User Manual

FaST-LMM

Factored Spectrally Transformed Linear Mixed Models

Version 1.08

Microsoft Research

March 13, 2012

Introduction

FaST-LMM, which stands for *Fac*tored *S*pectrally *T*ransformed *L*inear *M*ixed *M*odels is a program for performing genome-wide association studies (GWAS) on large data sets. It runs on both Windows and Linux systems, and has been tested on data sets with over 120,000 individuals.

This software is available as open source under the Apache license ver.2.0 at http://mscompbio.codeplex.com. A copy of the Apache License can also be found in the root of the project in the file LICENSE.TXT.

For help with the software, please contact

Christoph Lippert, christoph.a.lippert@gmail.com
Jennifer Listgarten, jennl@microsoft.com
Carl Kadie, carlk@microsoft.com
Bob Davidson, bobd@microsoft.com
David Heckerman, heckerma@microsoft.com

Citing FaST-LMM

If you use FaST-LMM in any published work, please cite both the software (using the link http://mscompbio.codeplex.com/) and the manuscript describing it:

> C. Lippert, J. Listgarten, Y. Liu, C.M. Kadie, R.I. Davidson, and D. Heckerman. FaST Linear Mixed Models for Genome-Wide Association Studies. *Nature Methods*, published online 4 Sep 2011 (doi:10.1038/nmeth.1681).

Also, we would appreciate it if you let us know that you are citing it.

Installing FaST-LMM

FaST-LMM is available as a .zip file that exacts to these directories:

`fastlmm/Bin`	contains the compiled executable files
`fastlmm/Cpp`	contains C++ source and project files
`fastlmm/CSharp`	contains C# source and project
`fastlmm/Data/sampledata`	contains sample data and command script
`fastlmm/Doc`	contains project documentation
`fastlmm/Externals`	contains other code FaSTLMM depends on

There are executables for Windows (64bit), and for Ubuntu Linux (64bit) under the fastlmm\Bin directory and all required .dll files are included in the respective directories. These executables use the MKL math library, which is optimized for Intel processors but also runs on AMD processors. If one of these options is suitable, please skip ahead to section "Data Preparation" to see how to run FaST-LMM on your data. If not, please see the next section.

Compiling FaST-LMM

In addition to the source code, the following external dependencies must be installed and met in order to build FaSTLMM:

Building for Windows

Both C# and C++ versions require Visual Studio 2010 (VS)-
A version of VS (Express through Universal) is capable of building FaSTLMM. If you do not already have a copy of Visual Studio, the Visual Studio 2010 Express edition can be freely downloaded from
http://www.microsoft.com/express/downloads

- For the C# version
  Parts of program are capable of running against the Microsoft HPC cluster environment. To build you must install the "HPC Pack 2008 R2 Client Utilities Redistributable Package with Service Pack 2". This is freely available at:
  http://www.microsoft.com/download/en/details.aspx?id=17017

  With the HPC library installed, no additional libraries are required to compile the C# version of FaSTLMM. Double-click the Gwas\FaSTLMM.sln file to load Visual Studio and then build the solution. If a reference to the HPC library is not resolved automatically during the load, examine the references and double click the indicated library. If the HPC library installed properly, Visual Studio should successfully resolve the request and you can proceed with your build. The program builds in Gwas\bin.

- For the C++ version
  FastLmmC uses a 3rd party math library for advanced math functions and performance. FastLmmC can use either Intel's MKL or AMD's ACML math libraries. Once you have installed the appropriate library, use the Visual Studio IDE to select the appropriate configuration from the solution and build. ACML requires an additional step to tell Visual Studio where it is located. You must set the environment variable ACML_ROOT to point to your install location or libraries will not be located—for example,

  ```
  C>set ACML_ROOT=C:\AMD\acml4.4.0
  ```

  You can find more about the math libraries at their respective web sites:
  http://software.intel.com/en-us/articles/intel-mkl
  http://developer.amd.com/libraries/acml/pages/default.aspx

  With a math library installed, no additional libraries are required to compile the C++ version of FaSTLMM (FastLmmC). Double-click the FastLmmC.sln file to load Visual Studio and then build the solution associated with your library.

Building the C++ version of FaSTLMM for Linux

FaSTLMM is primarily developed and tested on Windows although we are able to build the C++ version for Linux. We provide a simple script file that uses the GNU toolset with the 3rd party math library to compile the sources in a Linux environment.

- FastLmmC uses a 3rd party math library for advanced math functions and performance. The program has been run on Ubuntu Linux and can use either Intel's MKL or AMD's ACML math libraries for Linux. Once you have selected and installed the appropriate library, you can then build using the appropriate script file located in the Cpp directory. Review of the two files, DoMKL_linux and DoAcml_linux, will show very simple scripts to compile the program using g++ and then link the .o files with the appropriate math library. The *.o files are written to version specific directories, so it is necessary to create the appropriate directory prior to running the script. For more details, see the script.

  You can find more about the math libraries for Linux at their respective web sites:
  http://software.intel.com/en-us/articles/intel-mkl
  http://developer.amd.com/libraries/acml/pages/default.aspx

Data preparation

FaST-LMM uses four input files containing (1) the SNP data to be tested, (2) the SNP data used to determine the genetic similarities between individuals (which can be different from 1), (3) the phenotype data, and (4, optionally) a set of covariates.

When the realized relationship matrix (RRM) is used for genetic similarity, and when the number of SNPs used to construct the RRM is less than the number of individuals, the runtime and memory footprint of FaST-LMM scales linearly in the number of individuals in the data. When this condition is not met, the runtime and memory footprint of FaST-LMM are cubic and quadratic in the number of individuals, respectively.

All input files should be in ASCII.

Both SNP files (1 and 2 above) should be in PLINK format (`ped/map`, `tped/tfam`, `bed/bim/fam`, or `fam/dat/map`). For the most speed, use the binary format in SNP major order. The phenotype entries in these files must be set to some dummy value and will be ignored (our software uses a separate phenotype file). Sex should be encoded as a single digit. See the PLINK manual http://pngu.mgh.harvard.edu/~purcell/plink/ [1] for further details. Missing SNP values will be mean imputed. Dosages files are also allowed (see the end of this section).

The required file containing the phenotype (3 above) uses the PLINK alternate phenotype format. It should have at least three columns: `<familyID>`, `<individualID>`, and any number of `<phenotype value>`. The columns are delimited by whitespace (<tab> or <space>). The default option is to test the first phenotype only. A missing value should be denoted by `-9`, but this can be changed (see options below). The first column, `<familyID>`, is joined with the second column `<individualID>` to create a unique key for the individual that matches an entry for an individual in the PLINK files above.

Example phenotype file for two phenotypes `(fastlmm/data/sampledata/pheno.txt)`**:**

```
1     IND0  2     3.05043
1     IND1  2     1.72797
1     IND2  -9    4.19592
1     IND3  2     3.4492
1     IND4  1     -8.99843
1     IND5  1     -0.768613
1     IND6  2     6.73734
...
```

Optionally, the phenotype file may also have a header row, for example, as follows:

```
FID   IID   MyPheno    YourPheno
```

The optional file containing covariates should have at least three columns: `<familyID>`, `<individualID>`, and any number of `<covariate value>`. The columns should be tab delimited. The token for missing values must be the same as that used in the phenotype file. All covariates are processed. Covariate files should not have a header row.

Example covariate file `(fastlmm/data/sampledata/covariate.txt)`**:**

```
1     IND0  1
1     IND1  1
1     IND2  1
1     IND3  1
1     IND4  1
1     IND5  -9
1     IND6  1
...
```

Instead of SNP data from which genetic similarities are computed, the user may provide the genetic similarities directly using the `-sim <filename>` option. The file containing the genetic similarities should be tab delimited and have both row and column labels for the individual IDs. The value in the top-left corner of the file should be `var`.

Example similarities file:

```
var   IND0  IND1  IND2  IND3  ...
IND0  1.0   0.5   0.5   0.25  ...
IND1  0.5   1.0   0.5   0.5   ...
IND2  0.25  0.5   1.0   0.5   ...
...
```

SNP dosages are specified using a `.dat` file.

Example dosage file:

```
SNP     A1  A2   Fam1 Ind1   Fam1 Ind2   Fam2 Ind3
rs0001   A   C   0.98 0.02   1.00 0.00   0.00 0.01
rs0002   G   A   0.00 1.00   0.00 0.00   0.99 0.01
```

This file represents data for two SNPs on three individuals. The first three columns list the SNP, first nucleotide, and second nucleotide. The minor allele is coded `A1` and the major allele is coded `A2`. Each genotype is represented by two numbers. Here, the two numbers for the first SNP represent the probability of an `A/A`, then an `A/C` genotype. The probability of a `C/C` is 1 minus the sum of these. The header row is optional, but if used, it must start with '`SNP A1 A2`' and have a FamilyId / IndividualId pair for each genotype probability pair. If there is no header, the genotype entries must be in the same order as found in the `.fam` file. Dosage files typically do not contain missing data, but `-9 -9` may be used to specify a missing entry.

To use a dosage file, replace the `–file` and `–fileSim` commands with `–dosage` and `–dosageSim`, respectively. In addition to the `.dat` file, a `.fam` file is required. The entries in the `.dat` file must correspond to entries in the `.fam` file. A `.map` file is optional and will fill out the additional SNP location information.

Running FaST-LMM

Once you have prepared the files in the proper format, you can run FaST-LMM. Here is a sample call on the synthetic data provided in the .zip file from the directory containing that data and assuming fastlmmc is in the path:

```
> fastlmmc -tfile geno_test -tfilesim geno_cov -pheno pheno.txt -covar
covariate.txt -mpheno 1
```

You should see something like the following output on the screen (using the C++ version):

```
FastLmmC v1.08.20120313 - Factored Spectrally Transformed Linear Mixed Models
  Copyright Microsoft Corporation
  Compiled Mar 13 2012 at 06:20:44 by BOBD01 for Windows
  using MKL v10.3.5 - Build: 20110720

 ++    Start Processing CommandLine:
 --      End Processing CommandLine:
 ++      Start Loading FastLmm Data:
  ++  Start Loading Covariance Data:
   ++      Processing PLINK fileset: [geno_cov]
     Number of Individuals Selected:     270
    Number of Phenotypes/Individual:       1
          Number of SNPs/Individual:     200
     Number of SNPs/Individual Used:     200
   --  End Processing PLINK fileset: [geno_cov]
  --    End Loading Covariance Data:
  ++        Start Loading Test Data:
   ++      Processing PLINK fileset: [geno_test]
     Number of Individuals Selected:     270
    Number of Phenotypes/Individual:       1
          Number of SNPs/Individual:     200
     Number of SNPs/Individual Used:     200
   --  End Processing PLINK fileset: [geno_test]
  --          End Loading Test Data:
 --        End Loading FastLmm Data:
             Compute/Load EigenSym:
          Compute Woodbury low rank: ... done.
             Compute GWAs using LMM:
 --          Start lowrank training:
 --           Lowrank training done:
                      Write output: [geno_test.out.txt]
                 Total elapsed time: 311.918 ms
```

When the output file `[geno_test.out.txt]` is loaded in Excel, it should look as follows:

SNP	Chromosc	GeneticDi	Position	Pvalue	Qvalue	N	NullLogLik	AltLogLike	SNPWeigh	SNPWeigh	WaldStat	NullLogDe	NullGene	NullResid	NullBias
snp2	1	0	3	5.42E-08	1.08E-05	200	-1.83E+02	-1.68E+02	-1.76E-01	1.92E-03	0	1.69E+00	3.69E-02	2.01E-01	1.53E+00
snp110	1	0	111	6.29E-03	6.29E-01	200	-1.83E+02	-1.79E+02	1.27E-01	2.82E-03	0	1.69E+00	3.69E-02	2.01E-01	1.53E+00
snp55	1	0	56	1.60E-02	8.15E-01	200	-1.83E+02	-1.80E+02	8.12E-02	2.04E-03	0	1.69E+00	3.69E-02	2.01E-01	1.53E+00
snp167	1	0	168	1.74E-02	8.15E-01	200	-1.83E+02	-1.80E+02	8.54E-02	2.17E-03	0	1.69E+00	3.69E-02	2.01E-01	1.53E+00
snp140	1	0	141	2.04E-02	8.15E-01	200	-1.83E+02	-1.80E+02	-8.12E-02	2.12E-03	0	1.69E+00	3.69E-02	2.01E-01	1.53E+00
snp171	1	0	172	3.14E-02	8.26E-01	200	-1.83E+02	-1.80E+02	9.36E-02	2.64E-03	0	1.69E+00	3.69E-02	2.01E-01	1.53E+00
snp144	1	0	145	3.27E-02	8.26E-01	200	-1.83E+02	-1.80E+02	7.51E-02	2.13E-03	0	1.69E+00	3.69E-02	2.01E-01	1.53E+00

…

The standard and `-verboseOut` columns are:

`SNP`

The rs# or SNP identifier for the SNP tested. Taken from the PLINK file.

`Chromosome`

The chromosome identifier for the SNP tested or 0 if unplaced. Taken from the PLINK file.

`Genetic Distance`

The location of the SNP on the chromosome. Taken from the PLINK file. Any units are allowed, but typically centimorgans or morgans are used.

`Position`

The base-pair position of the SNP on the chromosome (bp units). Taken from the PLINK file.

`Phenotype [under -verboseOut]`

The name of the phenotype as specified in the header of the phenotype file. `NoName` means that no header row was specified.

`Pvalue`

The p-value computed for the SNP tested

`Qvalue`

The *q*-value computed for the SNP tested estimated from the *p*-values of all test-SNPs in the PLINK file using the procedure of Benjamini and Hochberg

`N`

The sample size or number of individuals that have a been used for this analysis

`NumSNPsExcluded [under -excludeByGeneticDistance]`

`IndexExclusionStart [under -excludeByGeneticDistance]`

`DOF [under -verboseOut]`

The degrees of freedom of the statistical test

`NullLogLike`

The log likelihood of the null model

`AltLogLike`

The log likelihood of the alternative model

`SnpWeight`

The fixed-effect weight of the SNP

`SnpWeightSE`

The standard error of the SnpWeight

`WaldStat`

The Wald stat of the SnpWeight

`NullLogDelta`

The ratio between the residual variance and the genetic variance $\delta = \sigma_e^2/\sigma_g^2$ on the null model

`NullGeneticVar`

The genetic variance σ_g^2 on the null model

`NullResidualVar`

The residual variance σ_e^2 on the alternative model

`NullBias`
: The offset term in the null model

`LogDelta [under –verboseOut]`
: The ratio between the residual variance and the genetic variance $\delta = \sigma_e^2/\sigma_g^2$ on the alternative model

`geneticVar [under –verboseOut]`
: The genetic variance σ_g^2 on the alternative model

`ResidualVar [under –verboseOut]`
: The residual variance σ_e^2 on the alternative model

`NullBias [under –verboseOut]`
: The offset term in the alternative model

`SNPIndex`
: The column index of the SNP tested in the PLINK file starting at 1

`SNPCount`
: The number of SNPs tested

Speed vs. accuracy considerations

The FaST-LMM inference involves a search over the ratio δ of genetic and environmental variances. As this step represents a non-convex optimization FaST-LMM performs an optimization procedure over several intervals on a logarithmic scale, invoking iterative calls to the likelihood function. The total run-time of this step scales linear in the sample size times a constant that approximately equals the number of intervals considered for the search.

For maximum speed, the command line option `-simLearnType Once` is set by default, removing this constant factor for every SNP tested. Using this option, the ratio δ is found on the null-model only and is fixed to that value throughout the testing procedure. Note, though, that on some data sets this could lead to slight loss of power when SNPs with a large effect are tested.

Use the command line option `-simLearnType Full` to perform "exact" LMM inference that avoids this potential loss of power by refitting the ratio δ of variances for every SNP tested.

Additionally, the number and coarseness of the search intervals can be adjusted via the command line options `-brentStarts <int>` for the number of intervals,

`-brentMinLogVal <double>` for the minimum of the search scope of log-δ values, and
`-brentMaxLogVal <double>`, for the maximum of the search scope of log-δ values.

By default the search is set conservatively to span 100 intervals over δ values between $\ln(-10)$ and $\ln(10)$.

Command line options

`-file basefilename`
: basename for PLINK's `.map` and `.ped` files

`-bfile basefilename`
: basename for PLINK's binary `.bed`, `.fam`, and `.bin` files

`-tfile basefilename`
: basename for PLINK's transposed `.tfam` and `.tped` files

`-dosage basefilename`
: basename for PLINK's `.dat`, `.fam`, and (optionally) `.map` files

`-pheno filename`
: name of phenotype file

`-mpheno index`
: index for phenotype in -pheno file to process, starting at 1 for the first phenotype column. Cannot be used together with `-pheno-name`. Default: 1.

`-pheno-name name`
: phenotype name for phenotype in `-pheno` file to process. If this option is used, the phenotype name must be specified in the header row. Cannot be used together with `-mpheno`.

`-fileSim basefilename`
: basename for PLINK's `.map` and `.ped` files for computing genetic similarity

`-bfileSim basefilename`
: basename for PLINK's binary `.bed`, `.fam`, and `.bin` files for building genetic similarity

`-tfileSim basefilename`
: basename for PLINK's transposed `.tfam` and `.tped` files for building genetic similarity

`-dosageSim basefilename`
: basename for PLINK's `.dat`, `.fam`, and (optionally) `.map` files for building genetic similarity

`-sim filename`
: specifies that genetic similarities are to be read directly from this file

`-simOut filename`
: specifies that genetic similarities are to be written to this file

`-linreg`
: specifies that linear regression will be performed. When this option is used, no genetic similarities should be specified.

`-covar filename`
: optional file containing the covariates

`-missingPhenotype <dbl>`
: identifier for missing values. If the phenotype for an individual is missing, then the individual is ignored. If a covariate value for an individual is missing, then it is mean imputed. Default: `-9`.

`-out filename`
: the name of the output file. Default value is `[basefilename].out.txt`

`-simLearnType [Full/Once]`
: if set to `Once` (the default), then delta, the ratio of residual to genetic covariance, is optimized only for the null model and used for each alternate model. If set to `Full`, then the ratio is re-estimated for each alternative model.

`-simType [RRM/COVARIANCE]`
: if set to `RRM` (the default), then the RRM is used for genetic similarity. If set to `COVARIANCE`, then the empirical SNP covariance matrix is used.

`-ML`
: use maximum likelihood parameter learning (default is `ML` with the likelihood ratio test)

`-REML`
: use restricted maximum likelihood parameter learning (default `ML`). `REML` will automatically invoke the F-test.

`-Ftest`
: use F-test (with `ML` or `REML`).

`-brentStarts <int>`
: number of interval boundary points for optimization of delta (see Section 2.1 of the Supplemental Information). Default: `100`.

`-brentMaxIter <int>`
: maximum number of iterations per interval for the optimization of delta. Default: `1e5`.

`-brentMinLogVal <double>`
: lower interval threshold for (log) delta optimization. Default: `-10`.

`-brentMaxLogVal <double>`
: upper interval threshold for (log) delta optimization. Default: `10`.

`-brentTol <double>`
: convergence tolerance of Brent's method used to optimize delta. Default: `1e-6`.

`-runGwasType [RUN/NORUN]`
: run GWAS or exit after computing the spectral decomposition of the genetic similarity matrix. Use `NORUN`, to cache the spectral decomposition. This option, in combination with the next, is useful for parallelizing the tests of many SNPs. Default: `RUN`.

`-eigen [directoryname]`
: load the spectral decomposition object from the directory name. The computations leading to the spectral decomposition of the genetic similarity matrix are skipped (note that that SNP file specifying the genetic similarities must still be given).

`-eigenOut [directoryname]`
: save the spectral decomposition object to the directory name. Can be used with `-runGwasType` option.

`-numjobs <int>`

Partition the SNPS into <int> groups and run FaSTLMM on the partition specified by -thisjob.

`-thisjob <int>`

Specifies which partition of SNPS created by -numjobs to process with FaSTLMM.

`-extract filename`

This is a SNP filter option. FaSTLMM will only analyze the SNPs explicitly listed in the 'filename' (no header, one SNP per line, where the SNP is indicated by the rs# or snp identifier).

`-extractSim filename`

This is a genetic similarity SNP filter option. FaSTLMM will only use SNPs explicitly listed in the 'filename' for computing genetic similarity.

`-extractSimTopK filename <int>`

Similar to –extractSim, this is a genetic similarity SNP filter option. FaSTLMM will only use the first <int> SNPs explicitly listed in the 'filename' for computing genetic similarity.

`-verboseOut`

Enable a more detailed and verbose output file with more columns. (See output)

`-MaxThreads <int>`

The option is passed to the MKL math libraries to 'suggest' the level of parallelism to use. Assigning a number larger than the number of cores on your machine may cause the program to run slower. Assigning a number less than the number of cores on your machine may allow your computer to run FastLmmC without consuming all the CPU resources in different phases of the program. The MaxThreads option is currently ignored when using ACML math libraries.

References

[1] Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ & Sham PC (2007) PLINK: a toolset for whole-genome association and population-based linkage analysis. *American Journal of Human Genetics,* 81.

Revision History

Date	Author(s)	Description of Changes
12/2/2011	Heckerman, Davidson	Update for v1.04 add -dosage support
3/13/2012		Update for v1.08 document new output formats option -`verboseOut` document new similarity option -`extractSim`