'LSMM' Package to integrating functional annotations with genome-wide association studies

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

September 5, 2017

1 Overview

This vignette provides an introduction to the 'LSMM' package. R package 'LSMM' implements LSMM (Latent Sparse Mixed Model), an efficient statistical approach to integrating functional annotations with genome-wide association studies. It provides model parameter estimation and statistical inference for risk SNPs and relevant annotations.

The package can be loaded with the command:

```
R> library("LSMM")
```

This vignette is organized as follows. Section 2.1 discusses how to fit LSMM in various settings. Section 2.2 explains command lines for statistical inference of risk SNPs and relevant annotations using LSMM.

Always feel free to contact Jingsi Ming at jsming@life.hkbu.edu.hk for any questions or suggestions regarding the 'LSMM' package.

2 Workflow

In this vignette, we use the simulated ExampleData in the package. We set the number of SNPs, fixed effects and random effects to be 100,000, 10 and 500 respectively. Users can find the p-value in the 'ExampleData$Pvalue', fixed effects in 'ExampleData$Z' and random effects in 'ExampleData$A'.

```
R> data(ExampleData)
R> Pvalue <- ExampleData$Pvalue
R> Z <- ExampleData$Z
R> A <- ExampleData$A
R> length(Pvalue)

[1] 100000

R> dim(Z)
```

```
[1] 100000     10
```

```
R> dim(A)
```

```
[1] 100000    500
```

The length of '`Pvalue`' is assumed to be the same as the number of rows of matrix provided to '`Z`' and '`A`'. When we analyze real data, we need to take the intersection of the SNPs in '`Pvalue`', '`Z`' and '`A`' and keep the SNPs in the same order.

2.1 Fitting the LSMM

We are now ready to fit LSMM using the data described above. R package `LSMM` provides flexible analysis framework and automatically adjusts its model structure based on the provided data.

First, assuming that there is no annotation data, we fit LSMM with the command:

```
R> fit.LSMM.noZA <- LSMM(Pvalue, Z = NULL, A = NULL)
```

or equivalently (which is actually simpler command),

```
R> fit.LSMM.noZA <- LSMM(Pvalue)
```

Now, LSMM reduces to the Two Groups Model.

When we also have related annotation data, this annotation data can be easily incorporated into LSMM by providing it in the second or the third argument of '`LSMM`' function. The second argument is regarded as fixed effect and the third argument is regarded as random effect. If we only consider fixed effects, then we can fit LSMM with the command:

```
R> fit.LSMM.Z <- LSMM(Pvalue, Z = Z, A = NULL)
```

If we only consider random effects,

```
R> fit.LSMM.A <- LSMM(Pvalue, Z = NULL, A = A)
```

If we consider both fixed and random effects,

```
R> fit.LSMM.ZA <- LSMM(Pvalue, Z = Z, A = A)
```

'`fit.LSMM.ZA`' is a list containing parameter estimation, the posterior probability and iteration times of each stage and the value of lower bound of log-likelihood.

```
R> str(fit.LSMM.ZA)
```

```
List of 14
 $ alpha            : num 0.199
 $ pi1.stage1       : num [1:100000, 1] 0.1149 0.0962 0.446 0.1087 0.0813 ...
 $ pi1.stage2       : num [1:100000, 1] 0.0916 0.0894 0.4256 0.1068 0.0755 ...
 $ pi1              : num [1:100000, 1] 5.28e-02 9.14e-06 1.77e-01 1.14e-02 4.19e-02 ...
 $ b                : num [1:11, 1] -1.785 -0.606 0.177 -0.866 1.54 ...
 $ sigma2           : num 1.11
 $ omega            : num 0.161
 $ omegak           : num [1:500, 1] 0.00444 0.00465 0.00475 0.00463 0.00516 ...
 $ beta             : num [1:500, 1] 1.85e-05 3.91e-05 -4.37e-05 3.75e-05 7.25e-05 ...
 $ Lq               : num [1:81, 1] 55936 57299 58124 58668 59052 ...
 $ iter_times.stage1: num 23
 $ iter_times.stage2: num 12
 $ iter_times.stage3: num 17
 $ iter_times.stage4: num 81
```

2.2 Statistical inference for risk SNPs and relevant annotations

Now, based on the fitted LSMM, we can make statistical inference for risk SNPs:

```
R> assoc.SNP.LSMM <- assoc.SNP(fit.LSMM.ZA, FDRset = 0.1, fdrControl="global")
R> str(assoc.SNP.LSMM)

List of 3
 $ gamma.stage1: num [1:100000] 0 0 0 0 0 1 1 0 0 0 ...
 $ gamma.stage2: num [1:100000] 0 0 0 0 0 1 0 0 0 0 ...
 $ gamma       : num [1:100000] 0 0 0 0 0 1 0 0 0 0 ...

R> table(assoc.SNP.LSMM$gamma.stage1)

    0     1
84723 15277

R> table(assoc.SNP.LSMM$gamma.stage2)

    0     1
84398 15602

R> table(assoc.SNP.LSMM$gamma)

    0     1
74941 25059
```

'`assoc.SNP`' function returns list of binary values indicating association of SNPs for the phenotype under different stages, where one indicates that the SNP is associated with the phenotype and zero otherwise. '`assoc.SNP`' allows both local ('`fdrControl="local"`') and global FDR controls ('`fdrControl="global"`') and users can set the threshold using the argument '`FDRset`'. For ExampleData, Two Groups Model (stage1) detected 15277 SNPs, whereas LFM which uses fixed effects (stage2) and LSMM which integrates both fixed effects and random effect identified 15602 and 25059 SNPs respectively, under the global FDR control at 0.1 level.

We can also make statistical inference for relevant annotations:

```
R> relev.Anno.LSMM <- relev.Anno(fit.LSMM.ZA, FDRset = 0.1, fdrControl="local")
R> str(relev.Anno.LSMM)

 num [1:500] 0 0 0 0 0 0 0 1 0 0 ...

R> table(relev.Anno.LSMM)

relev.Anno.LSMM
  0   1
422  78
```

'`relev.Anno`' function returns a list of binary values indicating relevance of annotations for the phenotype, where one indicates that the annotation is relevant to the phenotype and zero otherwise. '`relev.Anno`' allows both local ('`fdrControl="local"`') and global FDR controls ('`fdrControl="global"`') and users can set the threshold using the argument '`FDRset`'. For ExampleData, LSMM identified 78 relevant annotations, under the local FDR control at 0.1 level.

References