Meta analysis SKAT Package

Seunggeun (Shawn) Lee

March 7, 2020

1 Overview

MetaSKAT is a package for meta-analysis Burden test, SKAT, SKAT-O. Users can carry out a gene-based test with all individual level genotype data or summary statistics from each study cohort. The package also provides functions to generate summary statistics.

2 Meta-analysis with individual level data

An example dataset (Example) has genotypes, phenotypes and covariates of 3 study cohorts.

```
> library(MetaSKAT)
> data(Example)
> names(Example)

[1] "y.list" "x.list" "n.g"    "Z.list"

> attach(Example)
```

To test associations, one needs to run Meta_Null_Model function first to obtain parameters and residuals from the null model of no associations. After, p-values can be calculated by running MetaSKAT_wZ.

```
> # continuous trait
> obj<-Meta_Null_Model(y.list, x.list, n.cohort=3, out_type="D")
> # rho=0 (SKAT)
> MetaSKAT_wZ(Z.list[[1]], obj)$p.value

[1] 0.1412563

> # rho=1 (weighted burden test)
> MetaSKAT_wZ(Z.list[[1]], obj, r.corr=1)$p.value

[1] 0.02621213
```

```
> # SKAT-O
> MetaSKAT_wZ(Z.list[[1]], obj, method="optimal")$p.value

[1] 0.04866566

>
```

In this example, MetaSKAT/MetaSKAT-O are conducted with assuming that genetic effects are homogeneous across study cohorts. In addition, the common weights from pooled MAFs are used. If one assumes genetic effects are heterogeneous across study cohorts and wants to use study specific MAFs to calculate weights, please use is.separate = TRUE (heterogeneous genetic effects) and combined.weight = FALSE (study specific MAFs).

```
> # rho=0 (SKAT)
> MetaSKAT_wZ(Z.list[[1]], obj, is.separate = TRUE, combined.weight=FALSE )$p.value

[1] 0.1067605

> # SKAT-O
> MetaSKAT_wZ(Z.list[[1]], obj, method="optimal", is.separate = TRUE,
+ combined.weight=FALSE)$p.value

[1] 0.09014461

>
```

Groups of study cohorts can be specified using Group_Idx to run tests with group specific heterogeneity. Suppose the first two cohorts are European-based and the last cohort is African American-based. If the ancestry group specific heterogeneity is assumed, one can set Group_Idx=c(1,1,2), which indicates the first two cohorts belong to the same group. The following example carries out MetaSKAT/MetaSKAT-O with group specific heterogeneity and group specific weights.

```
> # rho=0 (SKAT). First two cohorts belong to the same group
> MetaSKAT_wZ(Z.list[[1]], obj, is.separate = TRUE
+ , combined.weight=FALSE, Group_Idx=c(1,1,2))$p.value

[1] 0.1323768

> # SKAT-O. First two cohorts belong to the same group
> MetaSKAT_wZ(Z.list[[1]], obj, method="optimal"
+ , is.separate = TRUE, combined.weight=FALSE, Group_Idx=c(1,1,2))$p.value

[1] 0.06057264

>
```

3 Meta-analysis with summary data

3.1 Generate Meta SSD (MSSD) and Info (MInfo) files

MetaSKAT has a function to generate MSSD and MInfo files that have summary statistics. MSSD is a binary file with between relationship matrices of SNPs, and MInfo is a tex file with information on SNP sets. To generate them, the original data should be stored in binary plink formatted files, and users should provide a SetID file that defines SNP sets. The following code reads 01.bed, 01.bim, 01.SetID files and generates 01.MSSD and 01.MInfo files.

```
> File.SetID<-"./01.SetID"
> File.Bed<-"./01.bed"
> File.Bim<-"./01.bim"
> File.Fam<-"./01.fam"
> File.Mat<-"./01.MSSD"
> File.SetInfo<-"./01.MInfo"
> FAM<-read.table(File.Fam, header=FALSE)
> y<-FAM[,6]
> ########################################
> # Test Main File
>
> # need SKAT package to use SKAT_Null_Model function
> library(SKAT)
> N.Sample<-length(y)
> obj<-SKAT_Null_Model(y~1)
> Generate_Meta_Files(obj, File.Bed, File.Bim
+ , File.SetID, File.Mat, File.SetInfo,N.Sample)

Read SetID file
SetID file has 10 sets
Read Bim file
Bim file has 828 markers
Merge datasets and get set info
Save was done successfully!
Close the opened Bed file: /tmp/RtmpfneB3R/Rbuild369c7b0b5423/MetaSKAT/vignettes/01.bed

>
```

The following code generates MSSD and MInfo files of cohort 2 and 3.

```
> for( IDX_G in 2:3){
+
+         File.SetID<-sprintf("./%02d.SetID",IDX_G)
+         File.Bed<-sprintf("./%02d.bed",IDX_G)
+         File.Bim<-sprintf("./%02d.bim",IDX_G)
+         File.Fam<-sprintf("./%02d.fam",IDX_G)
```

```
+              File.Mat<-sprintf("./%02d.MSSD",IDX_G)
+              File.SetInfo<-sprintf("./%02d.MInfo",IDX_G)
+
+              FAM<-read.table(File.Fam, header=FALSE)
+              y<-FAM[,6]
+              N.Sample<-length(y)
+              obj<-SKAT_Null_Model(y~1)
+              re1<-Generate_Meta_Files(obj, File.Bed, File.Bim,
+              File.SetID, File.Mat, File.SetInfo, N.Sample)
+
+ }

Read SetID file
SetID file has 10 sets
Read Bim file
Bim file has 828 markers
Close the opened MSSD file: ./01.MSSD
Merge datasets and get set info
Save was done successfully!
Close the opened Bed file: /tmp/RtmpfneB3R/Rbuild369c7b0b5423/MetaSKAT/vignettes/02.bed
Read SetID file
SetID file has 10 sets
Read Bim file
Bim file has 828 markers
Close the opened MSSD file: ./02.MSSD
Merge datasets and get set info
Save was done successfully!
Close the opened Bed file: /tmp/RtmpfneB3R/Rbuild369c7b0b5423/MetaSKAT/vignettes/03.bed

>
```

3.2   Read Meta SSD and Info files, and run MetaSKAT

The following code opens MSSD and MInfo files from three study cohorts, and
then computes p-values.

```
> File.Mat.vec<-rep("",3)
> File.Info.vec<-rep("",3)
> for( IDX_G in 1:3){
+
+              File.Mat<-sprintf("./%02d.MSSD",IDX_G)
+              File.Info<-sprintf("./%02d.MInfo",IDX_G)
+
+              File.Mat.vec[IDX_G]<-File.Mat
+              File.Info.vec[IDX_G]<-File.Info
+
+ }
```

4

```
> # open files
>
> Cohort.Info<-Open_MSSD_File_2Read(File.Mat.vec, File.Info.vec)

Number of cohorts =  3
997 samples, 10 sets, 528 SNPs and 465 unique SNPs
997 samples, 10 sets, 506 SNPs and 457 unique SNPs
997 samples, 10 sets, 521 SNPs and 467 unique SNPs

> # get a p-value of the first set.
>
> MetaSKAT_MSSD_OneSet(Cohort.Info, SetID="1")$p.value

[1] 0.3383756

> # get p-values of all sets
>
> MetaSKAT_MSSD_ALL(Cohort.Info)

   SetID    p.value
1      1 0.3383756
2      2 0.5974347
3      3 0.8057104
4      4 0.7546481
5      5 0.9531031
6      6 0.6272782
7      7 0.7523350
8      8 0.9935740
9      9 0.6628574
10    10 0.9954041

>
```