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

Eric Frichot
efrichot@gmail.com

September 12, 2013

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

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

# 1 Description

We proposed an integrated framework based on population genetics, ecological modeling and machine learning techniques for screening genomes for signatures of local adaptation. We implemented fast algorithms using a hierarchical Bayesian mixed model based on a variant of principal component analysis in which residual population structure is introduced via unobserved factors. These algorithms can detect correlations between environmental and genetic variation at the same time as they infer the background levels of population structure. A description of the method is available in our paper:

Eric Frichot, Sean Schoville, Guillaume Bouchard, Olivier François, 2013. *Landscape genomic tests for associations between loci and environmental gradients* Molecular Biology and Evolution, in press.

# 2 Installation

We provide a set of R and perl scripts convert to LFMM format and to display manhattan plot. By consequence, R and perl are mandatory to convert to LFMM format and to display manhattan plot. they are not mandatory to execute LFMM.
To install LFMM CL version, you just have to execute the install script (install.sh) in LFMM main directory. To execute it in a terminal shell, go to LFMM main directory and write "./install.sh". If the script is not executable, type "chmod +x install.sh" and then "./install.sh". A binary called `LFMM` should be created in LFMM main directory.

# 3 Data format

Input files are composed of two files: a genotype file and a variable file.

The **genotype file** is a SNP matrix of n lines for n individuals and L columns for L loci. Each element can be 0, 1 or 2. A missing element will be notify by the value 9 or -9. Each element of the matrix is separated by one or several spaces. There should be no space after the last value of each line. A line should not contain only missing data (9 or -9). Below, an example of genotype file for $n = 3$ individuals and $L = 5$ loci.

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

The **variable file** is a vector composed of n lines and D columns. Each line is the values of the D variables for the corresponding individual. Below, an example of variable file for $n = 3$ individuals and $D = 1$ covariable. Warning: If you set several covariables, the program will be launched for each covariable sequentially and independently.

```
0.252477
0.216618
-0.47509
```

The **output file** for 1 environmental variable is composed of 1 line for each SNP. Each line is composed of 3 columns. 1st column $z$-score, 2nd column is the $-log_{10}(pvalue)$, and 3rd column is the pvalue for the corresponding loci. Below, an example of output file for $L = 4$ loci.

```
0.0259401 0.00908548 0.979297
0.0616506 0.02191 0.950802
0.0210902 0.0073732 0.983166
0.00991587 0.00346154 0.992061
```

# 4 Run the programs

The program is executed by a command line. The format is:

```
./LFMM -g genotype_file -v variable_file -n individual_number -L loci_number -D variable_number
-K latent_factors_number -i iterations_number -b burnin_number -o output_file
```

All the previous options are mandatory. There is no order for the options in the command line. Here is a more precise description of the options:

- `-g genotype_file` is the path for the genotype file.
- `-v variable_file` is the path for the variable file.
- `-n individual_number` is the number of individuals in your input file.
- `-L loci_number` is the number of loci in your input file.
- `-D variable_number` is the number of covariable in your variable file. Warning: If you set several covariables, the program will be launched for each covariable sequentially and independently.
- `-K latent_factor_number` is the number of latent factors.
- `-i iteration_number` is the number of iterations in the Gibbs Sampling algorithm. This number should be large. One way to test if it is large enough is to check if the results are similar for several runs with this number of iterations.
- `-b burnin_number` is the number of burning iterations in the Gibbs Sampling algorithm. This number of iterations is included in the total number of iterations.
- `-o output_file` is the prefixe name of the output files. An output file will be created for each environmental variable. For example if you provided two variables, the results for variable 1 will be stored in `output_file_1` and the results for variable 2 will be stored in `output_file_2`.

Two options are not mandatory:

- `-m` indicates that some data are missing. By default the program assumes that there is no missing data. The program is a bit slower with missing data. Please, do not use this option if it is not necessary.
- `-p p` is the number of processes that you choose to use if you run the algorithm in parrallel. Be careful, the number of process has to be lower or equal than the number of physical processes available on your computer. By default, the number of process is 1.
- `-C dev_file` is the path to DIC output file. By default, it is "DIC.txt".
- `-d dth_variable`. If this option is set, LFMM is launched only with the d-th covariable of the environmental file.

If you need a summary of the options, you can use the `-h` option by typing the command line

```
./LFMM -h
```

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

**How to choose $K$, the number of latent factors** . The number of latent factors is the number of principal components (or latent factors) that is required to describe the neutral structure of the data. Several values should be tested. A too small value of $K$ leads to liberal tests and may generate False Positive results. A too large value of $K$ leads to conservative tests and may generate False Negative results. In our paper, we used the number of significative principal components in the Tracy-Widom test of `SmartPCA` (http://helix.nih.gov/Applications/eigensoft.html) [3]. This heuristic may be a bit too conservative. We also used the Bayesian clustering programs `STRUCTURE` (http://pritch.bsd.uchicago.edu/software/structure2_1.html) [5] and `TESS` (http://membres-timc.imag.fr/Olivier.Francois/tess.html) [1, 2] to find K the number of components which could better describe our simulated data. We advise you to be really careful in the choice of $K$ and to test several values of $K$.

# 5 Scripts

We provide a set of R and perl scripts convert to LFMM format and to plot manhattan plot.

## 5.1 Data Format

Input files are composed of two mandatory files (a genotype file and an environmental variable file) and one optional file (the snp information file). The snp file is interesting to analyze zscore results and display results with manhattan plots. It is not necessary to provide information about individuals. All data formats are described with the same example. These files are available in `example/format_example/`. Each file should end with its format name.

### 5.1.1 Genotype Data

- lfmm (example.lfmm)
  The genotype file is 1 line per individual. There is 1 genotype column for each SNP (in the order the SNPs are specified in the snp file). Each element can be 0, 1 or 2. A missing element is identified by the value 9 or $-9$. Each element of the matrix is separated by a single space. There should be no space after the last value of each line. Lines containing only missing data ($-9$ or 9) should be removed.

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

- ped (example.ped)
  The genotype file is 1 line per individual. Each line contains 6 columns of information about the individual, plus two genotype columns for each SNP in the order the SNPs are specified in the snp file. Genotype format must be either 0ACGT or 01234, where 0 means missing data. The first 6 columns of the genotype file are: 1st column is family ID, 2nd column is sample ID, 3rd and 4th column are sample IDs of parents, 5th column is gender (male is 1, female is 2), 6th column is case/control status (1 is control, 2 is case), quantitative trait value or population group label. In the two genotype columns for each SNP, missing data is represented by 0.

```
     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 genotype file contains 1 line per valid genotype. There are 3 columns: 1st column is SNP name, 2nd column is sample ID, 3rd column is number of reference alleles (0 or 1 or 2), Missing genotypes are encoded by the value $-9$ or 9 in the genotype file.

```
              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
```

- eigenstratgeno (example.eigenstratgeno)
  The genotype file contains 1 line per SNP. Each line contains 1 character per individual: 0 means zero copies of reference allele. 1 means one copy of reference allele. 2 means two copies of reference allele. 9 means missing data.

```
112
010
091
121
```

*Tips:* As LFMM does not model allele frequencies, genotype file can be the number of copy of either the reference allele or the derived allele.

### 5.1.2 Snp Data

Warning: SNP data information has to be in the same order as in genotypic data file.

- pedsnp (example.pedsnp or example.map)
  The snp file contains 1 line per SNP. There are 6 columns (last 2 optional): 1st column is chromosome. Use X for X chromosome, 2nd column is SNP name, 3rd column is genetic position (in Morgans), 4th column is physical position (in bases), Optional 5th and 6th columns are reference and variant alleles.

```
11      rs0000     0.000000           0 A C
11      rs1111     0.001000      100000 A G
11      rs2222     0.002000      200000 A T
```

- snp (example.snp)
  The snp file contains 1 line per SNP. There are 6 columns (last 2 optional): 1st column is SNP name, 2nd column is chromosome. Use X for X chromosome, 2nd column is SNP name, 3rd column is genetic position (in Morgans) (If unknown, ok to set to 0.0), 4th column is physical position (in bases), Optional 5th and 6th columns are reference and variant alleles.

```
rs0000  11        0.000000              0 A C
rs1111  11        0.001000         100000 A G
rs2222  11        0.002000         200000 A T
```

- lfmmsnp (example.lfmmsnp)
  The snp file contains 1 line per SNP. There are 3 columns: 1st column is SNP name, 2nd column is chromosome, 3th column is physical position (in bases).

```
rs0000  11        0
rs1111  11        100000
rs2222  11        200000
```

*Tips:* SNP data information is not mandatory. But if you have it, we advise you to provide it. It is useful for post-treatment of LFMM analysis.

## 5.2 Data conversion

The LFMM command-line engine allows data in lfmm format. You can convert from ped, eigenstratgeno, ancestrymap to lfmm format using perl scripts. The format is ($n$ is the number of individuals, $L$ the number of loci):

- for example.ped

```
perl ./scripts/ped2lfmm.pl data/example.ped example.lfmm L n
```

- for example.ancestrymap

```
perl ./scripts/ancestrymap2lfmm.pl data/example.ancestrymap example.lfmm L n
```

- for example.eigenstratgeno or example.geno

```
perl ./scripts/eigenstratgeno2lfmm.pl data/example.eigenstratgeno example.lfmm L n
```

### 5.3 Transform LFMM results

The LFMM command line engine outputs results without taking into account snp data informations. You can add these informations and display results in a table similar as the one in the GUI by using perl scripts. The output format will be called .res. The format is ($L$ the number of loci):

- without snp data

```
perl ./scripts/nothing2lfmm.pl zscore.txt zscore.res L
```

- for example.pedsnp

```
perl ./scripts/pedsnp2lfmm.pl zscore.txt example.pedsnp zscore.res L
```

- for example.snp

```
perl ./scripts/snp2lfmm.pl zscore.txt example.snp zscore.res L
```

- for example.lfmmsnp

```
perl ./scripts/lfmmsnp2lfmm.pl zscore.txt example.lfmmsnp zscore.res L
```

### 5.4 Manhattan plot

You can create a manhattan plot in the pdf format using a provided R script. If you want, you can highlight a list of specific snps in green. The list of SNPs you want to display should be written in file `manhattan_table.txt`. The list of SNPs you want to highlight in green should be written in file `toHighlight_table.txt`.

```
Rscript scripts/manhattan.R manhattan_table.txt toHighlight_table.txt manhattan_plot.pdf
```

The format of these files is the same as the export format of a zscore table (`zscore.res`). Here is an example of format:

```
Name        Chr        Position        Zscore        -log10(p-value) p-value
rs0000        11        0        0.070834        0.0252502        0.943517
rs1111        11        100000        0.0534096        0.0189187        0.957373
rs2222        11        200000        0.0126014        0.00440559        0.989907
rs3333        11        300000        0.0261071        0.00915623        0.979138
rs4444        11        400000        0.0181521        0.00635249        0.985479
rs5555        11        500000        0.00728521        0.00253695        0.994175
rs6666        11        600000        0.00500635        0.00175344        0.995971
```

## 6 Tutorial

### 6.1 Data set

The data set that we analyze in this tutorial is an Asian human data set of SNPs data. This data is a worldwide sample of genomic DNA (10757 SNPs) from 388 individuals, taken from the Harvard Human Genome Diversity Project - Centre Etude Polymorphism Humain (Harvard HGDP-CEPH)2 . In those data, each marker has been ascertained in samples of Mongolian ancestry (referenced population HGDP01224) [4]. We selected all samples from Asia. Using Tracy-Widom tests implemented in `SmartPCA` [3], we found that the number of principal components with P-values smaller than 0.01 was around $KTW = 10$. Using the Bayesian clustering programs `STRUCTURE` [5] and `TESS` [1, 2], we found that $K = 7$ components could better describe our simulated 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 set is in directory `examples/human_example/`. The genotypic information are in `panel11_Asia.lfmm`. the SNPs information is in `panel11.pedsnp`. The environmental file is `cov_panel11_Asia.env`. There are 2 variables, one proxy for temperature and one for precipitation.

### 6.2 Run LFMM

Here is an example of command line to analyse our dataset

```
./LFMM -g examples/human_example/panel11_Asia.lfmm -v examples/human_example/cov_panel11_Asia.env
-n 388 -L 10757 -D 2 -K 7 -i 1200 -b 200 -o examples/human_example/zscore -p 1
```

Here is an example of command line to create `zscore.res` for the second variable.

```
perl ./scripts/pedsnp2zscore.pl examples/human_example/zscore_2 examples/human_example/panel11.pedsnp
examples/human_example/zscore_2.res 10757
```

Then, we created a file `examples/human_example/toHiglight_table.txt` as follows:

```
Name    Chr     Position        Zscore  -log10(p-value) p-value
Affx-3561055 10 67583134 4.59106 5.35564 4.40916E-06
Affx-3582668 10 69220278 3.97663 4.15557 6.98929E-05
```

Here is an example of command line to create a manhattan plot (in `examples/human_example/manhattan_plot_2.pdf`) with all SNPs and the two previous SNPs highlighted in green:

```
Rscript scripts/manhattan.R examples/human_example/zscore_2.res
examples/human_example/toHighlight_table.txt examples/human_example/manhattan_plot_2.pdf
```

## 7 Contact

If you need assistance, do not hesitate to send me an email (efrichot@gmail.com). A FAQ (Frequently Asked Questions) section is available on our webpage (ttp://membres-timc.imag.fr/Olivier.Francois/lfmm.html). 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] N Patterson, A L Price, and D Reich. Population structure and eigenanalysis. *PLoS Genetics*, 2:20, 2006.

[4] 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.

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