HIBAG: an R Package for HLA Genotype Imputation with Attribute Bagging

Xiuwen Zheng

March 5, 2014

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

3.1 The pre-fit HIBAG models for HLA imputation

```
library(HIBAG)

# Load the published parameter estimates from European ancestry
model.list <- get(load("European-HLA4.RData"))

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

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

# SNPs in the model
head(model$snp.id)
# "rs2523442" "rs9257863" "rs2107191" "rs4713226" "rs1362076" "rs7751705"
head(model$snp.position)
# 29525796 29533563 29542274 29542393 29549148 29549597

# plot SNP information
plot(model)

# 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)

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

```
# The HLA type of the first individual is 01:02/02:01, the second is 05:01/03:01, ...
train.HLA <- hlaAllele(geno$sample.id, H1=c("01:02", "05:01", ...),
    H2=c("02:01", "03:01", ...), locus="A")

# Or the HLA types are saved in a text file "YourHLATypes.txt":
#   SampleID Allele1 Allele2
#   NA001101 01:02 02:01
#   NA001201 05:01 03:01
#   ...
D <- read.table("YourHLATypes.txt", header=TRUE, stringsAsFactors=FALSE)
train.HLA <- hlaAllele(D$SampleID, H1=D$Allele1, H2=D$Allele2, locus="A")

summary(train.HLA)

# Selected SNPs, two options:
# 1) the flanking region of 500kb on each side,
#    or an appropriate flanking size without sacrificing predictive accuracy
  snpid <- hlaFlankingSNP(geno$snp.id, geno$snp.position, "A", 500*1000)
# 2) the SNPs in our pre-fit models
  model.list <- get(load("European-HLA4.RData"))
  snpid <- model.list[["A"]]$snp.id
# Subset training SNP genotypes
train.geno <- hlaGenoSubset(geno, snp.sel=match(snpid, geno$snp.id))

# Building ...
set.seed(1000)
model <- hlaAttrBagging(train.HLA, train.geno, nclassifier=100, verbose.detail=TRUE)
summary(model)

# Save your model
model.obj <- hlaModelToObj(model)
save(model.obj, file="your_model.RData")

# Predict ...
model.obj <- get(load("your_model.RData"))
model <- hlaModelFromObj(model.obj)
# best-guess genotypes and all posterior probabilities
pred.guess <- predict(model, newgeno, type="response+prob")
summary(pred.guess)
pred.guess$value
pred.guess$postprob
```

3.3 Build and predict in parallel

```
library(parallel)
library(HIBAG)

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

# The HLA type of the first individual is 01:02/02:01, the second is 05:01/03:01, ...
train.HLA <- hlaAllele(geno$sample.id, H1=c("01:02", "05:01", ...),
    H2=c("02:01", "03:01", ...), locus="A")

# Or the HLA types are saved in a text file "YourHLATypes.txt":
#   SampleID Allele1 Allele2
#   NA001101 01:02 02:01
#   NA001201 05:01 03:01
#   ...
D <- read.table("YourHLATypes.txt", header=TRUE, stringsAsFactors=FALSE)
train.HLA <- hlaAllele(D$SampleID, H1=D$Allele1, H2=D$Allele2, locus="A")

summary(train.HLA)

# Create an environment with an appropriate cluster size
cl <- makeCluster(8)

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

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

3.4 Evaluate overall accuracy, sensitivity, specificity, etc

```
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 "hlaAlleleClass" object
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)
# "training"   "validation"
summary(hlatab$training)
summary(hlatab$validation)

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

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

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

# 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
# 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   --
# ......

hlaReport(comp, type="tex")
```

3.5 Release HIBAG models without confidential information

```
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 "hlaAlleleClass" object
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")

# 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: http://hla.alleles.org/alleles/g_groups.html.

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.

[2] J Robinson, J Halliwell, H McWilliam, R Lopez, P Parham, and S Marsh. The IMGT/HLA database. *Nucleic Acids Res*, 41(Database issue):1222–1227, Jan 2013.

[3] X Zheng, J Shen, C Cox, J C Wakefield, M G Ehm, M R Nelson, and B S Weir. HIBAG – HLA genotype imputation with attribute bagging. *Pharmacogenomics J*, May 2013.

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

[5] Leo Breiman. Random forests. *Mach. Learn.*, 45:5–32, 2001.

[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.