User Manual

FaST-LMM

Factored Spectrally Transformed Linear Mixed Models

C++ Version 2.07

Microsoft Research

Nov 12, 2014

Introduction

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

This manual describes the C++ version. A newer Python version is available at https://github.com/MSRCompBio/fastlmm.

For help with the software, please contact fastlmm@microsoft.com.

Citing FaST-LMM

If you use FaST-LMM in any published work, please cite the manuscripts describing it:

Single-SNP testing:

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* **8**: 833-835, Oct 2011 (doi:10.1038/nmeth.1681).

C. Widmer, C. Lippert, O. Weissbrod, N. Fusi, C.M. Kadie, R.I. Davidson, J. Listgarten, and D. Heckerman. Further Improvements to Linear Mixed Models for Genome-Wide Association Studies. *Scientific Reports* **4**, 6874, Nov 2014 (doi:10.1038/srep06874).

SNP-set testing:

C. Lippert, J. Xiang, D. Horta, C. Widmer, C.M. Kadie, D. Heckerman, J. Listgarten. Greater Power and Computational Efficiency for Kernel-Based Association Testing of Sets of Genetic Variants. *Bioinformatics* 2014 (doi: 10.1093/bioinformatics/btu504).

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/Data/DemoData`	contains sample data and command script
`fastlmm/Doc`	contains project documentation
`fastlmm/Externals`	contains other code FaST-LMM 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[1]

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

Building for Windows

C++ of versions FaST-LMM are built with Visual Studio 2012 (VS).
Any version of VS (Express through Universal) is capable of building FaST-LMM.
If you do not already have a copy of Visual Studio, the Visual Studio 2012 Express edition can be freely downloaded from
http://www.microsoft.com/express/downloads

- For the C++ version
  FaST-LMMC 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

[1] When building the C++ version of FaST-LMM the build host machine name is captured and used in the program banner to help identify which version of the code is being run. If you build fastlmmc.exe and you do not want the machine name captured, you should modify the banner string in Splash.cpp to remove it .

With a math library installed, no additional libraries are required to compile the C++ version of FaST-LMM (`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 FaST-LMM for Linux

FaST-LMM 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

This version of FaST-LMM is designed for use with randomly ascertained data with Gaussian residuals. If you have case-control data with substantial ascertainment bias, you should first transform your phenotype(s) using LEAP[1], which is available at http://bioinfo.cs.technion.ac.il/LEAP/leap.zip. If you are analyzing continuous phenotypes with non-Gaussian residulas, you should first transform your phenotype(s) using Warped-LMM[2], available at https://github.com/MSRCompBio/warpedLMM.

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 the genetic similarity matrix (GSM), 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 quadratic in the number of individuals.

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/[3] for further details. Missing SNP values will be mean imputed.

FaST-LMM supports dosage files in PLINK formats 1 and 2. The files may be uncompressed or compressed (with a `.gz` extension). Use `-dfile1` or `-dfile2` followed by the file prefix to load test SNPs in format 1 or 2, respectively. Use `-dfile1Sim` or `-dfile2Sim` followed by the file prefix to load similarity SNPs in format 1 or 2, respectively. As described in the PLINK manual, the .map file is optional and will be loaded if present.

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.

A sample dataset is provided with this release in `tests\datasets\synth`. The data has 500 samples with 5000 SNPs, and was generated from a Balding-Nichols model with FST=0.05.

The head of the phenotype file is as follows:

```
cid0P0 cid0P0 0.4853395139922632
cid1P0 cid1P0 -0.2076984565752155
cid2P0 cid2P0 1.4909084058931985
cid3P0 cid3P0 -1.2128996652683697
cid4P0 cid4P0 0.4293203431508744
...
```

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

```
FID   IID   MyPheno
```

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 are nearly identical to phenotype files in form, but covariate files cannot have a header row.

Instead of SNP data from which genetic similarities are computed, the user may provide the GSM 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 IDs (family ID and individual ID separated by a space). The value in the top-left corner of the file should be `var`.

Running FaST-LMM

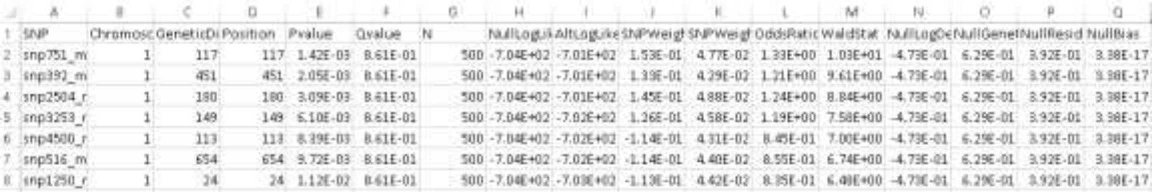
Once you have prepared the files in the proper format, you can run FaST-LMM. First move to the directory `tests\datasets\synth`, and then execute the command

```
C:\> fastlmmc -bfile chr1 -bfilesim allbutchr1 -pheno pheno_10_causals.txt -out
out.txt
```

You should see something similar to the following output on the screen:

```
FastLmmC v2.07.20140209 - Factored Spectrally Transformed Linear Mixed Models [Release]
  Copyright Microsoft Corporation -- Licensed Only for Non-Commercial use.
  Compiled Feb  9 2014 at 18:49:57 by BOBD01 for Windows
  using MKL v11.00.04 - Build: 20130517

 ++    Start Processing CommandLine:
 --      End Processing CommandLine:

 ++      Start Loading FastLmm Data:
  ++  Start Loading Covariance Data:
   ++      Processing PLINK fileset: [allbutchr1]
   ReadBinaryFiles4()* elapsed time:  90.194 ms
     Number of Individuals Selected:     500
               Number of Phenotypes:       1
                Number of SNPs Read:    4000
                Number of SNPs Used:    4000
   --  End Processing PLINK fileset: [allbutchr1]
  --    End Loading Covariance Data:
  ++        Start Loading Test Data:
   ++      Processing PLINK fileset: [chr1]
   ReadBinaryFiles4()* elapsed time:  46.689 ms
     Number of Individuals Selected:     500
               Number of Phenotypes:       1
                Number of SNPs Read:    1000
                Number of SNPs Used:    1000
   --  End Processing PLINK fileset: [chr1]
  --          End Loading Test Data:

 --        End Loading FastLmm Data:

              Compute/Load EigenSym:
Warning : The kernel has a few Eigenvalues that are a tiny bit smaller
           than zero and are considered to be numerically zero.
Warning : Make the kernel positive semi definite.
Warning : Setting negative eigenvalues to zero.
             Compute GWAS using LMM:
                  GWAS elapsed time: 189.102 ms
                  Write output file: [out.txt]
                 Total elapsed time:  1.190 sec
```

When the output file `(out.txt)` is loaded in Excel, it should look as follows:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	SNP	Chromosc	GeneticDi	Position	Pvalue	Qvalue	N	NullLogLi	AltLogLike	SNPWeigt	SNPWeigt	OddsRatic	WaldStat	NullLogDe	NullGenet	NullResid	NullBias
2	snp751_m	1	117	117	1.42E-03	8.61E-01	500	-7.04E+02	-7.01E+02	1.53E-01	4.77E-02	1.33E+00	1.03E+01	-4.73E-01	6.29E-01	3.92E-01	3.38E-17
3	snp392_m	1	451	451	2.05E-03	8.61E-01	500	-7.04E+02	-7.01E+02	1.39E-01	4.29E-02	1.21E+00	9.61E+00	-4.73E-01	6.29E-01	3.92E-01	3.38E-17
4	snp2504_r	1	180	180	3.09E-03	8.61E-01	500	-7.04E+02	-7.01E+02	1.45E-01	4.88E-02	1.24E+00	8.84E+00	-4.73E-01	6.29E-01	3.92E-01	3.38E-17
5	snp3253_r	1	149	149	6.10E-03	8.61E-01	500	-7.04E+02	-7.02E+02	1.26E-01	4.58E-02	1.19E+00	7.58E+00	-4.73E-01	6.29E-01	3.92E-01	3.38E-17
6	snp4500_r	1	113	113	8.39E-03	8.61E-01	500	-7.04E+02	-7.02E+02	-1.14E-01	4.31E-02	8.45E-01	7.00E+00	-4.73E-01	6.29E-01	3.92E-01	3.38E-17
7	snp516_m	1	654	654	9.72E-03	8.61E-01	500	-7.04E+02	-7.02E+02	-1.14E-01	4.40E-02	8.55E-01	6.74E+00	-4.73E-01	6.29E-01	3.92E-01	3.38E-17
8	snp1250_r	1	24	24	1.12E-02	8.61E-01	500	-7.04E+02	-7.03E+02	-1.13E-01	4.42E-02	8.35E-01	6.48E+00	-4.73E-01	6.29E-01	3.92E-01	3.38E-17

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

`OddsRatio`

The odds ratio of the SNP

`WaldStat`

The Wald statistic

`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

`Bias [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

SNP standardization

By default, each SNP is standardized to have mean zero and standard deviation one across all individuals. Another standardization method that can be used is to scale SNP values by multiplying by the beta(MAF,a,b) probability density function. This method is called using the `beta` flag, which takes a and b as parameters. If a and b are not specified, then a and b are set to 1 and 25, respectively, as recommended in ref[4]. If one parameter is specified, then both parameters must be specified.

Avoiding proximal contamination

When using a linear mixed model for association analysis, it is important to avoid proximal contamination[5]. To understand proximal contamination, first note that a LMM with no fixed effects, using a realized relationship matrix (RRM) for genetic similarities, is mathematically equivalent to linear regression of the SNPs on the phenotype, with weights integrated over independent Normal distributions having the same variance[6]. That is, a LMM using a given set of SNPs for genetic similarity is equivalent to a form of linear regression using those SNPs as covariates to correct for confounding. This equivalence implies that, when testing a given SNP, that SNP (and SNPs physically close to it) should be excluded from the computation of genetic similarity. If not, when testing a particular SNP, we would also be using that same SNP as a covariate, making the log likelihood of the null model higher than it should be, thus leading to deflation of the test statistic and loss of power.

Excluding the SNP you are testing from the genetic similarity matrix and also those SNPs in close proximity to it in a naïve way is extremely computationally expensive. A computationally efficient approach for performing the exclusion is to use a similarity matrix computed from all but chromosome *i* when testing SNPs on chromosome *i*.[5] We call this approach leave out one chromosome (LOOC). The analysis just described does this.

Speed vs. accuracy considerations

The FaST-LMM analysis 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.

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

Use the command line option `-simLearnType Once` to gain a constant factor speedup. 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.

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 $\ln(-5)$ 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

-dfile1 basefilename

basename for PLINK's .dat, .fam, and (optionally) .map files, format=1

-dfile2 basefilename

basename for PLINK's .dat, .fam, and (optionally) .map files, format=2

-noDosageRangeCheck

disables range checking for dosage files. Default: false.

-beta <a b>

scales values for each SNP across individuals by dividing by beta(a,b). Default values for a and b are 1. If either is specified, both must be specified.

-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

-dfile1Sim basefilename

basename for PLINK's .dat, .fam, and (optionally) .map files for building genetic similarity, format=1

-dfile2Sim basefilename

basename for PLINK's .dat, .fam, and (optionally) .map files for building genetic similarity, format=2

-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.

-logreg

specifies that logistic 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`. If the extension `.csv` is used, then the output is comma separated. Otherwise, the output is tab separated.

`-simLearnType [Full/Once]`

if set to `Once`, 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` (the default), 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 `REML`)

`-REML`

use restricted maximum likelihood parameter learning. `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: `-5`.

`-brentMaxLogVal <double>`

upper interval threshold for (log) delta optimization. Default: `5`.

`-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 FaST-LMM on the partition specified by -thisjob.

`-thisJob <int>`

specifies which partition of SNPS created by -`numjobs` to process for this run of FaST-LMM.

`-extract filename`

this is a SNP filter option. FaST-LMM 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. FaST-LMM 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. FaST-LMM 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)

`-setOutputPrecision <int>`

FastLmmC uses doubles for computation and has a default output precision of 16 digits after the decimal point. When working with large numbers of SNPs, writing full 16 digit precision can produce output that is quite large and this output precision may not be necessary. You can reduce the digits written in the output and reduce the file size using the `-SetOutputPrecision <int>` option. The parameter is restricted to the range `3 <= <int> <= 18`.

`-pValuePrintThreshold <dbl>`

this option sets a threshold filter to restrict the report output to include only those SNPs that have a p-value less than the specified value `<dbl>`. When large datasets are used, PvaluePrintThreshold produces smaller and more manageable output files.

The parameter is restricted to the range `0.0 < <dbl> <= 1.0`.

`-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.

`-excludeByGeneticDistance <dbl>`

excludes the SNP tested and those within this distance from the genetic similarity matrix. To use this feature genetic distances must be included in the PLINK files specifying both the test SNPs and the SNPs used for genetic similarity. The SNPs must be in non-decreasing order according to chromosome and the distance used.

`-excludeByPosition <int>`

excludes the SNP tested and those within this distance from the genetic similarity matrix. To use this feature positions must be included in the PLINK files specifying both the test SNPs and the SNPs used for genetic similarity. The SNPs must be in non-decreasing order according to chromosome and the distance used.

`-memoryFraction <dbl>`

specifies the fraction of memory to use when invoking `-topKByLinReg`. Default is 0.2.

References

1. Weissbrod, O., Lippert, C., Geiger, D. & Heckerman, D. Accurate liability estimation substantially improves power is ascertained case control studies. *arXiv* (2014).

2. Fusi, N., Lippert, C., Lawrence, N. D. & Stegle, O. Warped linear mixed models for the genetic analysis of transformed phenotypes. *Nat. Commun.* **5,** 4890 (2014).

3. Purcell, S. *et al.* PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **81,** 559–75 (2007).

4. Wu, M. C. *et al.* Rare-variant association testing for sequencing data with the sequence kernel association test. *Am. J. Hum. Genet.* **89,** 82–93 (2011).

5. Lippert, C. *et al.* FaST linear mixed models for genome-wide association studies. *Nat. Methods* **8,** 833–5 (2011).

6. Hayes, B. J., Visscher, P. M. & Goddard, M. E. Increased accuracy of artificial selection by using the realized relationship matrix. *Genet. Res. (Camb).* **91,** 47–60 (2009).

7. Lippert, C. *et al.* An exhaustive epistatic SNP association analysis on expanded Wellcome Trust data. *Sci. Rep.* **3,** 1099 (2013).