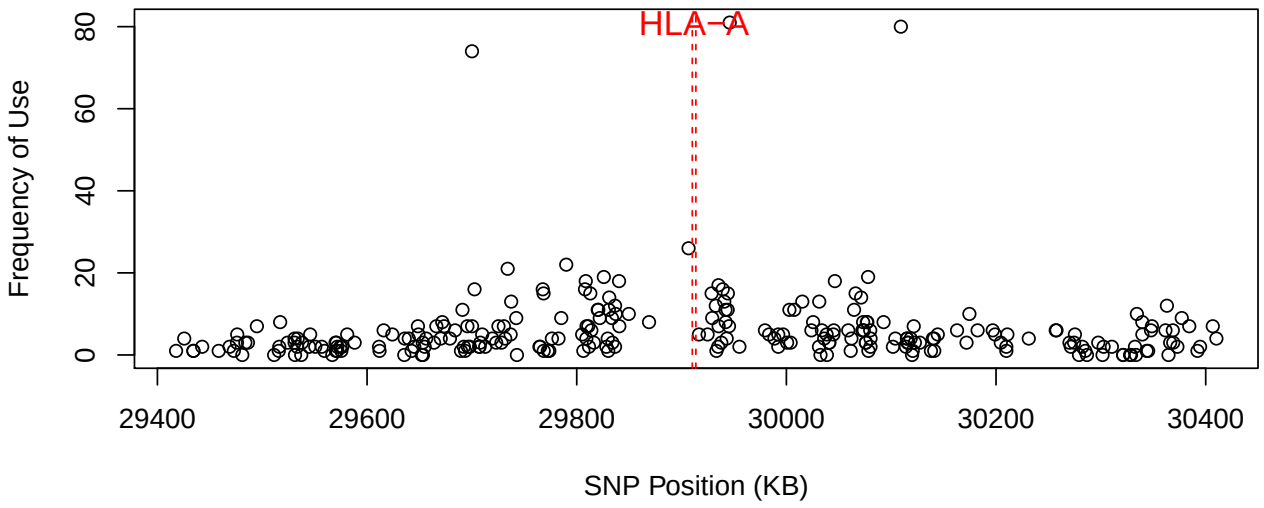
```
library(HIBAG)
```

### 3.1 Pre-fit HIBAG models for HLA imputation

```
# load the published parameter estimates from European ancestry
#   e.g., filename <- "HumanOmniExpressExome-European-HLA4-hg19.RData"
#   here, we use example data in the package
filename <- system.file("extdata", "ModelList.RData", package="HIBAG")
model.list <- get(load(filename))

# HLA imputation at HLA-A
hla.id <- "A"
model <- hlaModelFromObj(model.list[[hla.id]])
summary(model)
```

```
## Gene: HLA - A
## Training dataset: 60 samples X 266 SNPs
##  # of HLA alleles: 14
##  # of individual classifiers: 100
##  total # of SNPs used: 245
##  average # of SNPs in an individual classifier: 15.92, sd: 2.43, min: 10, max: 24
##  average # of haplotypes in an individual classifier: 39.89, sd: 14.28, min: 18, max: 87
```

```
##  average out-of-bag accuracy: 93.77%, sd: 4.56%, min: 78.95%, max: 100.00%
## Genome assembly: hg19

# SNPs in the model
head(model$snp.id)

## [1] "rs2523442" "rs9257863" "rs2107191" "rs4713226" "rs2746150" "rs1233492"

head(model$snp.position)

## [1] 29417816 29425583 29434294 29434413 29442700 29458476

# plot SNP information
plot(model)
```




```
################################################################################
# Import your PLINK BED file
#
yourgeno <- hlaBED2Geno(bed.fn=".bed", fam.fn=".fam", bim.fn=".bim")
summary(yourgeno)

# best-guess genotypes and all posterior probabilities
pred.guess <- predict(model, yourgeno, type="response+prob")
summary(pred.guess)

pred.guess$value
pred.guess$postprob
```

## 3.2 Build a HIBAG Model for HLA Genotype Imputation

```
library(HIBAG)

# load HLA types and SNP genotypes in the package
data(HLA_Type_Table, package="HIBAG")
data(HapMap_CEU_Geno, package="HIBAG")

# a list of HLA genotypes
# e.g., train.HLA <- hlaAllele(sample.id, H1=c("01:02", "05:01", ...),
#                         H2=c("02:01", "03:01", ...), locus="A")
#     the HLA type of the first individual is 01:02/02:01,
#                 the second is 05:01/03:01, ...
hla.id <- "A"
train.HLA <- hlaAllele(HLA_Type_Table$sample.id,
    H1 = HLA_Type_Table[, paste(hla.id, ".1", sep="")],
    H2 = HLA_Type_Table[, paste(hla.id, ".2", sep="")],
    locus=hla.id, assembly="hg19")

# selected SNPs:
#    the flanking region of 500kb on each side,
#    or an appropriate flanking size without sacrificing predictive accuracy
region <- 500   # kb
snpid <- hlaFlankingSNP(HapMap_CEU_Geno$snp.id,
    HapMap_CEU_Geno$snp.position, hla.id, region*1000, assembly="hg19")
length(snpid)

# training genotypes
#   import your PLINK BED file,
#   e.g., train.geno <- hlaBED2Geno(bed.fn=".bed", fam.fn=".fam", bim.fn=".bim")
#   or,
train.geno <- hlaGenoSubset(HapMap_CEU_Geno,
    snp.sel = match(snpid, HapMap_CEU_Geno$snp.id))
summary(train.geno)

# train a HIBAG model
set.seed(100)
model <- hlaAttrBagging(train.HLA, train.geno, nclassifier=100)
```

```
summary(model)
```

```
## Gene: HLA - A
## Training dataset: 60 samples X 266 SNPs
##  # of HLA alleles: 14
##  # of individual classifiers: 100
##  total # of SNPs used: 245
##  average # of SNPs in an individual classifier: 15.92, sd: 2.43, min: 10, max: 24
##  average # of haplotypes in an individual classifier: 39.89, sd: 14.28, min: 18, max: 87
##  average out-of-bag accuracy: 93.77%, sd: 4.56%, min: 78.95%, max: 100.00%
## Genome assembly: hg19
```

### 3.3 Build and Predict in Parallel

```
library(HIBAG)
library(parallel)

# load HLA types and SNP genotypes in the package
data(HLA_Type_Table, package="HIBAG")
data(HapMap_CEU_Geno, package="HIBAG")

# a list of HLA genotypes
# e.g., train.HLA <- hlaAllele(sample.id, H1=c("01:02", "05:01", ...),
#                         H2=c("02:01", "03:01", ...), locus="A")
#     the HLA type of the first individual is 01:02/02:01,
#                  the second is 05:01/03:01, ...
hla.id <- "A"
train.HLA <- hlaAllele(HLA_Type_Table$sample.id,
    H1 = HLA_Type_Table[, paste(hla.id, ".1", sep="")],
    H2 = HLA_Type_Table[, paste(hla.id, ".2", sep="")],
    locus=hla.id, assembly="hg19")

# selected SNPs:
#    the flanking region of 500kb on each side,
#    or an appropriate flanking size without sacrificing predictive accuracy
region <- 500   # kb
snpid <- hlaFlankingSNP(HapMap_CEU_Geno$snp.id,
    HapMap_CEU_Geno$snp.position, hla.id, region*1000, assembly="hg19")
length(snpid)

# training genotypes
#   import your PLINK BED file,
#   e.g., train.geno <- hlaBED2Geno(bed.fn=".bed", fam.fn=".fam", bim.fn=".bim")
#   or,
train.geno <- hlaGenoSubset(HapMap_CEU_Geno,
    snp.sel = match(snpid, HapMap_CEU_Geno$snp.id))
summary(train.geno)


# Create an environment with an appropriate cluster size,
#   e.g., 2 -- # of (CPU) cores
cl <- makeCluster(2)

# Building ...
set.seed(1000)
hlaParallelAttrBagging(cl, train.HLA, train.geno, nclassifier=100,
    auto.save="AutoSaveModel.RData", stop.cluster=TRUE)
model.obj <- get(load("AutoSaveModel.RData"))
model <- hlaModelFromObj(model.obj)
summary(model)
```

### 3.4 Evaluate Overall Accuracy, Sensitivity, Specificity, etc

The function `hlaReport()` can be used to automatically generate a tex or HTML report when a validation dataset is available.

```
library(HIBAG)

# load HLA types and SNP genotypes in the package
data(HLA_Type_Table, package="HIBAG")
data(HapMap_CEU_Geno, package="HIBAG")

# make a list of HLA types
hla.id <- "A"
hla <- hlaAllele(HLA_Type_Table$sample.id,
    H1 = HLA_Type_Table[, paste(hla.id, ".1", sep="")],
    H2 = HLA_Type_Table[, paste(hla.id, ".2", sep="")],
    locus=hla.id, assembly="hg19")

# divide HLA types randomly
set.seed(100)
hlatab <- hlaSplitAllele(hla, train.prop=0.5)
names(hlatab)
```

```
## [1] "training"   "validation"
```

```
summary(hlatab$training)
```

```
## Gene: HLA - A
## Range: [29910247bp, 29913661bp] on hg19
## # of samples: 34
## # of unique HLA alleles: 14
## # of unique HLA genotypes: 24
```

```
summary(hlatab$validation)
```

```
## Gene: HLA - A
## Range: [29910247bp, 29913661bp] on hg19
## # of samples: 26
## # of unique HLA alleles: 12
## # of unique HLA genotypes: 15
```

```
# SNP predictors within the flanking region on each side
region <- 500   # kb
snpid <- hlaFlankingSNP(HapMap_CEU_Geno$snp.id,
    HapMap_CEU_Geno$snp.position, hla.id, region*1000, assembly="hg19")
length(snpid)
```

```
## [1] 275
```

```
# training and validation genotypes
train.geno <- hlaGenoSubset(HapMap_CEU_Geno,
    snp.sel = match(snpid, HapMap_CEU_Geno$snp.id),
    samp.sel = match(hlatab$training$value$sample.id,
        HapMap_CEU_Geno$sample.id))
summary(train.geno)
```

```
## SNP genotypes:
##  34 samples X 275 SNPs
##  SNPs range from 29417816bp to 30410205bp on hg19
## Missing rate per SNP:
##  min: 0, max: 0.0882353, mean: 0.0864171, median: 0.0882353, sd: 0.0123037
## Missing rate per sample:
##  min: 0, max: 0.974545, mean: 0.0864171, median: 0, sd: 0.280441
## Minor allele frequency:
##  min: 0, max: 0.483871, mean: 0.216655, median: 0.193548, sd: 0.135696
## Allelic information:
## A/C A/G C/T G/T
##  21  97 125  32
```

```
test.geno <- hlaGenoSubset(HapMap_CEU_Geno, samp.sel=match(
    hlatab$validation$value$sample.id, HapMap_CEU_Geno$sample.id))
```

```
# train a HIBAG model
set.seed(100)
model <- hlaAttrBagging(hlatab$training, train.geno, nclassifier=100)
```

```
summary(model)
```

```
## Gene: HLA - A
## Training dataset: 34 samples X 266 SNPs
##  # of HLA alleles: 14
##  # of individual classifiers: 100
##  total # of SNPs used: 244
##  average # of SNPs in an individual classifier: 14.60, sd: 2.76, min: 10, max: 21
##  average # of haplotypes in an individual classifier: 39.68, sd: 16.31, min: 15, max: 84
##  average out-of-bag accuracy: 83.80%, sd: 8.69%, min: 62.50%, max: 100.00%
## Genome assembly: hg19
```

```
# validation
pred <- predict(model, test.geno)
```

```
## HIBAG model: 100 individual classifiers, 266 SNPs, 14 unique HLA alleles.
## Predicting based on the averaged posterior probabilities from all individual classifiers.
## Model assembly: hg19, SNP assembly: hg19
## No allelic strand orders are switched.
## The number of samples: 26.
## Predicting: Thu Apr 16 22:32:06 2015 0%
## Predicting: Thu Apr 16 22:32:06 2015 100%
```

```
# compare
```

```
comp <- hlaCompareAllele(hlatab$validation, pred, allele.limit=model,
    call.threshold=0)
comp$overall
```

```
##   total.num.ind crt.num.ind crt.num.haplo   acc.ind acc.haplo call.threshold n.call
## 1            26          24            50 0.9230769 0.9615385              0     26
##   call.rate
## 1         1
```

Output to plain text format:

```
# report overall accuracy, per-allele sensitivity, specificity, etc
hlaReport(comp, type="txt")
```

```
## Allele Num. Freq. Num. Freq. CR ACC SEN SPE PPV NPV Miscall
##  Train Train Valid. Valid. (%) (%) (%) (%) (%) (%) (%)
## ----
## Overall accuracy: 96.2%, Call rate: 100.0%
## 01:01 14 0.2059 11 0.2115 100.0 98.1 100.0 97.6 91.7 100.0 --
## 02:01 23 0.3382 20 0.3846 100.0 98.1 95.0 100.0 100.0 97.0 29:02 (100)
## 02:06 1 0.0147 0 0 -- -- -- -- -- -- --
## 03:01 4 0.0588 5 0.0962 100.0 100.0 100.0 100.0 100.0 100.0 --
## 11:01 3 0.0441 2 0.0385 100.0 100.0 100.0 100.0 100.0 100.0 --
## 23:01 2 0.0294 1 0.0192 100.0 100.0 100.0 100.0 100.0 100.0 --
## 24:02 6 0.0882 5 0.0962 100.0 98.1 80.0 100.0 100.0 97.9 01:01 (100)
## 24:03 1 0.0147 0 0 -- -- -- -- -- -- --
## 25:01 3 0.0441 2 0.0385 100.0 100.0 100.0 100.0 100.0 100.0 --
## 26:01 2 0.0294 1 0.0192 100.0 100.0 100.0 100.0 100.0 100.0 --
## 29:02 3 0.0441 1 0.0192 100.0 98.1 100.0 98.0 50.0 100.0 --
## 31:01 2 0.0294 1 0.0192 100.0 100.0 100.0 100.0 100.0 100.0 --
## 32:01 2 0.0294 2 0.0385 100.0 100.0 100.0 100.0 100.0 100.0 --
## 68:01 2 0.0294 1 0.0192 100.0 100.0 100.0 100.0 100.0 100.0 --
```

Output to a tex file, and please add "\usepackage{longtable}" to your tex file:

```
# report overall accuracy, per-allele sensitivity, specificity, etc
hlaReport(comp, type="tex", header=FALSE)
```

Table 1: The sensitivity (SEN), specificity (SPE), positive predictive value (PPV), negative predictive value (NPV) and call rate (CR).

| Allele | Num. Train | Freq. Train | Num. Valid. | Freq. Valid. | CR (%) | ACC (%) | SEN (%) | SPE (%) | PPV (%) | NPV (%) | Miscall (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Overall accuracy: 96.2%, Call rate: 100.0%* | | | | | | | | | | | |
| 01:01 | 14 | 0.2059 | 11 | 0.2115 | 100.0 | 98.1 | 100.0 | 97.6 | 91.7 | 100.0 | – |
| 02:01 | 23 | 0.3382 | 20 | 0.3846 | 100.0 | 98.1 | 95.0 | 100.0 | 100.0 | 97.0 | 29:02 (100) |
| 02:06 | 1 | 0.0147 | 0 | 0 | – | – | – | – | – | – | – |
| 03:01 | 4 | 0.0588 | 5 | 0.0962 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |
| 11:01 | 3 | 0.0441 | 2 | 0.0385 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |
| 23:01 | 2 | 0.0294 | 1 | 0.0192 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |

Continued on next page ...

Table 1 – Continued from previous page

| Allele | Num. Train | Freq. Train | Num. Valid. | Freq. Valid. | CR (%) | ACC (%) | SEN (%) | SPE (%) | PPV (%) | NPV (%) | Miscall (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 24:02 | 6 | 0.0882 | 5 | 0.0962 | 100.0 | 98.1 | 80.0 | 100.0 | 100.0 | 97.9 | 01:01 (100) |
| 24:03 | 1 | 0.0147 | 0 | 0 | – | – | – | – | – | – | – |
| 25:01 | 3 | 0.0441 | 2 | 0.0385 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |
| 26:01 | 2 | 0.0294 | 1 | 0.0192 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |
| 29:02 | 3 | 0.0441 | 1 | 0.0192 | 100.0 | 98.1 | 100.0 | 98.0 | 50.0 | 100.0 | – |
| 31:01 | 2 | 0.0294 | 1 | 0.0192 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |
| 32:01 | 2 | 0.0294 | 2 | 0.0385 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |
| 68:01 | 2 | 0.0294 | 1 | 0.0192 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | – |

## 3.5 Release HIBAG Models without Confidential Information

```
library(HIBAG)

# make a list of HLA types
hla.id <- "DQA1"
hla <- hlaAllele(HLA_Type_Table$sample.id,
    H1 = HLA_Type_Table[, paste(hla.id, ".1", sep="")],
    H2 = HLA_Type_Table[, paste(hla.id, ".2", sep="")],
    locus=hla.id, assembly="hg19")

# training genotypes
region <- 500   # kb
snpid <- hlaFlankingSNP(HapMap_CEU_Geno$snp.id, HapMap_CEU_Geno$snp.position,
    hla.id, region*1000, assembly="hg19")
train.geno <- hlaGenoSubset(HapMap_CEU_Geno,
    snp.sel = match(snpid, HapMap_CEU_Geno$snp.id),
    samp.sel = match(hla$value$sample.id, HapMap_CEU_Geno$sample.id))

set.seed(1000)
model <- hlaAttrBagging(hla, train.geno, nclassifier=100)
summary(model)

# remove unused SNPs and sample IDs from the model
mobj <- hlaPublish(model,
    platform = "Illumina 1M Duo",
    information = "Training set -- HapMap Phase II",
    warning = NULL,
    rm.unused.snp=TRUE, anonymize=TRUE)

save(mobj, file="Your_HIBAG_Model.RData")
```

## 3.6 Release a Collection of HIBAG Models

```
# assume the HIBAG models are stored in R objects: mobj.A, mobj.B, ...

ModelList <- list()
ModelList[["A"]] <- mobj.A
ModelList[["B"]] <- mobj.B
...

# save to an R data file
save(ModelList, file="HIBAG_Model_List.RData")
```

## 4 Resources

1. Allele Frequency Net Database (AFND): http://www.allelefrequencies.net.
2. IMGT/HLA Database: http://www.ebi.ac.uk/imgt/hla.
3. HLA Nomenclature: G Codes (http://hla.alleles.org/alleles/g_groups.html) and P Codes (http://hla.alleles.org/alleles/p_groups.html) for reporting of ambiguous allele typings.

## 5 Session Info

```
toLatex(sessionInfo())
```

- R version 3.2.0 (2015-04-16), x86_64-unknown-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=C, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: HIBAG 1.4.0
- Loaded via a namespace (and not attached): BiocStyle 1.6.0, digest 0.6.8, evaluate 0.6, formatR 1.1, highr 0.4.1, htmltools 0.2.6, knitr 1.9, rmarkdown 0.5.1, stringr 0.6.2, tools 3.2.0, yaml 2.1.13

## References

[1] Takashi Shiina, Kazuyoshi Hosomichi, Hidetoshi Inoko, and Jerzy Kulski. The HLA genomic loci map: expression, interaction, diversity and disease. *Journal of Human Genetics*, 54(1):15–39, 2009. doi:10.1038/jhg.2008.5.

[2] James Robinson, Jason A. Halliwell, Hamish McWilliam, Rodrigo Lopez, Peter Parham, and Steven G.E. Marsh. The IMGT/HLA database. *Nucleic Acids Res*, 41(Database issue):1222–1227, Jan 2013. doi:10.1093/nar/gks949.

[3] Xiuwen Zheng, Judong Shen, Charles Cox, Jonathan C. Wakefield, Margaret G. Ehm, Matthew R. Nelson, and Bruce S. Weir. HIBAG – HLA genotype imputation with attribute bagging. *Pharmacogenomics J*, May 2013. doi:10.1038/tpj.2013.18.

[4] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996. doi:10.1023/A:1018054314350.

[5] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001. doi:10.1023/A:1010933404324.

[6] Robert Bryll, Ricardo Gutierrez-Osuna, and Francis Quek. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36(6):1291–1302, 2003. doi:10.1016/S0031-3203(02)00121-8.