Genetic Analysis incorporating Pleiotropy and Annotation
with ‘GPA’ Package

Dongjun Chung 1, Can Yang 2, Cong Li 3, Joel Gelernter 4,5,6,7, and Hongyu Zhao 3,6,8,9
1Department of Public Health Sciences, Medical University of South Carolina,
Charleston, SC, USA.
2 Department of Mathematics, Hong Kong Baptist University,
Hong Kong.
3 Program in Computational Biology and Bioinformatics, Yale University,
New Haven, CT, USA.
4 Department of Psychiatry, Yale School of Medicine,
New Haven, CT, USA.
5 VA CT Healthcare Center, West Haven, CT, USA.
6 Department of Genetics, Yale School of Medicine, West Haven, CT, USA.
7 Department of Neurobiology, Yale School of Medicine, New Haven, CT, USA.
8 Department of Biostatistics, Yale School of Public Health,
New Haven, CT, USA.
9 VA Cooperative Studies Program Coordinating Center, West Haven, CT, USA.

October 24, 2023

Contents

1 Installation 1
2 Overview 2
3 Workflow 2
   3.1 Fitting the GPA Model 3
   3.2 Association Mapping 5
4 Hypothesis Testing for Pleiotropy and Annotation Enrichment 7
5 Advanced Use 8
6 Session Info 10

1 Installation

> if(!requireNamespace("BiocManager", quietly = TRUE))
  + install.packages("BiocManager")
> BiocManager::install("GPA")
2 Overview

This vignette provides an introduction to the genetic analysis using the ‘GPA’ package. R package ‘GPA’ implements GPA (Genetic analysis incorporating Pleiotropy and Annotation), a flexible statistical framework for the joint analysis of multiple genome-wide association studies (GWAS) and its integration with various genetic and genomic data. It implements a flexible parametric mixture modeling approach for such integrative analysis and also provides hypothesis testing procedures for pleiotropy and annotation enrichment.

The package can be loaded with the command:

\[
\text{R} > \text{library("GPA")}
\]

This vignette is organized as follows. Section 3.1 discusses how to fit GPA models in various settings. Section 3.2 explains command lines for association mapping using GPA. Section 4 discusses steps of the hypothesis testing for pleiotropy and annotation enrichment. Finally, Section 5 discusses some methods useful for more advanced users.

We encourage questions or requests regarding ‘GPA’ package to be posted on our Google group https://groups.google.com/d/forum/gpa-user-group. Users can find the most up-to-date versions of ‘GPA’ package in our GitHub webpage (http://dongjunchung.github.io/GPA/).

3 Workflow

[Note] All the results below are based on the 100 EM iterations for quick testing and building of the R package. These results are provided here only for the illustration purpose and should not be considered as real results. We recommend users to use sufficient number of EM iterations for the real data analysis, as we use 10,000 EM iterations for all the results in our manuscript [1].

In this vignette, we use the GWAS data of five psychiatric disorders [3, 4], where traits include attention deficit-hyperactivity disorder (ADHD), autism spectrum disorder (ASD), bipolar disorder (BPD), major depressive disorder (MDD), and schizophrenia (SCZ). We downloaded summary statistics of the five psychiatric disorders from the section for cross-disorder analysis at the Psychiatric Genomics Consortium (PGC) website and took the intersection of their SNPs, resulting in a \( p \)-value matrix of \( 1,219,805 \times 5 \). We also consider the binary annotation data using genes preferentially expressed in the central nervous system (CNS) as an annotation data [5, 6]. We generated an annotation matrix of size \( 1,219,805 \times 1 \), where the entries corresponding to SNPs within 50-kb of the genes from the CNS set were set to be one and zero otherwise. ‘gpaExample’ package provides this example dataset.

\[
\text{R} > \text{library(gpaExample)}
\]

\[
\text{R} > \text{data(exampleData)}
\]

\[
\text{R} > \text{dim(exampleData$pval)}
\]

\[
[1] \quad 1219805 \quad 5
\]

\[
\text{R} > \text{head(exampleData$pval)}
\]
ADHD    ASD    BPD    MDD    SCZ
[1,] 0.4452 0.2847 0.4362 0.2527 0.7716
[2,] 0.4592 0.1030 0.7423 0.3543 0.6478
[3,] 0.8735 0.0687 0.7020 0.3395 0.6296
[4,] 0.7395 0.4837 0.4929 0.0236 0.7704
[5,] 0.5756 0.6122 0.5755 0.0125 0.7167
[6,] 0.4997 0.1784 0.4844 0.1611 0.7810

R> dim(exampleData$ann)

[1] 1219805   1

R> head(exampleData$ann)

   V1
[1,] 0
[2,] 1
[3,] 1
[4,] 1
[5,] 1
[6,] 1

R> table(exampleData$ann)

   0  1
952636 267169

3.1 Fitting the GPA Model

We are now ready to fit a GPA model using the GWAS p-value data described above (exampleData$pval). R package GPA provides flexible analysis framework and automatically adjusts its model structure based on the provided data. However, we note that although, in principle, any number of GWAS data can be analyzed in the GPA model, R package GPA has been investigated and tested most extensively for the case of two GWAS data. Hence, if users have more than two GWAS data of interest, we recommend users to analyze each pair of GWAS data at a time. Based on this rationale, in this vignette, we focus on the joint analysis of BPD and SCZ, which correspond to the third and fifth columns of exampleData$pval.

First, assuming that there is no annotation data, we fit the GPA model with the command:

R> fit.GPA.noAnn <- GPA( exampleData$pval[, c(3,5) ], NULL )

or equivalently (which is actually simpler command),

R> fit.GPA.noAnn <- GPA( exampleData$pval[, c(3,5) ] )

When we also have related annotation data, this annotation data can be easily incorporated into the GPA model by providing it in the second argument of ‘GPA’ method. Note that ‘GPA’ method expects that the number of rows of data in the first and second arguments are same and also the elements of data in the second argument are either one (annotated) or zero (otherwise).

R> fit.GPA.wAnn <- GPA( exampleData$pval[, c(3,5) ], exampleData$ann )
The following command prints out a summary of GPA model fit, including data summary, model setting, parameter estimates, and their standard errors.

R> fit.GPA.wAnn

Summary: GPA model fitting results (class: GPA)
--------------------------------------------------
Data summary:
   Number of GWAS data: 2
   Number of SNPs: 1219805
   Number of annotation data: 1
Model setting:
   Theoretical null distribution is assumed.
Parameter estimates (standard errors):
   alpha: 0.595 0.544
           ( 0.007 0.004 )
   GWAS combination: 00 10 01 11
   pi: 0.803 0.047 0.092 0.059
        ( 0.003 0.003 0.003 0.003 )
   q:
      Annotation #1:
           0.205 0.246 0.241 0.36
        ( 0.001 0.019 0.01 0.011 )
Ratio of q over baseline (00):
   GWAS combination: 10 01 11
      Annotation #1:
           1.202 1.177 1.757
        ( 0.102 0.056 0.044 )
--------------------------------------------------

Parameter estimates and their standard errors can be extracted using methods ‘estimates’ and ‘se’, respectively.

R> estimates( fit.GPA.wAnn )

$pis
   00    10    01    11
0.80281274 0.04668881 0.09175225 0.05874620

$betaAlpha
[1] 0.5952023 0.5442478

$q1
   00    10    01    11
V1 0.2046673 0.2460793 0.2409415 0.3595191

$q1ratio
 [,1] [,2] [,3]
[1,] 1.202338 1.177235 1.756603
$betaAlpha
alpha_1 alpha_2
0.006610053 0.003777337

$pis
pi_00 pi_10 pi_01 pi_11
0.003108809 0.002824855 0.003133099 0.003046711

$q1
q1_1_1 q1_1_2 q1_1_3 q1_1_4
[1,] 0.001485032 0.01946034 0.01010343 0.01057017

$q1ratio
[,1] [,2] [,3]
[1,] 0.1024952 0.05632306 0.04425038

3.2 Association Mapping

Now, based on the fitted GPA model, we implement association mapping with the command:

\[
R > \text{assoc.GPA.wAnn} \leftarrow \text{assoc( fit.GPA.wAnn, FDR=0.20, fdrControl="global" )}
\]

\[
R > \text{dim(assoc.GPA.wAnn)}
\]

\[
[1] 1219805 2
\]

\[
R > \text{head(assoc.GPA.wAnn)}
\]

\[
[,1] [,2]
[1,] 0 0
[2,] 0 0
[3,] 0 0
[4,] 0 0
[5,] 0 0
[6,] 0 0
\]

\[
R > \text{table(assoc.GPA.wAnn[,1])}
\]

\[
0 1
1218460 1345
\]

\[
R > \text{table(assoc.GPA.wAnn[,2])}
\]

\[
0 1
1213496 6309
\]

‘assoc’ method returns a binary matrix indicating association of each SNP, where one indicates that a SNP is associated with the phenotype and zero otherwise. Its rows and columns match those of input p-value matrix for ‘GPA’ method. ‘assoc’ method allows both local (‘fdrControl="local"’
and global FDR controls (‘fdrControl=”global”’) and users can control FDR level using the argument ‘FDR’. Hence, the association mapping results above indicate that there are 1,345 and 6,309 SNPs associated with each of BPD and SCZ, respectively, under the global FDR control at 0.20 level.

‘fdr’ method for the output of ‘GPA’ method (‘fit.GPA.wAnn’ in this example) further provides the matrix of local FDR that a SNP is not associated with each phenotype, where its rows and columns match those of input p-value matrix for ‘GPA’ method. This method will be useful when users want to scrutinize association of each SNP more closely.

```
R> fdr.GPA.wAnn <- fdr(fit.GPA.wAnn)
R> dim(fdr.GPA.wAnn)
[1] 1219805  2
R> head(fdr.GPA.wAnn)
   BPD     SCZ
[1,] 0.9337519 0.9155452
[2,] 0.9129826 0.8776131
[3,] 0.9106763 0.8753579
[4,] 0.9021052 0.8796740
[5,] 0.9062500 0.8786411
[6,] 0.9017298 0.8800522
```

When users are interested in the association of a SNP for certain combination of phenotypes, users can specify it using ‘pattern’ argument in both ‘assoc’ and ‘fdr’ methods. Specifically, users can specify the pattern using 1 and *, where 1 and * indicate phenotypes of interest and phenotypes that are not of interest, respectively. For example, when there are three phenotypes, ‘pattern=”111”’ means a SNP associated with all of three phenotypes, while ‘pattern=”11*”’ means a SNP associated with the first two phenotypes (i.e., association with the third phenotype is ignored (averaged out)). If a pattern is specified, ‘assoc’ and ‘fdr’ methods return a corresponding vector instead of a matrix. The association mapping results below indicate that there are 478 SNPs associated with both BPD and SCZ under the global FDR control at 0.20 level.

```
R> assoc11.GPA.wAnn <- assoc( fit.GPA.wAnn, FDR=0.20, fdrControl="global", pattern="11" )
R> length(assoc11.GPA.wAnn)
[1] 1219805
R> head(assoc11.GPA.wAnn)
[1] 0 0 0 0 0 0
R> table(assoc11.GPA.wAnn)
assoc11.GPA.wAnn
  0  1
1219327 478
```

```
R> fdr11.GPA.wAnn <- fdr( fit.GPA.wAnn, pattern="11" )
R> length(fdr11.GPA.wAnn)
```
R> head(fdr11.GPA.wAnn)
[1] 0.9739450 0.9524217 0.9508731 0.9483990 0.9498141 0.9483539

4 Hypothesis Testing for Pleiotropy and Annotation Enrichment

In the joint analysis of multiple GWAS data, it is of interest to investigate whether there is pleiotropy, i.e., the signals from the two GWAS are related. We developed a hypothesis testing procedure to investigate pleiotropy and implemented it as ‘pTest’ method. Because this hypothesis testing procedure is based on the likelihood ratio test (LRT), we also need a GPA model fit under the null hypothesis of pleiotropy, i.e., the signals from the two GWAS are independent of each other. Users can easily fit the GPA model under the null hypothesis of pleiotropy by setting `pleiotropyH0=TRUE` when running ‘GPA’ method:

R> fit.GPA.pleiotropy.H0 <- GPA( exampleData$pval[, c(3,5)], NULL, pleiotropyH0=TRUE )
R> fit.GPA.pleiotropy.H0

Summary: GPA model fitting results (class: GPA)
--------------------------------------------------
Data summary:
Number of GWAS data: 2
Number of SNPs: 1219805
Number of annotation data: (not provided)
Model setting:
Theoretical null distribution is assumed.
GPA is fitted under H0 of pleiotropy LRT.
Parameter estimates (standard errors):
alpha: 0.589 0.544
( 0.007 0.004 )
GWAS combination: 00 10 01 11
pi: 0.77 0.082 0.134 0.014
( 0.004 0.003 0.003 0.002 )
--------------------------------------------------

Now, based on these GPA model, we can implement the hypothesis testing for pleiotropy with the command:

R> test.GPA.pleiotropy <- pTest( fit.GPA.noAnn, fit.GPA.pleiotropy.H0 )

Hypothesis testing for pleiotropy
---------------------------------
GWAS combination: 00 10 01 11
pi: 0.802 0.048 0.092 0.058
( 0.003 0.003 0.003 0.003 )
test statistics: 1573.934
p-value: 0
---------------------------------
The hypothesis testing results indicate that there is strong evidence for pleiotropy between BPD and SCZ.

When annotation data is also available, we can further investigate whether there is statistical evidence for enrichment of GWAS signals in this annotation data. Again, this hypothesis testing procedure is based on LRT and we need to fit a GPA model under the null hypothesis of annotation enrichment, i.e., GWAS signals are not enriched in the annotation data. This null model can easily be obtained by fitting the GPA model without annotation data, which corresponds to the `fit.GPA.noAnn` object we already obtained above. Now, we can implement the hypothesis testing for annotation enrichment using `aTest` method:

```R
R> test.GPA.annotation <- aTest( fit.GPA.noAnn, fit.GPA.wAnn )
```

Hypothesis testing for annotation enrichment
( Note: This version of test is designed for single annotation data )
--------------------------------------------------
q:
GWAS combination: 00 10 01 11
Annotation # 1 :
   0.205 0.246 0.241 0.36
   ( 0.001 0.019 0.01 0.011 )
Ratio of q over baseline ( 00 ):
GWAS combination: 10 01 11
Annotation # 1 :
   1.202 1.177 1.757
   ( 0.102 0.056 0.044 )
test statistics: 613.576
p-value: 1.148541e-132
--------------------------------------------------

The hypothesis testing results indicate that there is strong evidence for enrichment of GWAS signals in our CNS gene annotation data. Currently, `aTest` method works only for one annotation data but we are now working on relaxing this limitation.

5 Advanced Use

Methods `print` and `cov` might be useful for more advanced users. `print` method provides the matrix of posterior probability that a SNP belongs to each combination of association status and this method will be useful when users want to scrutinize the joint analysis results more closely. `cov` method provides the covariance matrix of GPA model and this can be useful, for example, in the case that users want to calculate the standard error for certain transformation of parameter estimates using Delta method.

```R
R> dim(print(fit.GPA.wAnn))
[1] 1219805 4
R> head(print(fit.GPA.wAnn))
```
```
00  10  01  11
[1,] 0.8753521  0.04019309  0.05839976  0.02605503
[2,] 0.8381740  0.03943915  0.07480858  0.04757829
[3,] 0.8351611  0.04019677  0.07551521  0.04912690
[4,] 0.8333802  0.04629380  0.06872496  0.05160100
[5,] 0.8350770  0.04356413  0.07117297  0.05018589
[6,] 0.834281  0.04662408  0.06830171  0.05164613
```

R> cov(fit.GPA.wAnn)

<table>
<thead>
<tr>
<th></th>
<th>pi_10</th>
<th>pi_01</th>
<th>pi_11</th>
<th>alpha_1</th>
<th>alpha_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>pi_10</td>
<td>7.979806e-06</td>
<td>-6.631431e-08</td>
<td>-2.849873e-06</td>
<td>7.708394e-06</td>
<td>-4.253844e-06</td>
</tr>
<tr>
<td>pi_01</td>
<td>-6.631431e-08</td>
<td>9.816308e-06</td>
<td>-5.790746e-06</td>
<td>-1.183169e-05</td>
<td>4.363675e-06</td>
</tr>
<tr>
<td>pi_11</td>
<td>-2.849873e-06</td>
<td>-5.790746e-06</td>
<td>9.282449e-06</td>
<td>1.235563e-05</td>
<td>4.630787e-06</td>
</tr>
<tr>
<td>alpha_1</td>
<td>7.708394e-06</td>
<td>-1.183169e-05</td>
<td>1.235563e-05</td>
<td>4.369280e-05</td>
<td>1.837662e-07</td>
</tr>
<tr>
<td>alpha_2</td>
<td>-4.253844e-06</td>
<td>4.363675e-06</td>
<td>4.630787e-06</td>
<td>1.837662e-07</td>
<td>1.426288e-05</td>
</tr>
<tr>
<td>q1_1_1</td>
<td>-2.094119e-07</td>
<td>-8.048740e-08</td>
<td>-3.649913e-07</td>
<td>-5.585724e-07</td>
<td>-1.234925e-07</td>
</tr>
<tr>
<td>q1_1_2</td>
<td>9.950054e-07</td>
<td>-1.510694e-06</td>
<td>3.052268e-06</td>
<td>1.732083e-06</td>
<td>-6.280012e-06</td>
</tr>
<tr>
<td>q1_1_3</td>
<td>-1.232532e-06</td>
<td>2.961907e-06</td>
<td>-1.266104e-06</td>
<td>-1.150016e-05</td>
<td>-8.885295e-07</td>
</tr>
<tr>
<td>q1_1_4</td>
<td>3.911655e-06</td>
<td>4.117694e-06</td>
<td>-9.246730e-06</td>
<td>-2.684745e-06</td>
<td>-1.919373e-06</td>
</tr>
</tbody>
</table>

References


6 Session Info

R> sessionInfo()

R version 4.3.1 (2023-06-16)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 22.04.3 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.18-bioc/R/lib/Rblas.so
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0

locale:
[1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
[3] LC_TIME=en_GB
[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

time zone: America/New_York
tzcode source: system (glibc)

attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] gpaExample_1.13.0 GPA_1.14.0 Rcpp_1.0.11

loaded via a namespace (and not attached):
[1] Matrix_1.6-1.1 gtable_0.3.4 dplyr_1.1.3 vegan_2.6-4
[5] compiler_4.3.1 promises_1.2.1 tidyselect_1.2.0 parallel_4.3.1
[9] later_1.3.1 cluster_2.1.4 splines_4.3.1 scales_1.2.1
[13] fastmap_1.1.1 mime_0.12 lattice_0.22-5 plyr_1.8.9
[17] ggplot2_3.4.4 R_6_2.5.1 generics_0.1.3 MASS_7.3-60
[21] ggrepel_0.9.4 tibble_3.2.1 munsell_0.5.0 shiny_1.7.5
[25] pillar_1.9.0 rlang_1.1.1 utf8_1.2.4 httpuv_1.6.12
[29] cli_3.6.1 magrittr_2.0.3 mgcv_1.9-0 shinyBS_0.61.1
[33] digest_0.6.33 grid_4.3.1 xtable_1.8-4 permute_0.9-7
[37] lifecycle_1.0.3 nlme_3.1-163 vctrs_0.6.4 glue_1.6.2
[41] fansi_1.0.5 colorspace_2.1-0 pkgconfig_2.0.3 tools_4.3.1
[45] ellipsis_0.3.2 htmltools_0.5.6.1