PedGFLMM Vignette

Yingda Jiang, Chi-Yang Chiu, Daniel E. Weeks, Ruzong Fan

April 10, 2019

Contents

1 Overview

This document describes our R package `PedGFLMM` which implements family-based additive generalized linear mixed models (GLMM) and generalized functional linear mixed models (GFLMM) for gene-based association testing of dichotomous traits (Jiang et al, 2019). Section 2 briefly describes the installation of the program. Section 3 describes the data formats. Section 4 explains how to run the program using one example. Section 5 offers explanation of the results and warnings to use the programs. Section 6 provides some suggestions and parameter choices for real data analysis.

The theoretical basis for this program is given in our research papers in Section 7 "References" below; the primary paper describing this work is Jiang et al (2019). Please cite the references if you use our program in any published work. In case of suggestions and questions and/or problems, you can contact us via e-mail (rf740@georgetown.edu).

An HTML version of this vignette can be found at https://github.com/DanielEWeeks/PedGFLMM.

2 Installation

The package is written in R language. To install, proceed as follows:

1. Install the `devtools` package by starting up R and issuing this command:

```
install.packages("devtools")
```

2. Load the `devtools` library to make its commands available:

```
library(devtools)
```

3. Install the `PedGFLMM` R package from the github repository via this command:

```
install_github("DanielEWeeks/PedGFLMM/pkg")
```

If you wish to have this vignette installed and accessible within your R help pages, use this command instead (but note that this will be slower):

```
install_github("DanielEWeeks/PedGFLMM/pkg", build_opts = c("--no-resave-data", "--no-manual"),
   build_vignettes = TRUE)
```

Note that this vignette is available online at https://github.com/DanielEWeeks/PedGFLMM.

After the `PedGFLMM` R package has been installed, you can view this vignette by issuing these commands at the R prompt:

```
library(PedGFLMM)
browseVignettes("PedGFLMM")
```

3 Data Format

The program needs data frame in R to define the pedigree structure (typical format used by LINKAGE and PLINK), genotypes, SNP positions, and covariates.

3.1 The pedigree file

The pedigree file is in the same format as that used by the `pedgene` R package except for a column named `ID` (Schaid et al. 2013) and has the following columns:

- ID: identify of each individual.
- ped: pedigree ID, character or numeric allowed.
- person: person ID, a unique ID within each pedigree, numeric or character allowed.
- father: father ID, NA if no father.
- mother: mother ID, NA if no mother.
- sex: coded as 1 for male, 2 for female.
- trait: phenotype, either case-control status coded as 1 for affected and 0 for unaffected. Subjects with missing (NA) will be removed from the analysis.

Table 1: Table 1: The first 6 lines of the example pedigree file

ID	ped	person	father	mother	sex	trait
1	1	1	0	0	1	0
2	1	2	0	0	2	0
3	1	3	1	2	1	0
4	1	4	1	2	1	1
5	1	5	1	2	2	1
6	1	2134	0	0	2	0

3.2 The genotype file

The genotype file is a matrix with genotypes for subjects (rows) at each variant position (columns). The first two columns are required to be named 'ped' and 'person', which are used to match subjects to their data in the pedigree data.frame. The genotypes are coded as 0, 1, 2 for autosomal markers (typically a count of the number of the minor allele).

Table 2: Table 2: The first 6 lines of the example genotype file (first 15 columns)

ped	person	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
1	3	0	1	0	0	0	0	0	0	0	0	0	0	0
1	4	0	1	0	0	0	0	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2134	0	0	0	0	0	1	0	0	0	0	0	0	0

3.3 The map file

The map file provides SNP positions for each SNP. The first column is required for the chromosome number, the second column is for the name of SNPs in the genotype file, and the third column is the position of SNPs in base pairs.

Table 3: Table 3: The first 6 lines of the example map file

chr	snp	pos
1	V1	448234
1	V2	448239
1	V3	448275
1	V4	448318
1	V5	448418
1	V6	448438

3.4 The covariate file (optional)

The covariate file contains covariates. The first two columns are required to be named 'ped' and 'person', which are used to match subjects to their data in the pedigree data.frame.

Table 4: Table 4: The first 6 lines of the example covariate file

ped	person	X1	X2
1	1	0	0.2165418
1	2	0	0.9083806
1	3	0	1.6420150
1	4	1	1.1938604
1	5	1	-0.8231785
1	2134	0	-0.0987421

4 How to Run the Program

There are three main functions in this package which implement the statistics described in Jiang et al (2019):

1. `PedGLMM_additive_effect_model`
2. `PedGFLMM_beta_smooth_only`
3. `PedGFLMM_fixed_model`

After installing this `PedGFLMM` R package, you can access help pages for each of these functions easily, which contain example code. For example, to access the help page for the `PedGFLMM_beta_smooth_only` function, proceed as follows in R:

```
library(PedGFLMM)
?PedGFLMM_beta_smooth_only
```

We also provide an example `Mega2PedGFLMM` function that illustrates how to use functions from the `Mega2R` R package to easily loop through genes, computing the `PedGFLMM_beta_smooth_only` statistics for each gene.

4.1 Load the example data

To illustrate the functions in this package, we first load the data.

```
library(PedGFLMM)
data(exampleData)
# The 'exampleData' contains four data frames: Ped, geno, cov, snpPos
ls()
```

```
## [1] "cov"    "geno"   "Ped"    "snpPos"
```

4.2 The PedGLMM_additive_effect_model function

This function carries out a region-based association test using our additive generalized linear mixed model (GLMM), as described in Jiang et al (2019). This statistics may not be powerful when the number of genetic variants is large.

```
add = PedGLMM_additive_effect_model(ped = Ped, geno = as.matrix(geno), covariate = as.matrix(cov))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Hessian is numerically singular: parameters are not uniquely
## determined
```

```
add
```

```
## $LRT
## [1] 0.1932793
```

For the case without covariates,

```
add_no_cov = PedGLMM_additive_effect_model(ped = Ped, geno = as.matrix(geno), covariate = NULL)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?
```

```
add_no_cov
```

```
## $LRT
## [1] 0.2396751
```

As you can see from the examples above, the `PedGLMM_additive_effect_model` function had convergence problems. If this happens, instead of using the `PedGLMM_additive_effect_model`, one can use either `PedGFLMM_beta_smooth_only` or `PedGFLMM_fixed_model` instead.

4.3 The PedGFLMM_beta_smooth_only function

This function carries out a region-based association test using our 'beta smooth only' generalized functional linear mixed model (GFLMM), where the genetic effect function is assumed to be continuous/smooth. For details, see Jiang et al (2019).

The genetic effect function can be expanded using either B-spline or Fourier basis functions, and the order and number of basis functions need to be specified by the user. For small sample size datasets, one may have problems of convergence. Then, one can use a smaller number of basis functions, e.g., `betabasis_Bsp` = 6, etc.

```
betabasis_Bsp = 10
betabasis_Fsp = 11

order = 4

bsmooth_bsp = PedGFLMM_beta_smooth_only(ped = Ped, geno = as.matrix(geno), pos = snpPos$pos,
    order = order, beta_basis = betabasis_Bsp, covariate = as.matrix(cov), base = "bspline")
bsmooth_bsp
```

```
## $LRT
## [1] 0.5503745
```

```
bsmooth_fsp = PedGFLMM_beta_smooth_only(ped = Ped, geno = as.matrix(geno), pos = snpPos$pos,
    order = order, beta_basis = betabasis_Fsp, covariate = as.matrix(cov), base = "fspline")
bsmooth_fsp
```

```
## $LRT
## [1] 0.4967854
```

For the case without covariates,

```
bsmooth_bsp_no_cov = PedGFLMM_beta_smooth_only(ped = Ped, geno = as.matrix(geno),
    pos = snpPos$pos, order = order, beta_basis = betabasis_Bsp, covariate = NULL,
    base = "bspline")
bsmooth_bsp_no_cov
```

```
## $LRT
## [1] 0.5443719
```

```
bsmooth_fsp_no_cov = PedGFLMM_beta_smooth_only(ped = Ped, geno = as.matrix(geno),
    pos = snpPos$pos, order = order, beta_basis = betabasis_Fsp, covariate = NULL,
    base = "fspline")
bsmooth_fsp_no_cov
```

```
## $LRT
## [1] 0.6264945
```

4.4 The PedGFLMM_fixed_model function

This function carries out a region-based association test using our generalized functional linear mixed model (GFLMM). For details, see Jiang et al (2019).

The genetic variant function (GVF) and the genetic effect function can be expanded using either B-spline or Fourier basis functions, and the order and number of basis functions need to be specified by the user.

```
betabasis_Bsp = 10
genobasis_Bsp = 10

betabasis_Fsp = 11
genobasis_Fsp = 11
order = 4

fixed_bsp = PedGFLMM_fixed_model(ped = Ped, geno = as.matrix(geno), pos = snpPos$pos,
    order = order, beta_basis = betabasis_Bsp, geno_basis = genobasis_Bsp, covariate = as.matrix(cov),
    base = "bspline")
fixed_bsp
```

```
## $LRT
## [1] 0.5503745
```

```
fixed_fsp = PedGFLMM_fixed_model(ped = Ped, geno = as.matrix(geno), pos = snpPos$pos,
    order = order, beta_basis = betabasis_Fsp, geno_basis = genobasis_Fsp, covariate = as.matrix(cov),
    base = "fspline")
fixed_fsp
```

```
## $LRT
## [1] 0.4967854
```

For the case without covariates,

```
betabasis_Bsp = 10
genobasis_Bsp = 10

betabasis_Fsp = 11
genobasis_Fsp = 11
order = 4

fixed_bsp_no_cov = PedGFLMM_fixed_model(ped = Ped, geno = as.matrix(geno), pos = snpPos$pos,
    order = order, beta_basis = betabasis_Bsp, geno_basis = genobasis_Bsp, covariate = NULL,
    base = "bspline")
fixed_bsp_no_cov
```

```
## $LRT
## [1] 0.5443719
```

```
fixed_fsp_no_cov = PedGFLMM_fixed_model(ped = Ped, geno = as.matrix(geno), pos = snpPos$pos,
    order = order, beta_basis = betabasis_Fsp, geno_basis = genobasis_Fsp, covariate = NULL,
    base = "fspline")
fixed_fsp_no_cov
```

```
## $LRT
## [1] 0.6264945
```

4.5 The Mega2PedGFLMM function

The `Mega2PedGFLMM` function illustrates how to use functions from the `Mega2R` R package (Baron et al, 2018) to easily loop through genes, computing the `PedGFLMM_beta_smooth_only` statistics for each gene.

It will skip genes that contain less than two polymorphic markers.

The `Mega2R` R package is designed to read a database of phenotypes and genetic data, created by the data-reformatting program `Mega2` (Baron et al, 2014), into R as a set of coordinated data frames. Once these data are nicely accessible in R, then `Mega2R` functions can be used to loop through gene regions, computing

statistics for each region. By studying the code of this example `Mega2PedGFLMM` function, one can figure out how to loop through one's own data in a similar fashion.

```
db = system.file("exdata", "seqsimmGFLMM.db", package = "PedGFLMM")
ENV = init_PedGFLMM(db)
ENV$verbose = TRUE
Mega2PedGFLMM(gs = 53:54)
```

```
## tryFn() No markers in range: NM_014705, DOCK4, 7, 111366163, 111846462
```

```
## CEP104 snp24037 34 35 0
## CEP104 snp24039 69 0 0
## CEP104 snp24041 49 20 0
## CEP104 snp24048 37 30 2
## CEP104 snp24494 60 8 1
## CEP104 snp24499 56 13 0
## CEP104 snp24506 24 36 9
## CEP104 snp24507 48 18 3
```

```
## tryFn() <simpleWarning:: Model failed to converge with max|grad| = 0.0264779 (tol = 0.001, component
```

```
## tryFn() <simpleWarning:: Model failed to converge with max|grad| = 0.0264779 (tol = 0.001, component
```

```
##   chr   gene nvariants   start     end LRT.bsm.Bsp LRT.bsm.fsp
## 1   1 CEP104         7 3728644 3773797   0.4465485   0.6334223
```

```
Mega2PedGFLMM(genes = "CEP104")
```

```
##
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
## tryFn() No markers in range: 9731, CEP104, CEP104, 4283, uc031pky.1, 1, -, 3732844, 3753716
```

```
## CEP104 snp24037 34 35 0
## CEP104 snp24039 69 0 0
## CEP104 snp24041 49 20 0
## CEP104 snp24048 37 30 2
## CEP104 snp24494 60 8 1
## CEP104 snp24499 56 13 0
## CEP104 snp24506 24 36 9
## CEP104 snp24507 48 18 3
```

```
## tryFn() <simpleWarning:: Model failed to converge with max|grad| = 0.0264779 (tol = 0.001, component
```

```
## tryFn() <simpleWarning:: Model failed to converge with max|grad| = 0.0264779 (tol = 0.001, component
```

```
##   chr   gene nvariants   start     end LRT.bsm.Bsp LRT.bsm.fsp
## 2   1 CEP104         7 3728645 3773797   0.4465485   0.6334223
```

5 Explanation of the Results and Warnings

As shown above, our program outputs the p-value based on likelihood ratio test (LRT). The LRT is conservative and has good power performance.

6 Suggestions and Parameters for Real Data Analysis

In this documentation, we present three R functions to perform gene-based association analysis of dichotomous traits using family data. The two PedGFLMM functions, 'PedGFLMM_fixed_model.R' and 'PedGFLMM_beta_smooth_only.R', usually provide very similar results. In practice, one may use one of them for data analysis. We suggest to use PedGFLMM_fixed_model by either B-spline or Fourier spline basis functions and report the p-values of LRT. If the number of SNPs is not very big, we suggest the following parameters for a data analysis:

First, set `order = 4`.

For a B-spline basis, set `beta_basis = 10` and `geno_basis = 10`.

For a Fourier spline basis, set `beta_basis = 11` and `geno_basis = 11`.

If the number of SNPs is large, one should try `order = 4` and:

For a B-spline basis, set `beta_basis = 15` and `geno_basis = 15`.

For a Fourier spline basis, set `beta_basis = 16` and `geno_basis = 16`.

It may be necessary to set `beta_basis` and `geno_basis` to even larger numbers. The point is that the parameters should be large enough to sufficiently expand information of the genetic data, but can not be too large to decrease the power.

Note if an even number is specified for the basis of the Fourier spline basis function, it is rounded up to the nearest odd integer.

7 References

Baron RV, Kollar C, Mukhopadhyay N, Weeks DE. (2014) Mega2: validated data-reformatting for linkage and association analyses. *Source Code Biol Med.* **9(1)**:26. doi: 10.1186/s13029-014-0026-y.

Baron RV, Stickel JR, Weeks DE. (2018) The Mega2R package: R tools for accessing and processing genetic data in common formats. *F1000Res.* **29(7)**:1352. doi: 10.12688/f1000research.15949.1.

Chiu CY, Yuan F, Zhang BS, Yuan A, Li X, Fang HB, Lange K, Weeks DE, Wilson AF, Bailey-Wilson JE, Lakhal-Chaieb ML, Cook RJ, McMahon FJ, Amos CI, Xiong MM, and Fan RZ (2019) Pedigree-based linear mixed models for association analysis of quantitative traits with next-generation sequencing data. *Genetic Epidemiology* **43(2)**:189-206.

Fan RZ, Wang YF, Mills JL, Wilson AF, Bailey-Wilson JE, and Xiong MM (2013) Functional linear models for association analysis of quantitative traits. *Genetic Epidemiology* **37(7)**:726-742.

Fan RZ, Wang YF, Mills JL, Carter TC, Lobach I, Wilson AF, Bailey-Wilson JE, Weeks DE, and Xiong MM (2014) Generalized functional linear models for case-control association studies. *Genetic Epidemiology* **38(7)**:622-637.

Jiang YD, Chiu CY, Yan Q, Chen W, Gorin MB, Conley YP, Lakhal-Chaieb ML, Cook RJ, Amos CI, Wilson AF, Bailey-Wilson JE, McMahon FJ, Vazquez AI, Yuan A, Zhong XG, Xiong MM, Weeks DE, and Fan RZ (2019) Gene-based association testing of dichotomous traits with generalized linear mixed models for family data.

Schaid DJ, McDonnell SK, Sinnwell JP, and Thibodeau SN (2013) Multiple genetic variant association testing by collapsing and kernel methods with pedigree or population structured data. *Genetic Epidemiology* **37**:409-418.

8 Copyright Information