'IGESS' Package to integrating individual-level genotype data and summary statistics in genome-wide association studies

Mingwei Dai [1,2], Jingsi Ming [2], Mingxuan Cai [2], Jin Liu [3], Can Yang [2], Xiang Wan [4], Zongbe Xu [1]
[1] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China.
[2] Department of Mathematics, Hong Kong Baptist University, Hong Kong.
[3] Centre of Quantitative Medicine, Duke-NUS Graduate Medical School, Singapore.
[4] Department of Computer Science, Hong Kong Baptist University, Hong Kong.
Hong Kong.

October 20, 2017

1 Overview

This vignette provides an introduction to the 'IGESS' package. IGESS is a statistical approach to integrating individual level genotype data and summary statistics in Genome Wide Association Studies. This package provides computationally efficient and user friendly interface to fit and evaluate the IGESS model. It accepts both the R-type data and binary plink files.

The package can be loaded with the command:

```
R> library("IGESS")
```

This vignette is organized as follows. Section 2.1 discusses how to fit IGESS in various settings. Section 2.2 show how to evaluate the performance in terms of cross validation. Section 2.3 shows how to predict by the generated model.

2 Workflow

In this vignette, three different simulated data sets are used for demonstration. (1). R-type D1 = {X0, y0, P0} are genotype, phenotype and p-values, they have no information for the SNPs; (2) R-type D2= {X, y, P} are the counterparts, but they contain the information for the SNPs; (3) the genotyp data in the plink format are 'sim0.bed', 'sim0.fam', 'sim0.bim', the p-values stored in {P} are with SNP information. For the simulation data, $\{X, X0\}$ are both $N \times M$ matrix, where $N = 2000$ is the sample size and $M = 3000$ is the number of SNPs; $\{y, y_0\}$ are both $N \times 1$ vector; $\{P, P0\}$ are both $M \times K$ matrix, where $M = 3000$ is for the number of SNPs, $K = 7$ is for the nubmer of GWAS.

The R-type data used in this package could be loaded by the command.

```
R> data(DB)
```

The binary plink files could be accessed by

```
R> plinkfile <- gsub(".bim","",system.file("extdata", "sim0.bim", package = "IGESS"))
```

2.1 Fitting the IGESS

R package `IGESS` provides flexible statistical framework and automatically adjusts its model structure based on the provided data. The `IGESS` model could be fitted in the following three ways.

2.1.1 R-type data with no SNPs' information

In this subsection, the matrices of genotype data and p-values, which have not any information for SNPs, are used. It requires that

```
R> nrow(X0) == length(y0)

[1] TRUE

R> ncol(X0) == nrow(P0)

[1] TRUE
```

The complete IGESS function is,

```
R> fit <- IGESS(X, y, SS = NULL, opts = NULL, logfile = "screen", lbPval = 1e-12)
```

The genotype data `X` and the phenotype data `y` must be specified, the remaining parameters are optional, they have default values. To be specific, `SS` is for the summary statistics, `opts` is for the runing parameter setting, `logfile` is for the log file name (the default value 'screen' indicates that it would print the information on the screen) and `lbPval` is for the restriction of the minimal value of p-values. The output `fit` contains the parameters for the IGESS model, the detail would be mentioned in the following part.

The parameter `opts` has two fields, '`max_iter`' for the max number of iterations and '`dis_gap`' for the display gap of the printing message. Their default values are $(600, 60)$. They could be specified individually or simultaneously by either of the following commands.

```
R> opts = list(dis_gap=1)
R> opts = list(max_iter = 300)
R> opts = list(max_iter = 300,dis_gap=5)
```

The order for the parameters does not matter.

The IGESS model is fitted only with the genotype data:

```
R> fit <- IGESS(X0, y0)
```

The IGESS model integrates the genotype data {X0, y0} and summary statistics {P0} with the following command

```
R> fit <- IGESS(X0, y0, SS = P0)
```

2.1.2 R-type data with SNPs' information

If the genotype data and summary statitistics share only part of the set of SNPs, IGESS would take their intersection automaticly. The information for the genotype data and p-values are as follows.

```
R> str(X)
```

```
 num [1:2000, 1:3000] 0.647 -0.353 0.647 -0.353 -0.353 0.647 -0.353 0.647 -0.353 -0.353 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:3000] "rs1" "rs2" "rs3" "rs4" ...
```

```
R> str(P)
```

```
'data.frame':        3000 obs. of  7 variables:
 $ lab1: num  0.3217 0.1431 0.0511 0.3605 0.3478 ...
 $ lab2: num  0.305 0.744 0.32 0.572 0.605 ...
 $ lab3: num  0.3002 0.0364 0.6813 0.8258 0.7895 ...
 $ lab4: num  1 0.8204 0.0952 0.4333 0.8873 ...
 $ lab5: num  0.8729 0.2482 0.3401 0.0705 0.4139 ...
 $ lab6: num  0.669 0.355 1 0.142 1 ...
 $ lab7: num  1 0.546 1 0.181 0.139 ...
```

```
R> geno_snps = colnames(X)
R> ss_snps = rownames(P)
R> num_intersect <- intersect(geno_snps,ss_snps)
R> print(length(num_intersect))
```

```
[1] 2900
```

According to the above output, it could be seen that the genotype data and the summary statistics share 2900 SNPs, IGESS uses the data with respect to the intersection of the SNPs to fit the model.

```
R> fit <- IGESS(X, y, SS = P)
```

2.1.3 Binary plink file with R-type data storing the SNPs information

IGESS package also supports the input of binary plink file, which saves huge space for the genotype data.

The complete IGESS function is,

```
R> fit <- IGESS_Plink(genoplinkfile, SS = NULL, opts = NULL, logfile = "screen",
                                                                    lbPval = 1e-12)
```

In this scene, genotype data in the plink format take the place of R-type data $\{X, y\}$

```
R> fit <- IGESS_Plink(plinkfile, SS = P)
```

For the simulated data in this package, all the information contained in the plink files is the same as $\{X, y\}$ in $D2$. IGESS will take intersection as it does for $D2$.

The output for the above fitting is like following

```
R> str(fit)
```

```
List of 12
 $ sigma2beta: num 0.00176
 $ sigma2e   : num 0.153
 $ gammas    : num [1:2900, 1] 2.90e-07 1.54e-05 1.79e-05 2.32e-05 4.39e-07 ...
```

```
 $ mu        : num [1:2900, 1] -0.003598 -0.004521 0.016769 0.008822 -0.000339 ...
 $ S         : num [1:2900, 1] 0.000234 0.000165 0.000142 0.000221 0.000151 ...
 $ pi        : num 0.0348
 $ P         : num 2900
 $ fdr       : num [1:2900, 1] 1 1 1 1 1 ...
 $ cov       : num -0.664
 $ L         : num 1781
 $ iter      : num 6
 $ param_beta: num [1:7, 1] 0.122 0.142 0.14 0.125 0.124 ...
```

12 items of output are listed as above, the first 7 fields correspond to the notations $\sigma_\beta^2, \sigma_e^2, \{\gamma_j\}_1^M$, $\{\mu_j\}_1^M$, $\{s_j^2\}_1^M$, π, M. *cov* corresponds to the regression intercept for the IGESS model, L is the final lower bound, *iter* is the total iterations taken and `param_beta` is the α parameter for each Beta distribution for the p-values.

2.2 Evaluate the performance of prediction by cross validation

This section shows how to evaluate the performance of the model in terms of prediction accuracy by cross validation. Two corresponding functions are as follows

```
R> performance <- IGESSCV(X, y, SS = NULL, opts = NULL, logfile = "screen", lbPval = 1e-12,
```

and

```
R> performance <- IGESSCV_Plink(plinkfile, SS = NULL, opts = NULL, logfile = "screen",
```

The performance could be measured by *auc* or *mse*(by default) specified by the parameter `measure`. Besides, the parameter `opts` have a field `n_fold` to specify the number of folds for cross-validation as the previous one, the default value is 5. It could be specified as

```
R> opts = list(n_fold = 10)
```

The model could be evaluated without p-values

```
R>   performance <- IGESSCV(X, y)
R>   print(performance)

$mse
[1] 0.1945321
```

or with p-values

```
R>   performance <- IGESSCV(X, y, SS = P,measure = "auc")
R>   print(performance)

$auc
[1] 0.852946
```

or with genotype data in the plink format

```
R> performance <- IGESSCV_Plink(plinkfile, SS = P, measure = "auc")
```

One could use the `IGESSCV` or `IGESSCV_Plink` function to check which of these GWAS with summary statistics could result in better performance first, then run `IGESS` function to get the final model.

2.3 Predict with the fitted model

Once a model is fitted by `IGESS`, it could be used to predict the phenotype of the given genotype data by the following command.

```
R> yhat <- IGESS_Predict(fit, X)
```

Please contact Mingwei Dai at `daimingwei@gmail.com` for any questions or suggestions regarding the '`IGESS`' package.

References

[1] Mingwei Dai, Jingsi Ming, Mingxuan Cai, Jin Liu, Can Yang, Xiang Wan, Zongben Xu . IGESS: A Statistical Approach to Integrating Individual Level Genotype Data and Summary Statistics in Genome Wide Association Studies. Bioinformatics, 2017, 33(18): 2882-2889.