

# GKnowMTest Package

July 27, 2017

## *Introduction*

GKnowMTest is a tool to integrate knowledge into SNP-level p-values (e.g. from a GWAS). Currently the package provides two functions. One is **anno.create** and other is a statistical function **prior.adjust**. First function creates a Mapping Object of class **annotatedSNPset**. It supports two kinds of priors on SNPs, namely SNP-level functional annotations (e.g. from ENCODE) and gene-level annotations (e.g. pathways). SNP-level functional annotations can also be obtained using another function **create.anno.mat**. A mapping object needs to be first created using the **anno.create** or **anno.merge** functions before running the statistical routine.

The statistical function **prior.adjust** calculates adjusted p-values based on some prior knowledge on the likelihood of SNPs to be associated. Currently prior knowledge can only be specified as groups of SNP sharing certain properties such as functional annotations or sets of pathways that they map to. The prior is specified as an **annotatedSNPset** object. Either Z-scores or raw p-values are taken as input, and prior-adjusted p-values are returned that can be compared to the usual genome-wide (e.g.  $5e-08$ ) or other multiple testing cutoffs.

```
> library(GKnowMTest)
```

## *How to use GKnowMtest*

### Creation of Mapping Object

First step is to create an object of 'annotatedSNPset' class which is referred as a Mapping Object. A mapping object is created by using 'anno.create' function. 'anno.create' takes rsIDs and pathwaylist/annotation data as input.

```
> snpfile <- system.file("sampleData", "snpData.rda", package="GKnowMTest")
> pathfile <- system.file("sampleData", "pathData.rda", package="GKnowMTest")
> anmfile <- system.file("sampleData", "anmData.rda", package="GKnowMTest")
```

Loads the input SNP data.frame, pathwaylist and annotation data from the rda files. SNP data.frame ('snpdf') contains SNPs with their Beta, SE and Z-score values. Pathway list ('pathlist') contains a list of 6 pathways where each pathway is a vector of gene-symbols. Annotation data is binary matrix of SNP-level annotation (DHS from ENCODE).

```

> ### snp dataframe ###
> load(snpfile)
> snp<-rownames(snpdf)
> head(snp)

[1] "rs45449492" "rs45452495" "rs617402"    "rs41460146" "rs12132517"
[6] "rs11240777"

> load(pathfile) ## loads a R list of gene symbols
> ##### First Pathway####
> pathlist[[1]][1:20]

[1] "AKR1B1" "G6PC"    "GAA"      "GALE"      "GALK1"    "GALK2"    "GALT"
[8] "GANC"   "GCK"     "B4GALT1"  "GLA"       "GLB1"     "HK1"      "HK2"
[15] "HK3"    "LALBA"   "LCT"      "PFKL"      "PFKM"     "PFKP"

> ### Last Pathway ###
> pathlist[[6]][100:120]

[1] "DVL3"    "EP300"   "KAT2A"    "HDAC1"    "HDAC2"    "HES1"     "RBPJ"     "JAG2"
[9] "LFNG"    "MFNG"    "NOTCH1"   "NOTCH2"   "NOTCH3"   "NOTCH4"   "PSEN1"    "PSEN2"
[17] "RFNG"    "ADAM17"  "NUMB"     "KAT2B"    "NUMBL"

> ##### Annotation Data #####
> load(anmfile)
> anm[1:8,1:10]

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
rs12132517  0  0  0  0  0  0  0  0  0  0
rs11240777  0  0  0  0  0  0  0  0  0  0
rs3748596   0  0  0  0  0  0  0  0  0  0
rs3766191   0  0  0  0  0  0  0  0  0  0
rs9442372   0  0  0  0  0  0  0  0  0  0
rs3737728   0  0  0  0  0  0  0  0  0  0
rs9442398   0  0  1  0  0  0  0  0  0  0
rs12726255  0  0  0  0  0  0  0  0  0  0

> ##### Creation of Mapping Object using Pathway list #####
>
> ob1=anno.create(snp,path.def=pathlist)

Obtaining base pair position..
Obtaining base pair position.. 25%
Obtaining base pair position.. 50%
Obtaining base pair position.. 75%
Obtaining base pair position..completed
Mapping Pathways To SNPs..
Mapping Pathways To SNPs.. 16%
Mapping Pathways To SNPs.. 33%
Mapping Pathways To SNPs.. 50%
Mapping Pathways To SNPs.. 66%
Mapping Pathways To SNPs.. 83%

```

```

Mapping Pathways To SNPs..completed
Creating Equivalence Class..
Creating Equivalence Class.. 8%
Creating Equivalence Class.. 16%
Creating Equivalence Class.. 25%
Creating Equivalence Class.. 33%
Creating Equivalence Class.. 41%
Creating Equivalence Class.. 50%
Creating Equivalence Class.. 58%
Creating Equivalence Class.. 66%
Creating Equivalence Class.. 75%
Creating Equivalence Class.. 83%
Creating Equivalence Class.. 91%
Creating Equivalence Class.. 100%

Creating Equivalence Class..completed
Created Mapping Object Successfully

> ##### Creation of Mapping Object using Annotation data #####
> #ob2<-anno.create(snp,anno.mat=anm)
>

```

## Overview of annotatedSNPset Object

Both ob1 and ob2 are objects of 'annotatedSNPset' class having four slots which are 'snp.df', 'snp.eq', 'eq.mat' and 'dim'. There are four accessor methods which are 'getDF', 'getEQ', 'getMAT' and 'getDIM' to access these slots respectively.

```

> ##### Slots of Mapping Object #####
> ob1

$snp.df ..... 10009 X 2 data.frame
      chrno  chrpos
rs45449492   NA      NA
rs45452495   NA      NA
rs617402      1 43480798
.....
.....
      chrno  chrpos
rs11599788   10 129023096
rs1186361    10 129025576
rs2489425    10 129034251

=====
Equivalence class of SNPs of length 10009
[1] 13 13 13 13 13 13 13 13 13 9 9 9 9 9 9 9 9 9 9

=====
Equivalence class By Path matrix of dimension 13 X 6
[,1] [,2] [,3] [,4]

```

```
[1,] 1 0 0 0
[2,] 1 1 0 0
[3,] 1 0 1 0
[4,] 0 1 0 0
```

```
=====
Equivalence class map matrix
NULL
```

```
=====
Dimensions of various elements of mapping Object
```

```
$ob_dim
      no_snps no_eq_class      no_anno      used_ob
      10009      13          6          1
```

```
$eq_class_arg
NULL
```

```
$no_anno_arg
NULL
```

```
> ### Acessor methods ###
>
> ### equivalence class of ob1 ####
> eq=getEQ(ob1)
> eq[1:20]
```

```
rs45449492 rs45452495      rs617402 rs41460146 rs12132517 rs11240777 rs3748596
      13      13      13      13      13      13      13
rs3766191 rs9442372 rs3737728 rs9442398 rs12726255 rs17160824 rs4072537
      13      13      9      9      9      9      9
rs10907182 rs9442380 rs4970362 rs11260542 rs6684820 rs10907175
      9      9      9      9      9      9
```

```
> ### equivalence class matrix of ob1 ###
>
> mat=getMAT(ob1)
> mat[1:10,1:6]
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 0 0 0 0 0
[2,] 1 1 0 0 0 0
[3,] 1 0 1 0 1 0
[4,] 0 1 0 0 0 0
[5,] 0 1 0 0 0 1
[6,] 0 0 1 0 0 0
[7,] 0 0 0 1 0 0
[8,] 1 0 0 1 0 0
[9,] 0 0 0 1 0 1
[10,] 0 0 0 0 1 0
```

```
> ### Dimension of ob1 ###
> getDIM(ob1)
```

no_snps	no_eq_class	no_anno	used_ob
10009	13	6	1

## Creation of Annotation Matrix

Package provides function 'create.anno.mat' to create annotation data which can be used as input of 'anno.create'. Here we have used DHS(from ENCODE) as SNP level annotation. User can use his/her own Annotation data. DHS(from ENCODE) can be downloaded via '<https://github.com/joepickrell/fgwas/>'

```
> #res=create.anno.mat(rownames(snpdf),base.path="/home/soumen/Datasets/Encode/",
> # fl.suffix=".annot.wdist.wcoding")
> # anm=res[[1]] #anm is the annotation matrix
>
> anm[1:8,1:10]
```

## Operations on Mapping Objects

### Merging of Mapping Objects

Multiple 'annotatedSNPset'-class objects can be merged as a single object of 'annotatedSNPset-class' by using the function 'anno.merge'

```
> obfile <- system.file("sampleData", "obData.rda", package="GKknowMTest")
> load(obfile) ##loads two mapping objects ob1 and ob2.
> ob3=anno.merge(list(ob1,ob2))
> ## Dimension of ob1 ##
> getDIM(ob1)
```

no_snps	no_eq_class	no_anno	used_ob
10009	13	6	1

```
> ## Dimension of ob2 ##
> getDIM(ob2)
```

no_snps	no_eq_class	no_anno	used_ob
10009	5217	450	1

```
> ## Dimension of ob3 ##
> getDIM(ob3)
```

no_of_snps	no_eq_class	no_of_anno	used_ob
10009	5404	456	2

### Omit SNPs from an Mapping object

Package provides function 'omit' to removes input SNPs from an mapping object and returns a new object.

```

> ## Dimension of ob1 before Omit ##
> getDIM(ob1)

      no_snps no_eq_class      no_anno      used_ob
      10009         13           6           1

> ob4<-omit(ob1,5:10)
> ## Dimension of ob1 after Omit ##
> getDIM(ob4)

      no_snps no_eq_class      no_anno      used_ob
      10003         13           6           1

```

## Use of statistical function

Once mapping object is created, the statistical function 'prior.adjust' is used to integrate knowledge into SNP-level p-values.

```

> snpfile <- system.file("sampleData", "snpData.rda", package="GKnowMTest")
> obfile <- system.file("sampleData", "obData.rda", package="GKnowMTest")
> load(snpfile) ## loads snp
> load(obfile) ## loads mapping object
> res.opt <- prior.adjust(snpdf,map.obj=ob1,prior.meth="Reg",adj.meth="pair.wt")

[1] "Fitting the global model ..."
[1] "Obtaining Prior ..."
[1] 1
[1] "Completed weighted analysis"
      user system elapsed
0.118  0.002  0.318

> head(res.opt[[1]],15)

      chrno  chrpos      Z      P      wP adjP      PPA
rs45449492   NA     NA -0.74434723 0.77166678 0.80559898 1 0.008220186
rs45452495   NA     NA  0.07139002 0.47154367 0.49227868 1 0.005388318
rs617402      1 43480798 -0.66348511 0.74649004 0.77931515 1 0.007690348
rs41460146   NA     NA -1.47675545 0.93012940 0.97102961 1 0.019191728
rs12132517      1  798801  0.21363924 0.41541420 0.43368104 1 0.005299064
rs11240777      1  798959  1.74833807 0.04020275 0.04197057 1 0.012959499
rs3748596      1  888639  0.52490902 0.29982322 0.31300723 1 0.005419903
rs3766191      1 1017587  0.50064060 0.30831204 0.32186932 1 0.005394632
rs9442372      1 1018704 -0.68293554 0.75267621 0.78577334 1 0.007810716
rs3737728      1 1021415 -1.58276081 0.94326200 1.00000000 1 0.017785465
rs9442398      1 1021695 -0.03932085 0.51568271 0.73150988 1 0.004352994
rs12726255      1 1049950  0.73780653 0.23031601 0.32670950 1 0.004545617
rs17160824      1 1060608  0.80909875 0.20922917 0.29679725 1 0.004680830
rs4072537      1 1065296  1.46767197 0.07109667 0.10085256 1 0.007500547
rs10907182      1 1066403  0.80563282 0.21022730 0.29821311 1 0.004673704

```