Multivariate Analysis Publication Code Part 1: mvBIMBAM

Jerry Z. Zhang, Lacey W. Heinsberg, and Daniel E. Weeks
Department of Human Genetics
University of Pittsburgh

April 05, 2022

Contents

1 Overview

Here we illustrate how we carried out the mvBIMBAM analyses.

Please see our README.md for instructions on installing mvBIMBAM and notes about this example code.

2 Load libraries

Load the libraries needed to run the code.

```r
library(tidyverse)
select = dplyr::select
library(ggplot2)
library(preprocessCore)
library(pander)
library(stringr)
```

3 Prepare data for mvBIMBAM

As with many programs (e.g., PLINK), mvBIMBAM requires the data to be in a specific format for analysis. In this section of the example code, we are preparing and formatting the data for analysis.

3.1 Read in the synthetic data set

Please see the README.md for information regarding the example synthetic data set and pre-processing instructions if you will be adapting this code to perform the analyses in your own data set.

```r
# Read in the synthetic data set created for use with this example analysis code
df_synth <- readRDS("git_synth_data.rds")

# Define the phenotypes of interest
# Anthropometry
anthro.traits <- c("BMI", "Height", "Weight", "FFM", "FM", "WHR", "Abd_Circ", "Hip_Circ")
# Lipids
lipids.traits <- c("HDL_C", "NetTG", "Cholesterol")
# Both
all.traits <- c(anthro.traits, lipids.traits)

# Phenotype abbreviations
# BMI=body mass index; FFM=fat-free mass; FM=fat mass; WHR=waist hip ratio;
# Abd_Circ=abdominal circumference; Hip_Circ=hip circumference;
# HDL_C=HDL cholesterol; NetTG=net triglycerides
```

In this example, we are preparing to analyze 'all.traits'.

3.2 Clean up data

```r
# Create data frames for regressing out covariates
# Note: Currently, mvBIMBAM does not allow missing phenotypes for the
# multivariate phenotype analysis, so select complete cases only
# The first data frame (df_i1_regress) contains all variables of interest ordered
# and filtered for complete cases
df_i1_regress <- df_synth %>%
  select(rs373863828, all_of(all.traits), Dec_Age, Gender) %>%
  filter(complete.cases(.))
# The second data frame (df_i1) contains all variables except the covariates that
# will be regressed out (in this example, age and gender)
df_i1 <- df_synth %>%
  select(rs373863828, all_of(all.traits)) %>%
  filter(complete.cases(.))
```

```
# Recode 0, 1, 2 genotypes to AA, AG, GG for later use in mvBIMBAM
df_synth <- df_synth %>% mutate(rs373863828_C = case_when(
                                    rs373863828 == 2 ~ "AA",
                                    rs373863828 == 1 ~ "AG",
                                    rs373863828 == 0 ~ "GG"))

# Create Genotype (G) and Phenotype (Y) matrices
G <- as.matrix(df_i1 %>% select(rs373863828))
Y <- as.matrix(df_i1 %>% select(all_of(all.traits)))
```

3.3 Normalize and adjust data for covariates

In this example, we are adjusting our phenotypes of interest for the covariates age and sex (labeled as “gender” in this synthetic data set) using ordinary linear regression models.

As described in the paper, the sensitivity of the Bayesian multivariate mvBIMBAM framework to outlier values and non-normality also necessitates the normalization of phenotypes. As shown below, residualized phenotypes (i.e., adjusted for age/sex) are order quantile-normalized.

```
# Create a function to perform residual adjustment for covariates (in this example,
# we are adjusting for age and gender)
f_quantile_norm_resid <- function(Y, df) {
  {o = apply(Y, 2, function(x) resid(lm(x ~ Dec_Age + Gender, data = df)))}
  return(o)
}

# Create function to 'super quantile normalize' the data
f_quantile_normalize_adjust <- function(Y, data, ...) {
  # Quantile normalize
  Y_qn = normalize.quantiles(Y)
  # Fit Y ~ Age + Gender, extra residual (using function created above)
  Y_qn_resid = f_quantile_norm_resid(Y = Y_qn, df = data, ...)
  # Quantile normalize the residual
  Y_qn_resid_qn = data.frame(normalize.quantiles(Y_qn_resid))
  return(Y_qn_resid_qn)
}
```

```
# Create a quantile normalized adjusted Y data frame (i.e., quantile normalization
# and covariate adjustment is performed in one fell swoop)
qn_resid_Y <- f_quantile_normalize_adjust(Y, data = df_i1_regress)
# Create a copy of this data frame for use later in this workflow
qn_resid_Y_b <- qn_resid_Y
# Rename the columns of the quantile normalized data frame to match the
# phenotypes of interest
names(qn_resid_Y) <- all.traits
```

3.4 Remove outliers

Observations in violation of multivariate normality at an alpha=0.01 level based on Mahalanobis distance-based test statistics are now removed to avoid spurious conclusions.

```
# Create a function to calculate Mahalanobis distance
getMD <- function(x) {
  Sx <- cov(x)
  m <- mahalanobis(x, colMeans(x), Sx)
  return(m)
}
```

```
# Drop individuals with data violating multivariate normality at alpha = 0.01
i_keep <- which(pchisq(getMD(qn_resid_Y_b), df = dim(Y)[2]) > 0.01)
```

```
# Record sample sizes in a summary table
table1 <- data.frame(study=rep(NA,1),N.traits=NA,N.total=NA,N.used=NA)
i <- 1
table1[i,"study"] <- "Synthetic Cohort - All Traits"
table1[i,"N.total"] <- nrow(qn_resid_Y)
table1[i,"N.used"]  <- nrow(qn_resid_Y[i_keep, ])
table1[i,"N.traits"] <- ncol(qn_resid_Y)
table1[i,]
```

```
##                           study N.traits N.total N.used
## 1 Synthetic Cohort - All Traits       11    1500   1356
```

```
cat(dim(Y)[1] - length(i_keep), " Obs removed due to violation of MV-Normality")
```

```
## 144  Obs removed due to violation of MV-Normality
```

3.5 Prepare final files for mvBIMBAM

```
# Write phenotypes to a text file for use in mvBIMBAM
if (!dir.exists("./inputs")) {
  dir.create("./inputs")
}
write.table(round(qn_resid_Y[i_keep,], 8),
            "./inputs/synthetic_pheno_bimbam.txt", sep = " ",
            row.names = F, col.names = F)

# Refine genotype data for mvBIMBAM and write file
Geno_write <- df_synth %>% select(rs373863828_C, all_of(all.traits)) %>%
  filter(complete.cases(.)) %>%
  select(rs373863828_C) %>%
  {.[i_keep,]}
Geno_String <- paste0(unlist(c(Geno_write)), collapse = ",")
Geno_String <- paste0("rs373863828,",Geno_String, collapse = "")
Geno_String <- paste0(length(Geno_write), "\n", 1, "\n",Geno_String,"\n")
# Write genotypes (no need to rewrite geno input file)
writeLines(Geno_String, con = "./inputs/synthetic_geno_bimbam.txt", sep = "")
```

4 mvBIMBAM analyses

As described in our paper, the association of rs373863828 with a panel of correlated phenotypes was performed with the Bayesian multivariate mvBIMBAM framework, which we will now apply to the synthetic data set we are working with.

In the mvBIMBAM framework, a global null model representing no association between phenotypes and genotype is compared with an exhaustive combination of alternative models, in which all different combinations of phenotypes are associated with the genotype. For the alternative models, the mvBIMBAM methodology splits phenotypes into all possible partitions of U, D, and I, each representing ‘unassociated’, ‘directly’, and ‘indirectly’ associated.

Call system() to run mvBIMBAM.

```
## [1] "bimbam -g ./inputs/synthetic_geno_bimbam.txt -p ./inputs/synthetic_pheno_bimbam.txt -o bimbam_ou
```

```
## Warning in system(call, intern = TRUE): running command 'bimbam -g ./inputs/
## synthetic_geno_bimbam.txt -p ./inputs/synthetic_pheno_bimbam.txt -o bimbam_out
## -mph 2 -f 11 -A 0.05 -A 0.1 -A 0.2 -A 0.4' had status 1
```

```
## [1] "-bimbam: file 0 has 1356 individual and 1 snps"
## [2] "-bimbam: read file 0 again "
## [3] "-bimbam: number of phenotypes = 11"
## [4] "total = 177147"
## [5] "output/bimbam_out.mph.txt has been created."
## [6] "output/bimbam_out.mph.prob.txt has been created."
## [7] "output/bimbam_out.mph.BFs.txt has been created."
## attr(,"status")
## [1] 1
```

```
bimbam -g ./inputs/synthetic_geno_bimbam.txt -p ./inputs/synthetic_pheno_bimbam.txt
-o bimbam_out -mph 2 -f 11 -A 0.05 -A 0.1 -A 0.2 -A 0.4
```

5 Results

5.1 Bayes factors (BF)

The evidence against the null hypothesis is the sum of Bayes factors (BF) (log10 scale) of all partitions weighted by a diffuse prior.

Note the code here has been annotated, but more detailed documentation can be found in the mvBIMBAM documentation on GitHub.

```
# Read in the mvBIMBAM Bayes Factor output file
s <- readLines("./output/bimbam_out.mph.BFs.txt")
# Clean up file for use
# Note: We can ignore these NA coercion warnings
m1 <- matrix(na.omit(as.numeric(str_split(s, " ")[[1]])), nrow = 1)
colnames(m1) = c("BF", "BF_All_Partition", all.traits)
```

```
# View results
pander(m1, digits = 4, caption = "Bayes Factors: Synthetic Cohort, All Traits")
```

Table 1: Bayes Factors: Synthetic Cohort, All Traits (continued below)

BF	BF_All_Partition	BMI	Height	Weight	FFM	FM	WHR
6.179	6.977	7.444	0.1935	5.491	2.407	5.868	0.1842

Abd_Circ	Hip_Circ	HDL_C	NetTG	Cholesterol
3.449	6.319	-0.4349	-0.4712	-0.09826

A note about interpretation: The above table presents the log10 BF for each trait. Strong evidence of association is defined as log10 BF > 5; suggestive evidence is defined as 1.5 < log10 BF < 5; and negligible evidence is defined as log10 BF < 1.5.

5.2 Bayesian posterior probabilities of association

In addition to log10 BFs, probabilities for no association, direct association, and indirect association are given as output while marginal posterior probabilities of association (MPPA) are calculated by summing the marginal posterior probabilities of direct and indirect association.

```
# Read in the mvBIMBAM probability output file
s <- readLines("./output/bimbam_out.mph.prob.txt")
# Clean up file for use
m2 <- matrix(na.omit(as.numeric(str_split(s, " ")[[1]])), nrow = 2)
m2 <- rbind(m2, 1 - (m2[1,] + m2[2,]))
colnames(m2) <- all.traits
rownames(m2) <- c("Unassociated", "Directly", "Indirectly")
```

```
pander(m2, digits = 4, caption = "Bayesian Probabilities: Synthetic Cohort, All Traits")
```

Table 3: Bayesian Probabilities: Synthetic Cohort, All Traits (continued below)

	BMI	Height	Weight	FFM	FM	WHR
Unassociated	0	0.1235	2e-05	0.00053	1e-05	0.05785
Directly	0.9867	0.6124	0.4322	0.4587	0.3782	0.4657
Indirectly	0.01326	0.2641	0.5678	0.5408	0.6218	0.4765

	Abd_Circ	Hip_Circ	HDL_C	NetTG	Cholesterol
Unassociated	0.00015	1e-05	0.4095	0.2282	0.1994
Directly	0.6292	0.4268	0.4794	0.2897	0.5221
Indirectly	0.3706	0.5732	0.111	0.4821	0.2785

A note about interpretation: The numbers above can be interpreted as the probability of no association, direct association, or indirect association, which together sum to 1 (i.e., 100%).

Both directly and indirectly associated phenotypes are associated with genotype, but indirectly associated phenotypes are conditionally independent of the genotype given the presence of a directly associated phenotype in the model.

For example, in this synthetic data set, there is a suggested <1% probability that there is no association between the variant of interest (rs373863828) and weight, a 43.2% probability that rs373863828 directly impacts weight, and a 56.8% probability that rs373863828 indirectly impacts weight conditional on another phenotype within the dataset. This is supported by a log10 BF>5 (see BF in Table 2 above), which provides suggests strong evidence of association.

5.3 Sample size summary table

Finally, we will create a table summarizing the nunmber of traits examined (N.traits), the total number of participants in the data set (N.total), the number of participants that were included in the analysis

(N.used), the number of participants dropped (N.removed), and the percentage of participants removed (Percent.removed).

```
system("wc inputs/*pheno*.txt", intern = TRUE)
```

```
## [1] "  1356  14916 179358 inputs/synthetic_pheno_bimbam.txt"
```

Table 5: Sample sizes (continued below)

study	N.traits	N.total	N.used	N.removed	Percent.removed
Synthetic Cohort - All Traits	11	1500	1356	144	9.6

6 Save data and results

To conclude, save (1) the quantile normalized adjusted data file and (2) the Bayes factors from the mvBIMBAM results as these data are used in the second markdown that provides example code to construct the Bayesian networks using bnlearn.

6.1 Quantile normalized data file for use in bnlearn

```
# Write data for bnlearn
saveRDS(data.frame(rs373863828 = Geno_write, round(qn_resid_Y[i_keep,], 8)), file = "SyntheticQuantNorm
```

6.2 mvBIMBAM BFs

```
save(m1, m2, table1, file = "mvBimBam_Results.RDdata")
```

7 Session information

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
## BLAS:   /usr/local/lib64/R/lib/libRblas.so
## LAPACK: /usr/local/lib64/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] pander_0.6.4          preprocessCore_1.56.0 forcats_0.5.1
##  [4] stringr_1.4.0         dplyr_1.0.7           purrr_0.3.4
##  [7] readr_2.1.0           tidyr_1.1.4           tibble_3.1.6
## [10] ggplot2_3.3.5         tidyverse_1.3.1       knitr_1.36
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1 xfun_0.28        haven_2.4.3      colorspace_2.0-2
##  [5] vctrs_0.3.8      generics_0.1.1   htmltools_0.5.2  yaml_2.2.1
##  [9] utf8_1.2.2       rlang_0.4.12     pillar_1.6.4     glue_1.5.0
## [13] withr_2.4.2      DBI_1.1.1        dbplyr_2.1.1     modelr_0.1.8
## [17] readxl_1.3.1     lifecycle_1.0.1  munsell_0.5.0    gtable_0.3.0
## [21] cellranger_1.1.0 rvest_1.0.2      evaluate_0.14    tzdb_0.2.0
## [25] fastmap_1.1.0    fansi_0.5.0      broom_0.7.10     Rcpp_1.0.7
## [29] scales_1.1.1     backports_1.3.0  jsonlite_1.7.2   fs_1.5.0
## [33] hms_1.1.1        digest_0.6.28    stringi_1.7.5    grid_4.1.2
## [37] cli_3.1.0        tools_4.1.2      magrittr_2.0.1   crayon_1.4.2
## [41] pkgconfig_2.0.3  ellipsis_0.3.2   xml2_1.3.2       reprex_2.0.1
## [45] lubridate_1.8.0  rstudioapi_0.13  assertthat_0.2.1 rmarkdown_2.11
## [49] httr_1.4.2       R6_2.5.1         compiler_4.1.2
```