Prioritizing GWAS Results and Identifying Risk SNP-Associated Functional Annotation Tree with '**GPATree**' Package

Aastha Khatiwada[1], Bethany J. Wolf[1], Ayse Selen Yilmaz[2], Paula S. Ramos[1,3], Maciej Pietrzak[2], Andrew Lawson[1], Kelly J. Hunt[1], Hang J. Kim[4], Dongjun Chung[2]

[1]Department of Public Health Sciences, Medical University of South Carolina, Charleston, South Carolina, USA
[2]Department of Biomedical Informatics, The Ohio State University, Columbus, Ohio, USA
[3]Department of Medicine, Medical University of South Carolina, Charleston, South Carolina, USA
[4]Department of Mathematical Sciences, University of Cincinnati, Cincinnati, Ohio, USA

02/17/2021

1 Overview

This vignette provides an introduction to the `GPATree` package. R package `GPATree` implements GPA-Tree, a novel statistical approach to prioritize genome-wide association studies (GWAS) results while simultaneously identifying the combinations of functional annotations associated with risk-associated genetic variants. GPA-Tree integrates GWAS summary statistics and functional annotation data within a unified framework, by combining a decision tree algorithm (CART)(Leo et al. 1984) within the hierarchical model.

The package can be loaded with the command:

```
> library(GPATree)
```

This vignette is organized as follows. Sections 2.1 and 2.2 illustrate the recommended `GPATree-ShinyGPATree` workflow, which provides convenient and interactive genetic data analysis interface. Advanced users might also find Sections 2.3.1 – 2.3.3 useful and these command lines can be used for integrating GPA-Tree as part of the more comprehensive genetic data analysis workflow, for example.

Please feel free to contact Dongjun Chung at `chung.911@osc.edu` for any questions or suggestions regarding the '`GPATree`' package.

2 Workflow

In this vignette, we illustrate the GPA-Tree analysis workflow, using the simulated data provided as the `GPATreeExampleData` in the `GPATree` package. In the simulated data, the number of SNPs is set to

$M = 10,000$ and the number of functional annotations is set to $K = 10$. The GWAS association p-values and the binary functional annotation information are stored in `GPATreeExampleData$gwasPval` and `GPATreeExampleData$annMat`, respectively. The number of rows in `GPATreeExampleData$gwasPval` is assumed to be the same as the number of rows in `GPATreeExampleData$annMat`, where the i-th $(i = 1, ..., M)$ row of `gwasPval` and `annMat` correspond to the same SNP.

```
> data(GPATreeExampleData)
> dim(GPATreeExampleData$gwasPval)
[1] 10000     1
> head(GPATreeExampleData$gwasPval)
          P1
SNP_1 0.7454
SNP_2 0.4894
SNP_3 0.6026
SNP_4 0.1496
SNP_5 0.2538
SNP_6 0.3161
> dim(GPATreeExampleData$annMat)
[1] 10000    10
> head(GPATreeExampleData$annMat)
      A1 A2 A3 A4 A5 A6 A7 A8 A9 A10
SNP_1  1  0  0  0  0  1  0  0  0   1
SNP_2  1  0  0  0  0  0  0  0  0   0
SNP_3  1  0  0  0  0  0  0  0  0   1
SNP_4  1  0  0  0  0  0  0  0  0   0
SNP_5  1  0  0  0  1  1  0  0  0   0
SNP_6  1  0  0  0  0  1  0  0  0   0
```

2.1 Fitting the GPA-Tree Model

We can fit the GPA-Tree model using the GWAS association p-values (`GPATreeExampleData$gwasPval`) and functional annotation data (`GPATreeExampleData$annMat`) described above, using the code shown below.

```
> fit.GPATree <- GPATree(gwasPval = GPATreeExampleData$gwasPval,
+                        annMat = GPATreeExampleData$annMat,
+                        initAlpha = 0.1,
+                        cpTry = 0.005)
```

```
> fit.GPATree
Summary: GPATree model results (class: GPATree)
-------------------------------------------------
Data summary:
    Number of GWAS data: 1
    Number of Annotations: 10
    Number of SNPs: 10000
    Alpha estimate: 0.4999
Functiona annotation tree description:
       local FDR A4 A2 A1 A3
LEAF 1    0.9849  0  0  -  -
LEAF 2    0.9834  0  1  0  -
LEAF 3    0.0203  0  1  1  -
LEAF 4    0.9850  1  -  -  0
```

```
LEAF 5    0.0154  1  -  -  1
-------------------------------------------------
```

2.2 ShinyGPATree

The following command can be used to initialize the ShinyGPATree app. ShinyGPATree allows for interactive and dynamic investigation of disease-risk-associated SNPs and functional annotation trees using R Shiny.

```
> ShinyGPATree(fit.GPATree)
```

2.3 Advanced use

2.3.1 Prunning GPA-Tree model fit

The following command will prune the GPA-Tree model using any cp value between 0 and 1.

```
> fit.GPATree.pruned <- prune(fit.GPATree, cp)
```

2.3.2 Functional annotation tree

The following command will plot the GPA-Tree functional annotation tree and provide information about the leaves (terminal nodes) in the tree.

```
> plot(fit.GPATree)
```

```
> leaf(fit.GPATree)
       local FDR A4 A2 A1 A3
LEAF 1    0.9849  0  0  -  -
LEAF 2    0.9834  0  1  0  -
LEAF 3    0.0203  0  1  1  -
LEAF 4    0.9850  1  -  -  0
LEAF 5    0.0154  1  -  -  1
```

2.3.3 Association mapping

Based on the fitted GPA-Tree model, we can make statistical inference about SNPs by identifying: (1) risk-associated SNPs and (2) the leaves of the GPA-Tree model in which the risk-associated SNP are located in using the code below.

```
> assoc.SNP.GPATree <- assoc(fit.GPATree,
+                            FDR = 0.05,
+                            fdrControl="global")
> head(assoc.SNP.GPATree)
      P1   leaf
SNP_1  0 LEAF 1
SNP_2  0 LEAF 1
SNP_3  0 LEAF 1
SNP_4  0 LEAF 1
SNP_5  0 LEAF 1
SNP_6  0 LEAF 1
> table(assoc.SNP.GPATree$P1)

   0    1
```

```
8542 1458
> table(assoc.SNP.GPATree$leaf)

LEAF 1 LEAF 2 LEAF 3 LEAF 4 LEAF 5
  7198    700    701    700    701
> table(assoc.SNP.GPATree$P1, assoc.SNP.GPATree$leaf)

    LEAF 1 LEAF 2 LEAF 3 LEAF 4 LEAF 5
  0   7154    695      0    693      0
  1     44      5    701      7    701
```

The `GPATree_assoc` function returns two columns. The first column contains binary values where 1 indicates that the SNP is associated with the phenotype and 0 indicates otherwise. The second column provides information regarding the leaf in which the SNP is located in the GPA-Tree plot. The `GPATree_assoc` allows both local (`fdrControl="local"`) and global FDR controls (`fdrControl="global"`) and users can set the threshold using the argument '`FDR`'. For `GPATreeExampleData`, GPA-Tree model identified 1458 risk SNPs at the nominal global FDR level of 0.05.

References

Leo, Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees*. CRC press.