

Statistical Approach for Pleiotropy Informed and Functional Annotation Tree Guided Prioritization of GWAS Results with ‘*multiGPATree*’ Package

Aastha Khatiwada¹ Ayse Selen Yilmaz² Bethany J. Wolf³ Maciej Pietrzak²
Dongjun Chung^{2,4}

¹Department of Biostatistics and Bioinformatics, National Jewish Health, Denver, CO

²Department of Biomedical Informatics, The Ohio State University, Columbus, OH

³Department of Public Health Sciences, Medical University of South Carolina, Charleston, SC

⁴Pelotonia Institute for Immuno-Oncology, The James Comprehensive Cancer Center, The Ohio State University, Columbus, OH

1 Overview

This vignette provides an introduction to the *multiGPATree* package. R package *multiGPATree* implements the *multiGPATree* method, a novel statistical approach to prioritizing risk-associated SNPs and the combinations of functional annotations related to one or more trait risk-associated SNPs. The *multiGPATree* approach employs a hierarchical model to integrate GWAS summary statistics for multiple traits and functional annotation information within a unified framework by combining an iterative procedure (EM algorithm (Dempster, Laird, and Rubin 1977)) and a multivariate decision tree algorithm (MVPART (De’Ath 2002)).

The package can be loaded with the command:

```
> # install.packages("devtools")
> # library(devtools)
> # devtools::install_github("cran/mvpart")
> # library(mvpart)
> # devtools::install_github("asthakhatiwada/multiGPATree")
> library(multiGPATree)
```

This vignette is organized as follows. Section 2 discusses the data structure required to implement the *multiGPATree* method and section 3 describes the workflow to implement the *multiGPATree* method. Section 3.1 discusses how to fit a *multiGPATree* model, section 3.2 describes how to prune a large *multiGPATree* model, section 3.3 describes the *multiGPATree* model plot and functional annotation tree, and finally, section 3.4 explain command lines for association mapping and identification of combination of functional annotations for one or more traits using *multiGPATree*.

Please feel free to contact Dongjun Chung at chung.911@osc.edu or Aastha Khatiwada at khatiwadaa@njhealth.org for any questions or suggestions regarding the *multiGPATree* package.

2 Data structure

In this vignette, we use the simulated data (*simdata*) provided in the package to fit the *multiGPATree* model for post-GWAS analysis. The simulated data includes information for two traits/phenotypes (P1 and P2). In the simulated data, the number of SNPs and the number of functional annotations are set to $M = 1000$ and $K = 10$, respectively. The GWAS association p -values for the SNPs are stored in *simdata\$gwasPval* and the binary functional annotation information for the SNPs are stored in *simdata\$annMat*, respectively. The number of rows in *simdata\$gwasPval* is assumed to be the same as the number of rows in *simdata\$annMat* where the i -th ($i = 1, \dots, M$) row of *gwasPval* and *annMat* correspond to the same SNPs.

```
> data("simdata")
> class(simdata)
[1] "list"
> names(simdata)
[1] "annMat" "gwasPval"
> dim(simdata$gwasPval)
[1] 1000    2
> dim(simdata$annMat)
[1] 1000   10
> head(simdata$gwasPval)
      P1      P2
SNP_1 0.98281150 0.42126064
SNP_2 0.52555968 0.59463902
SNP_3 0.07748063 0.91700262
SNP_4 0.63208756 0.15856348
SNP_5 0.28658882 0.11983350
SNP_6 0.96132083 0.06235469
> head(simdata$annMat)
      A1 A2 A3 A4 A5 A6 A7 A8 A9 A10
SNP_1  1  0  0  0  0  0  0  0  0  0  1
SNP_2  1  0  0  0  0  0  0  0  0  1  0
SNP_3  1  0  0  0  0  0  0  0  0  0  0
SNP_4  1  0  0  0  0  0  0  0  0  1  1
SNP_5  1  0  0  0  0  0  0  0  0  0  0
SNP_6  1  0  0  0  0  0  0  0  0  0  0
```

3 Workflow

3.1 Fitting the multiGPATree Model

We are now ready to fit a *multiGPATree* model using the GWAS p -value and functional annotation data described above. We can fit the *multiGPATree* model with the command:

```
> fit.mGPATree <- multiGPATree(gwasPval = simdata$gwasPval,
+                               annMat = simdata$annMat,
+                               initAlpha = 0.1,
+                               cpTry = 0.005)
> fit.mGPATree
Summary: multiGPATree model results (class: multiGPATree)
```

```

-----
Data summary:
  Number of GWAS data: 2
  Number of Annotations: 10
  Number of SNPs: 1000
  Alpha estimates: 0.3691, 0.434
Functional annotation tree description:
  local FDR P1 local FDR P2 A1 A5 A4 A3 A6 A2
LEAF 1      0.9568      0.95756 0 0 0 - - -
LEAF 2      0.9889      0.25543 0 0 1 1 - -
LEAF 3      0.9627      0.78918 0 0 1 0 - -
LEAF 4      0.8816      0.68446 0 1 - - 0 -
LEAF 5      0.1454      0.07836 0 1 - - 1 -
LEAF 6      0.0690      0.95888 1 - - - - 1
LEAF 7      0.7743      0.87926 1 - - - - 0
-----

```

3.2 Prunning the multiGPATree model

The `prune()` function will prune the *multiGPATree* model using any `cp` value between 0 and 1 as shown below.

```

> fit.mGPATree.pruned <- prune(fit.mGPATree, cp = 0.20)
> fit.mGPATree.pruned
Summary: multiGPATree model results (class: multiGPATree)
-----
Data summary:
  Number of GWAS data: 2
  Number of Annotations: 10
  Number of SNPs: 1000
  Alpha estimates: 0.3691, 0.434
Functional annotation tree description:
  local FDR P1 local FDR P2 A1 A5
LEAF 1      0.9608      0.8643 0 0
LEAF 2      0.5135      0.3814 0 1
LEAF 3      0.4216      0.9191 1 -
-----

```

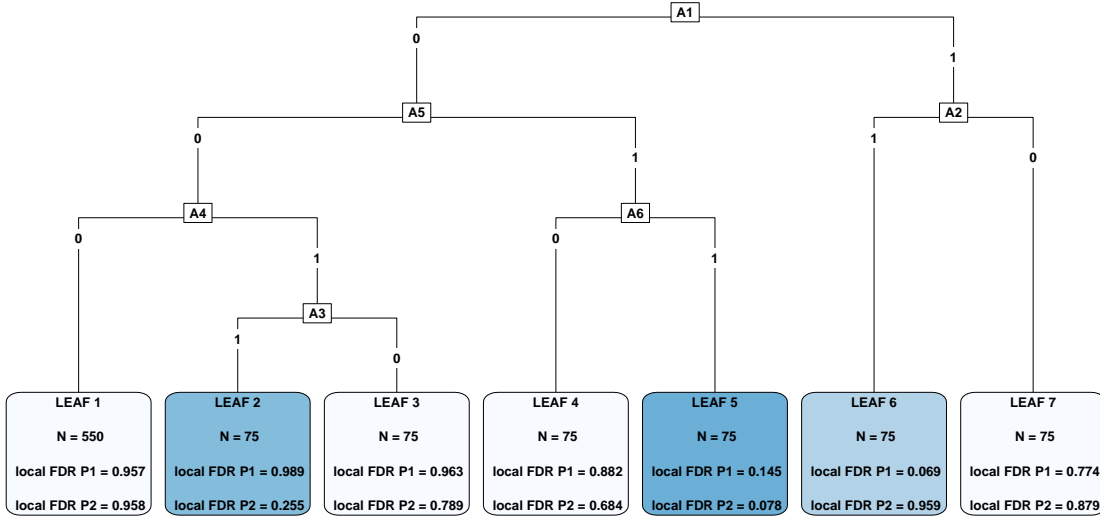
3.3 Functional annotation tree

The `plot()` and `leaf()` functions will plot the *multiGPATree* functional annotation tree and provide information about the leaves (terminal nodes) in the tree as shown below.

```

> plot(fit.mGPATree)

```



```
> leaf(fit.mGPATree)
      local FDR P1 local FDR P2 A1 A5 A4 A3 A6 A2
LEAF 1      0.9568      0.95756  0  0  0 - - -
LEAF 2      0.9889      0.25543  0  0  1  1 - -
LEAF 3      0.9627      0.78918  0  0  1  0 - -
LEAF 4      0.8816      0.68446  0  1  - -  0 -
LEAF 5      0.1454      0.07836  0  1  - -  1 -
LEAF 6      0.0690      0.95888  1  -  - - -  1
LEAF 7      0.7743      0.87926  1  -  - - -  0
```

3.4 Association mapping

For the fitted *multiGPATree* model, we can make inferences about SNPs using the `assoc()` function by: (1) prioritizing risk-associated SNPs for one or more traits, and (2) identifying the leaves of the *multiGPATree* model in which one or more trait risk-associated SNPs are located. In this case, the `assoc()` function returns four columns. The first column contains binary values where 1 indicates that the SNP is marginally associated with the first trait (P1) and 0 indicates otherwise. Similarly, the second column contains binary values where 1 indicates that the SNP is marginally associated with the second trait (P2) and 0 indicates otherwise. The third column contains binary values where 1 indicates that the SNP is jointly associated with both traits (P1 and P2) and 0 indicates otherwise. Finally, the fourth column provides information regarding the leaf in which a SNP is located in the *multiGPATree* plot. The `assoc()` function allows both local (`fdrControl="local"`) and global FDR controls (`fdrControl="global"`) and users can set the threshold to be between 0 and 1 using the 'FDR' argument.

For the simulated data, *multiGPATree* model identified 4 SNPs to be jointly associated with traits P1 and P2 at the nominal global FDR level of 0.01, all of which are located in leaf 5 which can be interpreted as the 4 risk SNPs being simultaneously annotated for both annotations A5 and A6. Similarly, *multiGPATree* model identified 38 SNPs to be marginally associated with trait P1 at the nominal global FDR level of 0.01. Of the 38 SNPs, 27 SNPs are located in leaf 6 and are simultaneously annotated for annotations A1 and A2 and 11 SNPs are located in leaf 5 and are simultaneously annotated for A5 and A6. The following lines of code can be used to investigate association mapping and functional annotation tree for *multiGPATree* models.

```
> assoc.mGPATree <- assoc(fit.mGPATree,
+                          FDR = 0.01,
```

```

+                                     fdrControl="global")
> head(assoc.mGPATree)
      P1 P2 P1_P2  leaf
SNP_1  0  0    0 LEAF 7
SNP_2  0  0    0 LEAF 7
SNP_3  0  0    0 LEAF 7
SNP_4  0  0    0 LEAF 7
SNP_5  0  0    0 LEAF 7
SNP_6  0  0    0 LEAF 7
> table(assoc.mGPATree$P1_P2)

 0  1
996 4
> table(assoc.mGPATree$P1_P2, assoc.mGPATree$leaf)

      LEAF 1 LEAF 2 LEAF 3 LEAF 4 LEAF 5 LEAF 6 LEAF 7
0      550    75    75    75    71    75    75
1         0     0     0     0     4     0     0
> table(assoc.mGPATree$P1)

 0  1
962 38
> table(assoc.mGPATree$P1, assoc.mGPATree$leaf)

      LEAF 1 LEAF 2 LEAF 3 LEAF 4 LEAF 5 LEAF 6 LEAF 7
0      550    75    75    75    64    48    75
1         0     0     0     0    11    27     0
> table(assoc.mGPATree$P2)

 0  1
977 23
> table(assoc.mGPATree$P2, assoc.mGPATree$leaf)

      LEAF 1 LEAF 2 LEAF 3 LEAF 4 LEAF 5 LEAF 6 LEAF 7
0      550    70    75    75    57    75    75
1         0     5     0     0    18     0     0
>
> table(assoc.mGPATree$P1, assoc.mGPATree$P2)

 0  1
0 944 18
1  33  5
> table(assoc.mGPATree$P1, assoc.mGPATree$P2, assoc.mGPATree$P1_P2)
, ,  = 0

      0  1
0 944 18
1  33  1

, ,  = 1

```

	0	1
0	0	0
1	0	4

References

- De'Ath, Glenn. 2002. "Multivariate Regression Trees: A New Technique for Modeling Species–Environment Relationships." *Ecology* 83 (4): 1105–17.
- Dempster, Arthur P, Nan M Laird, and Donald B Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1): 1–22.