A short manual for LFMM version 1.4
(command-line version)

Eric Frichot (eric.frichot@imag.fr)
Olivier François (olivier.francois@imag.fr)

July 24, 2015

Please, print this reference manual only if it is necessary.

This short manual aims to help users to run the LFMM command-line engine on Mac and Linux operating systems.

Contents

1 Description

LFMM is a computer program for testing associations between loci and environmental gradients using latent factor mixed models [3, 5]. LFMM implements an MCMC algorithm for regression analysis in which the confounding variables are modeled with unobserved (latent) factors. The program estimates correlations between environmental variables and allelic frequencies, and simultaneously. infers the background levels of population structure. LFMM can also be run from the `R` command line using the `R` package LEA. A detailed description of the methods is available in

- Frichot E, Schoville SD, Bouchard G, François O, 2013. Testing for associations between loci and environmental gradients using latent factor mixed models. *Molecular Biology and Evolution*, 30 (7), 1687-1699.

- Frichot E, François O, 2015. LEA: An R package for landscape and ecological association studies. *In press.*

2 Installation

The installation files install C programs and R scripts to convert data to the LFMM format and to perform statistical analyses. To install the LFMM Command Line version, execute the install script (install.sh) from the LFMM main directory. Open a terminal shell, and type "./install.sh". If the shell script is not executable, try typing "chmod +x install.sh" and then re-run "./install.sh". A binary file named `LFMM` will be created in the "./bin/" directory.

3 Data format

The input files for LFMM consist of two files: a multilocus genotype matrix in the **lfmm** format and an environmental variable matrix in the **env** format.

The **lfmm** format for the **genotype matrix** consists of one row for each individual. Each row contains the genotypic values for each locus (separated by spaces or tabulations). These values represent the number of alleles for each SNP. The number of alleles can be the number of reference alleles or the number of derived alleles as long as a same choice is made for an entire column. Missing genotypes are encoded with the value 9 (-9 is also accepted). Here is an example of a data set in the lfmm format for 3 diploid individuals genotype at 4 SNPs

```
1 0 0 1
1 1 9 2
2 0 1 1
```

The **lfmm** format can be used to analyze any type of allelic markers (SNP, ALFP, microsatellite) that are encoded as thabsence/presence of an allele. It can also be used for individual or population data. In the case of population allele frequency data, each population corresponds to a row in the genotype matrix, and the number of alleles is replaced by the allele frequency for each genetic marker.
The **env** format for the **environmental variable matrix** consists of one row for each individual. Each row contains one value for each environmental variable, and the values are separated by spaces or tabulations. Here is an example of an environmental variable file for $n = 3$ individuals and two environmental variables. Caution: For more than one variable, the program can be launched for each variable sequentially (Default option) or with all variables simultaneously (see command-line options).

```
0.252 2.532
0.216 17.36
-0.47 6.971
```

The output format for the results is the **zscore** format. A **zscore** matrix has one row for each marker. Each row contains three values: The first value is the z-score test statistic, the second value is the -log10(p-value) for this statistic, and the third value is the p-value. The values are separated by spaces or tabulations. Here is an example of an output file for $L = 4$ genetic markers.

```
0.698819 0.314558 0.484665
1.35961 0.759568 0.173953
0.771135 0.355929 0.440627
0.879092 0.420959 0.379351
```

4 Running the LFMM program

The LFMM program can be executed by a command line. The basic command is as follows

```
./bin/LFMM -x genotype_file -v variable_file -K latent_factors_number
```

The xvK options are mandatory, and they can be specified in any order. Here is a description of the xvK options

- `-x genotype_file` is the path for a genotype matrix in the **lfmm** format.
- `-v variable_file` is the path for an environmental variable file in the **env** format.

- `-K latent_factor_number` is the number of latent factors. Several methods to choose K are described in the tutorial section. Our preferred method is to choose K based on the cross-entropy criterion of sNMF [4].

LFMM has additional commands that are all optional

- `-d nd` runs `LFMM` with the nd-th variable from the env file. By default (if NULL and `-a` FALSE), `LFMM` runs separate analyses using each variable from the env file sequentially.

- `-a` is a boolean value. If set to TRUE, `LFMM` runs with all variables from the env file simultaneously. This option is not compatible with the previous option. The default value is `-a` FALSE.

- `-o output_file` is a path for recording the output files (in the **zscore** format). There is one output file for each environmental variable. By default, the output files have the same base name (prefix) as the input file, and the prefix is extended with the label of the environmental variable (using ”a” for the all option and ”s” otherwise), the number of latent factors and the **.zscore** suffix. For example, for an analysis using the 3rd variable and 3 latent factors, the output file name will ”input_file_s3.3.zscore”.

- `-m` is a boolean value. If set to TRUE, the input file can contain missing genotypes. By default the program assumes that there are no missing data. The program is a slightly slower with missing data. Do not use this option if it is not necessary or impute the missing genotypes prior to the LFMM runs.

- `-p p` is the number of CPU processes used by the multi-threated version of the program. The number of processes has to be lower or equal than the number of physical processes available on your computer. By default, the number of process is 1.

- `-i iteration_number` is the number of cycles in the Gibbs Sampler algorithm. This number should be large enough (default: 10,000).

- `-b burnin_number` is the number of burnin cycles in the Gibbs Sampler algorithm (default: 5,000).

- `-s seed` is the seed to initialize the random generator. By default, the seed is random.

- `-C dev_file` is the path for a file containing the deviance information criterion and the calue of the expected deviance. By default, the name of the output file is the name of the input file with the .dic extension.

To obtain a summary of all options, use the `-h` option by typing the following command

```
./LFMM -h
```

A full example is available at the end of this note.

5 Using LFMM in practice

In this section, we give practical recommendations for analyzing real data sets using the `LFMM` computer program. These recommendations should help users avoiding important mistakes when using the `LFMM` algorithm. Note that the following comments should not be taken too literally. Several alternative options might work equally well.

Preparing the gentoypic data. Genotypic data must be prepared using the `lfmm` format (any type of allelic data is allowed). The `LFMM` program can handle missing data, but the algorithm used for genotype imputation slows the program. We encourage users having more than 10% missing genotypes in their data to impute missing values by using *matrix completion* or *genotype imputation* programs such as `IMPUTE2` or `MENDEL-IMPUTE` before starting their analyses with `LFMM`. Generally low quality data should be filtered out prior to the analysis. We also recommend **filtering out rare variants and retain only markers with MAF** $> 3-5\%$.

Preparing the environmental data. Ecological data must be prepared using the `env` format. To decide which variables should be used among a large number of ecological indicators (eg, climatic variables), we suggest that users summarize their environmental data using linear combinations of those indicators. Considering principal component analysis or similar approaches and using the first components as new ecological variables is one of these approaches.

Setting the run parameters. The LFMM program is based on a stochastic algorithm (MCMC) which cannot provide exact results. We recommend using large number of cycles (e.g., `-i 10000`) and the burn-in period should set at least to one-half of the total number of cycles (`-b 5000`). We have noticed that the program results are sensitive to the run-length parameter when data sets have relatively small sizes (eg, a few hundreds of individuals, a few thousands of loci). We recommend increasing the burn-in period and the total number of cycles in this situation.

Deciding the number of latent factors. Deciding an appropriate value for the number of latent factors in LFMM can be based on the the **results of ancestry estimation programs** and from the **analysis of histograms of test *p*-values**. Here, the objective is to control the false discovery rate (FDR) while keeping reasonable power to reject the null hypothesis. To choose K, a careful analysis of population structure and estimates of the number of ancestral populations contributing to the genetic data indicates the range of values to be explored. For example if the sNMF programs estimate 5 ancestral populations, then running LFMM with $K = 5$ or $K = 4 - 7$ often provides reasonable results (having inflation factors close to 1.0). To evaluate inflation in the test statistic, we suggest using the genomic inflation factor. According to Devlin and Roeder (1999), this quantity is defined as

$$\lambda = \text{median}(z^2)/0.456\,.$$

The inflation factor usually decreases with increasing values of K. To compute the genomic inflation factor, we recommend using several runs and taking the median of the λ values obtained from the above formula (use 5 to 10 runs, see our script below). Choosing values of K for which the estimate of λ is close to (or slightly below) 1.0 warrants that the FDR can be controlled efficiently (see below).

Combining *z*-scores obtained from multiple runs. We recommend to combine *z*-scores from multiple runs using the Fisher-Stouffer method or a similar method. In practice, we found that using the median *z*-scores of 5-10 runs and re-adjusting the *p*-values afterwards increase the power of the LFMM test statistic. Implementing this method be achieved by using the following sequence of R commands. Assuming that results from 5 runs with a particular value of K are recorded in external files named `res1_s1.9.zscore`, `res2_s1.9.zscore`, etc, `res5_s1.9.zscore`, we can compute adjusted *p*-values using the the following commands.

```
z.table = NULL
for (i in 1:5){
        file.name = paste("res",i, "_s1.9.zscore", sep="")
        z.table = cbind(z.table, read.table(file.name)[,1])
}
z.score = apply(z.table, MARGIN = 1, median) #combines z-scores
lambda = median(z.score^2)/0.456
adjusted.p.values = pchisq(z.score^2/lambda, df = 1, lower = F) #re-adjust p-values
hist(adjusted.p.values, col = 3)
```

For an expected value of the FDR equal to $q = 10\%$, a list of candidate loci can be obtained by using the Benjamini-Hochberg procedure as follows. The list is stored in the R object `candidates`.

```
q = 0.1
L = length(adjusted.p.values)
w = which(sort(adjusted.p.values) < q * (1:L) / L)
candidates = order(adjusted.p.values)[w]
```

Checking that confounding effects are under control. Obtaining precise values for K or for λ is not the main point of the LFMM analysis. The most important point is to end with a set of correct *p*-values, showing that confounding effects are under control. Correct histograms of *p*-values look flat with a peak close to zero (Figure 1). We strongly recommend **displaying histograms** of *p*-values or **qq-plots** of -log10 (*p*-values)). Since the classical definition of the genomic inflation factor leads to overly conservative tests, use the histogram to modify the λ value in order to get better results.

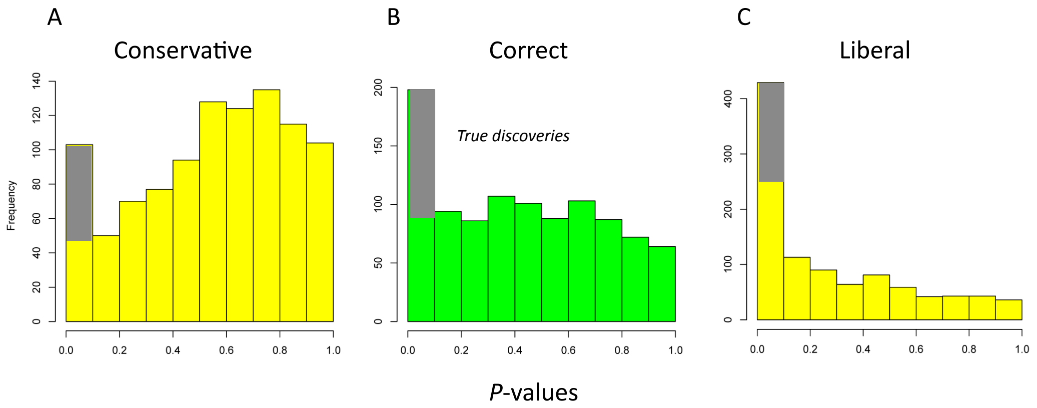

Figure 1: Histograms of p-values. The target of the LFMM analysis is to end with an histogram having the same shape as in the central panel (B).

6 Tutorial

6.1 Simulated data set

The data set contains 165 individuals genotyped for 1,000 SNPs. The last 100 SNPs were simulated to be truly associated with an environmental gradient. This data set is available in the directory "examples/simulated_example/".

An ancestry analysis for this data set suggests that ≈ 9 ancestral populations could explain the data (but, this is not an accurate estimate). Here is a shell script to launch 5 LFMM runs with $K = 9$.

```
mkdir res;
for i in 1 2 3 4 5; do
        ./bin/LFMM -x examples/simulated_example/genotypes.lfmm \
        -v examples/simulated_example/gradients.env -K 9 -o res/res$i
done
```

Then, open R and run the following R commands

```
z.table = NULL
for (i in 1:5){
        file.name = paste("res/res",i, "_s1.9.zscore", sep="")
        z.table = cbind(z.table, read.table(file.name)[,1])
}
z.score = apply(z.table, MARGIN = 1, median)
lambda = median(z.score^2)/0.456
adjusted.p.values = pchisq(z.score^2/lambda, df = 1, lower = F)
hist(adjusted.p.values, col = 3)

q = 0.1
L = length(adjusted.p.values)
w = which(sort(adjusted.p.values) < q * (1:L) / L)
```

```
candidates = order(adjusted.p.values)[w]

# estimated FDR and True Positif
estimated.FDR = length(which(candidates <= 900))/length(candidates)
estimated.TP = length(which(candidates > 900))/100
print(paste("expected FDR:", q))
print(paste("FDR:", estimated.FDR, "True Positive:", estimated.TP))
```

6.2 Real data set

6.2.1 Description

This data consist of a worldwide sample of human genomic DNA (10,757 SNPs) from 388 Asian individuals, taken from the Harvard Human Genome Diversity Project - Centre Etude Polymorphism Humain (Harvard HGDP-CEPH). In those data, each marker has been ascertained in samples of Mongolian ancestry (referenced population HGDP01224) [7]. We selected all population samples from Asia. Using Tracy-Widom tests implemented in SmartPCA [6], we found that the number of principal components with P-values smaller than 0.01 was around 10. Using the Bayesian clustering programs STRUCTURE [8] and TESS [1, 2], we found that $K = 7$ components could better describe the data.

We extracted climatic data population samples using the WorldClim data set at 30 arcsecond (1km2) resolution (Hijmans, Cameron, Parra, Jones, and Jarvis (2005)). We summarized the climatic variables by using the first axis of a principal component analysis for temperature variables and for precipitation variables. The data sets are available from the directory examples/human_example/. The genotypic information are in the input file panel11_Asia.lfmm. SNP information is available in panel11.pedsnp. The environmental file name is cov_panel11_Asia.env.

6.2.2 Running LFMM on the Asian data set

Here is a command line to analyze the data described above.

```
./LFMM -x examples/human_example/panel11_Asia.lfmm \
-v examples/human_example/cov_panel11_Asia.env  -K 7 -d 1
```

Output for LFMM

```
LFMM  Copyright (C) 2012 Eric Frichot
This program comes with ABSOLUTELY NO WARRANTY; for details type ’./LFMM -l’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ’./LFMM -l’ for details.

****                         LFMM Version 1.3                                   *****
****           E. Frichot, S. Schoville, G. Bouchard, O. Francois               *****
****                         Please cite our paper !                            *****
****   Information at http://membres-timc.imag.fr/Olivier.Francois/lfmm/        *****

./LFMM -x panel11_Asia.lfmm -v cov_panel11_Asia.env -K 7 -d 1
Summary of the options:

        -n (number of individuals)      388
        -L (number of loci)             10757
        -K (number of latent factors)   7
        -o (output file)                ./human_example/panel11_Asia
        -i (number of iterations)       10000
        -b (burnin)                     5000
        -s (seed random init)           4769701177742658440
        -p (number of processes (CPU))  1
        -x (genotype file)             human_example/panel11_Asia.lfmm
        -v (variable file)              human_example/cov_panel11_Asia.env
        -D (number of covariables)      2
        -d (the dth covariable)         1

```

```
Read variable file:
         examples/human_example/cov_panel11_Asia.env                OK.

Read genotype file:
         examples/human_example/panel11_Asia.lfmm                OK.

<<<<
         Analyse for variable 1

                Start of the Gibbs Sampler algorithm.

        [                                                                           ]
        [===========================================================================]

                End of the Gibbs Sampler algorithm.

        ED: 4173815.5         DIC: 4173900.09

        The statistics for the run are registered in:
                 examples/LFMM/human_example/panel11_Asia_s1.7.dic.

        The zscores for variable 1 are registered in:
                 examples/LFMM/human_example/panel11_Asia_s1.7.zscore.
        The columns are: zscores, -log10(p-values), p-values.

        -------------------------
        The execution for variable 1 worked without error.
>>>>
```

7 Additional programs

We provide additional C programs to run PCA and perform Tracy-Widom tests.

7.1 Principal Component Analysis

The program can be executed by the following command. The format is:

```
./bin/pca -x genotype_file
```

The x option is mandatory.

- `-x genotype_file` is the path for a genotype file in the **lfmm** format.

Other options:

- `-K K` computes K principal components only (default: $K = n$, the number of individuals).
- `-a eigenvalue_file` output eigenvalues file (default: genotype_file.eigenvalues).
- `-a eigenvector_file` output eigenvectors file (default: genotype_file.eigenvectors).
- `-a sdev_file` output standard deviation file (default: genotype_file.sdev).
- `-a projection_file` output projection file (default: genotype_file.projections).
- `-c` boolean value, data centered (default: FALSE).
- `-s` boolean value, data centered and scaled (default: FALSE).

For a summary of all options, type

```
./pca -h
```

Here is a small example:

```
./pca -x examples/simulated_example/genotypes.lfmm
```

A program is also available to perform Tracy-Widom tests on a set of eigenvalues. The program can be run as follows.

```
./tracyWidom -i input_file.eigenvalues [output_file.tracywidom]
```

where

- `input_file.eigenvalues` is the path for the input file containing the eigenvalues.
- `output_file.tracywidom` is the path for the output_file. By default, the name of the output file is the name of the input_file with a .tracywidom extension.

Here is a small example:

```
./tracyWidom -i examples/simulated_example/genotypes.eigenvalues
```

7.2 Other data format accepted

Input files consist of two mandatory files (a genotype file and an environmental variable file). It is not necessary to provide information about individuals. All data formats are described with the same example. These files are available in `examples/format_example/`.

- lfmm (example.lfmm)
  The **lfmm** format has one row for each individual. Each row contains one value per SNP (separated by spaces or tabulations): the number of alleles. The number of alleles can be the number of reference alleles or the number of derived alleles as long as it is the same choice for an entire SNP. The missing genotypes are encoded with the value 9.

  ```
  1 0 0 1
  1 1 9 2
  2 0 1 1
  ```

- ped (example.ped)
  The **ped** format has one row for each individual. Each row contains 6 columns of information for each individual, plus two genotype columns for each SNP. Each column must be separated by spaces or tabulations. The genotype format must be either 0ACGT or 01234, where 0 means missing genotype. The first 6 columns of the genotype file are: the 1st column is the family ID, the 2nd column is the sample ID, the 3rd and 4th columns are the sample IDs of parents, the 5th column is the gender (male is 1, female is 2), the 6th column is the case/control status (1 is control, 2 is case), the quantitative trait value or the population group label.

  The ped format is described here: http://pngu.mgh.harvard.edu/ purcell/plink/data.shtml.

  ```
  1       SAMPLE0 0 0 2 2 1 2 3 3 1 1 2 1
  2       SAMPLE1 0 0 1 2 2 1 1 3 0 4 1 1
  3       SAMPLE2 0 0 2 1 2 2 3 3 1 4 1 2
  ```

- ancestrymap (example.ancestrymap)
  The **ancestrymap** format has one row for each genotype. Each row has 3 columns: the 1st column is the SNP name, the 2nd column is the sample ID, the 3rd column is th number of alleles. It is assumed that the genotypes for a given SNP name are written in consecutive lines. It is also assumed that the genotypes for a set of individuals are given in the same order as lines. The number of alleles can be the number of reference alleles or the number of derived alleles as long as it is the same choice for an entire SNP. It is assumed that the missing genotypes are encoded with the value 9.

```
              rs0000            SAMPLE0   1
              rs0000            SAMPLE1   1
              rs0000            SAMPLE2   2
              rs1111            SAMPLE0   0
              rs1111            SAMPLE1   1
              rs1111            SAMPLE2   0
              rs2222            SAMPLE0   0
              rs2222            SAMPLE1   9
              rs2222            SAMPLE2   1
              rs3333            SAMPLE0   1
              rs3333            SAMPLE1   2
              rs3333            SAMPLE2   1
```

- geno (example.geno)
  The **geno** format has one row for each SNP. Each row contains 1 character per individual: 0 means zero copy of the reference allele. 1 means one copy of the reference allele. 2 means two copies of the reference allele. 9 means missing data.

```
112
010
091
121
```

- vcf (example.vcf)
  The **vcf** The vcf format is described here
  http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41

```
##fileformat=VCFv4.1
##FORMAT=<ID=GM,Number=1,Type=Integer,Description="Genotype meta">
##INFO=<ID=VM,Number=1,Type=Integer,Description="Variant meta">
##INFO=<ID=SM,Number=1,Type=Integer,Description="SampleVariant meta">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT SAMPLE0 SAMPLE1 SAMPLE2
1 1001 rs0000 T C 999 . VM=1;SM=100 GT:GM 1/0:1 0/1:2 1/1:3
1 1002 rs1111 G A 999 . VM=2;SM=101 GT:GM 0/0:6 0/1:7 0/0:8
1 1003 notres G AA 999 . VM=3;SM=102 GT:GM 0/0:11 ./.:12 0/1:13
1 1004 rs2222 G A 999 . VM=3;SM=102 GT:GM 0/0:11 . 1/0:13
1 1003 notres GA A 999 . VM=3;SM=102 GT:GM 0/0:11 ./.:12 0/1:13
1 1005 rs3333 G A 999 . VM=3;SM=102 GT:GM 1/0:11 1/1:12 0/1:13
```

7.3 Data conversion

The LFMM command-line engine uses data in the lfmm format. Files in ped, eigenstratgeno, ancestrymap can be converted to the lfmm format using C programs as follows.

The format of the command is (replace ¡format¿ by ped, ancestrymap, or geno):

```
./<format>2lfmm input_file [output_file]
```

where

- `input_file` is the path for the input file.
- `output_file` is the path for the output_file (in the lfmm format). By default, the name of the output file is the name of the input_file with the .lfmm extension.

For examples,

- example.ped

```
./ped2lfmm examples/format_example/example.ped
```

- example.ancestrymap

```
./ancestrymap2lfmm examples/format_example/example.ancestrymap
```

- example.geno

```
./geno2lfmm examples/format_example/example.geno
```

- example.vcf

```
./vcf2geno examples/format_example/example.vcf
```

7.4 Manhattan plot

To display a manhattan plot for the results, consider the R package `qqman`.

8 Contact

A FAQ (Frequently Asked Questions) section is available on our webpage (ttp://membres-timc.imag.fr/Olivier.Francois/lfmm. The LFMM software is still under development. All your comments and feedbacks are more than welcome.

References

[1] Chibiao Chen, Eric Durand, Florence Forbes, and Olivier François. Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. *Molecular Ecology Notes*, 7(5):747–756, 2007.

[2] E Durand, F Jay, O E Gaggiotti, and O François. Spatial inference of admixture proportions and secondary contact zones. *Molecular Biology and Evolution*, 26(9):1963–1973, 2009.

[3] Eric Frichot and Olivier François. Lea: an r package for landscape and ecological association studies. *Methods in Ecology and Evolution*, 2015.

[4] Eric Frichot, François Mathieu, Théo Trouillon, Guillaume Bouchard, and Olivier François. Fast and efficient estimation of individual ancestry coefficients. *Genetics*, 196(4):973–983, 2014.

[5] Eric Frichot, Sean D Schoville, Guillaume Bouchard, and Olivier François. Testing for associations between loci and environmental gradients using latent factor mixed models. *Molecular Biology and Evolution*, 30(7):1687–1699, 2013.

[6] N Patterson, A L Price, and D Reich. Population structure and eigenanalysis. *PLoS Genetics*, 2:20, 2006.

[7] Nick J. Patterson, Priya Moorjani, Yontao Luo, Swapan Mallick, Nadin Rohland, Yiping Zhan, Teri Genschoreck, Teresa Webster, and David Reich. Ancient admixture in human history. *Genetics, doi:10.1534/genetics.112.145037*, 2012.

[8] J K Pritchard, M Stephens, and P Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.