A short manual for sNMF
(command-line version)

Eric Frichot
efrichot@gmail.com

February 5, 2014

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

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

1   Description

Inference of individual ancestry coefficients, which is important for population genetic and association studies, is commonly performed using computer-intensive likelihood algorithms. With the availability of large population genomic data sets, fast versions of likelihood algorithms have attracted considerable attention. Reducing the computational burden of estimation algorithms remains, however, a major challenge. Here, we present a fast and efficient method for estimating individual ancestry coefficients based on sparse non-negative matrix factorization algorithms. We implemented our method in the computer program sNMF, and applied it to human and plant data sets. The performances of sNMF were then compared to the likelihood algorithm implemented in the computer program ADMIXTURE. Without loss of accuracy, sNMF computed estimates of ancestry coefficients with run-times approximately 10 to 30 times shorter than those of ADMIXTURE.

[1] Eric Frichot, François Mathieu, Théo Trouillon, Guillaume Bouchard, Olivier François. *Fast Inference of Individual Ancestry Coefficients Using Non-negative Matrix Factorization Algorithms*, submitted.

2   Installation

To install sNMF command-line version, unzip the sNMF_CL.zip file, and run the install script (install.command) from the sNMF directory. From a terminal shell, go to the sNMF main directory and type "./install.command". If the script is not executable, type "chmod +x install.command" and then "./install.command". A set of binaries should be created in the sNMF directory.

3   Data format

3.1   input file

The sNMF input file consists of a single genotype file in the **geno** format.

- **geno** (example.geno)

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

  Below, an example of geno file for $n = 3$ individuals and $L = 5$ loci.

  ```
  112
  010
  091
  121
  ```

sNMF also propose C functions to convert from the following formats to the geno format.

- **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. 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 columns 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. The ped format is also described here.

  Below, an example of ped file for $n = 3$ individuals and $L = 5$ loci.

  ```
  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
  ```

  The format of the command line is:

  ```
  ./bin/ped2geno  input_file [output_file]
  ```

  where

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

- **ancestrymap** (example.ancestrymap)

  The **ancestrymap** format has one row for each genotype. Each row has 3 columns: 1st column is SNP name, 2nd column is sample ID, 3rd column is 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 missing genotypes are encoded by the value 9.

  Below, an example of ancestrymap file for $n = 3$ individuals and $L = 5$ loci.

  ```
  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   SAMPLE3 1
  ```

  The format of the command line is:

  ```
  ./bin/ancestrymap2geno  input_file [output_file]
  ```

  where

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

- **vcf** (example.vcf)

  The **vcf** format is described here.

  Below, an example of vcf file for $n = 3$ individuals and $L = 5$ loci.

```
##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
```

The format of the command line is:

```
./bin/vcf2geno  input_file [output_file]
```

where

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

- **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. Missing genotypes are encoded by the value 9 or -9.

  Below, an example of lfmm file for $n = 3$ individuals and $L = 5$ loci.

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

The format of the command line is:

```
./bin/lfmm2geno  input_file [output_file]
```

where

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

3.2   output files

There are 2 **output files**.

- The first file (with extension **.Q**) contains individual admixture coefficients. It is a matrix with n rows (the number of individuals) and K columns (the number of ancestral populations).

- The second file (with extension **.G**) contains the ancestral genotype frequencies. It is a matrix with $n_a \times L$ lines (the number of alleles times the number of SNPs) and K columns (the number of ancestral populations). For each SNP, the first row contains the ancestral frequencies for allele 0, the second line for allele 1, ...

4   Run the program

The program can be executed from a command line. The format is:

```
./bin/sNMF -x genotype_file.geno -K number_of_ancestral_populations
```

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

- `-x genotype_file.geno` is the path for the genotype file (in .geno format).

- `-K number_of_ancestral_populations` is the number of ancestral populations.

Additional options are available:

- `-a alpha` is the value of the regularization parameter (by default: 0). Results can depend on the value of this parameter, especially for small data sets.

- `-q output_Q` is the path for the output file with ancestry coefficients. By default, the name of the output file is the name of the input file with extension .K.Q.

- `-g output_G` is the path for the output file with ancestral genotype frequencies. By default, the name of the output file is the name of the input file with extension .K.G.

- `-c perc` percentage of masked genotypes. If this option is set, the cross-entropy criterion is calculated (see section Cross-Entropy criterion). If the option is set without percentage, the default percentage is 0.05.

- `-e tolerance` is the tolerance error (by default: 0.0001).

- `-i iteration_number` is the max number of iterations of the algorithm (default: 200).

- `-I nb_SNPs` starts the algorithm with a run of sNMF using a subset of nb_SNPs random SNPs. If this option is set with nb_SNPs, the number of randomly chosen SNPs is the minimum between 10000 and 10 % of all SNPs. The option can considerably speeds up sNMF estimation for very large data sets.

- `-Q input_Q` is the path for an initialisation file for Q, the individual admixture coefficients. If both `-I` and `-Q` are set, `-Q` is chosen.

- `-s seed` is the initialization for the random parameter (by default: random).

- `-m ploidy` 1 if haploid, 2 if diploid (default: 2).

- `-p p` is the number of CPU that you choose to use if you run the algorithm in a parallel computer. Be aware that the number of processes has to be lower or equal than the number of physical processes (CPU units) available on your computer (default: 1).

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

```
./bin/sNMF -h
```

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

5   Cross-Entropy criterion

Goal   The cross-entropy criterion is a technique based on the prediction of masked genotypes to evaluate the error of ancestry estimation. This criterion will help to choose the best number of ancestral population (K) and the best run among a set of runs. A smaller value of cross-entropy means a better run. An example is displayed in the tutorial section.

programs   The cross-Entropy criterion of a single `sMNF` run can be calculated using the `-c` option in `sNMF`. We also provide two programs that compute the cross-entropy criterion for the data separately.

- A first program creates a data set with a given percentage of masked data from your original data set. The command line format is:

```
./bin/createDataSet -x genotype_file.geno
```

The mandatory option is:

– `-x genotype_file.geno` is the path for the genotype file (in .geno format).

It will create a file with around 5 % of masked data with the name genotype_file_**I**.geno. _**I** extension differentiates this file from the original file.

Other options (not mandatory):

– `-r percentage` is the percentage of masked data in your data set (default: 0.05).

4

- – `-e tolerance` is the tolerance error (by default: 0.0001).
- – `-s seed` is the initialization for the random parameter (by default: random).
- – `-m ploidy` is 1 if haploid, 2 if diploid (default: 2).

- A second program calculates the cross-entropy criterion for all data and for the masked data from the output of `sNMF`. The cross-entropy criterion is useful to choose the best run for numbers of ancestral distinct populations (K) and different values of the regularization parameter (α). A smaller value of cross-entropy with missing data means a better prediction of the masked data. The command line format is:

```
./bin/crossEntropy -x genotype_file.geno -K number_of_ancestral_populations
```

The mandatory options are:

- – `-x genotype_file.geno` is the path for the genotype file (in .geno format).
- – `-K number_of_ancestral_populations` is the number of ancestral populations.

In this case, the output from `sNMF`, the files with masked data, the original files and the results files are stored in the same directory.

Other option (not mandatory):

- – `-m ploidy` 1 if haploid, 2 if diploid (default: 2).

To summarize, the two following sets of command-lines will both calculate the cross-entropy criterion.

```
./bin/sNMF -x genotype_file.geno -K K -c
```

```
./bin/createDataSet -x genotype_file.geno
./bin/sNMF -x genotype_file_I.geno -K K
./bin/crossEntropy -x genotype_file.geno -K K
```

6 Tutorial

6.1 Data set

The data set that we analyze in this tutorial is an Asian human data set. This data is a worldwide sample of genomic DNA (10757 SNPs) from 934 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) [2].

6.2 Example

Create a data set with masked data   In the main directory, type:

```
./bin/createDataSet -x examples/panel11.geno
```

output for createDataSet

```
summary of the options:

     -n (number of individuals)           934
     -L (number of loci)                  10757
     -s (seed random init)                11162993670188721480
     -r (percentage of masked data)       0.05
     -x (genotype file)                   examples/panel11.geno
     -o (output file)                     examples/panel11_I.geno
     - diploid

Write genotype file with masked data examples/panel11_I.geno:            OK.
```

A file with 5 % of masked data with path `examples/panel11_I.geno` has been created.

Run sNMF   Then, run sNMF for the data set with 5 % of masked data (with $K = 5$ for example):

```
./bin/sNMF -x examples/panel11_I.geno -K 5
```

output for sNMF

```
./bin/sNMF -x examples/panel11_I.geno -K 5
summary of the options:

        -n (number of individuals)       934
        -L (number of loci)              10757
        -K (number of ancestral pops)    5
        -x (input file)                  examples/panel11_I.geno
        -q (individual admixture file)   examples/panel11_I.5.Q
        -g (ancestral frequencies file)  examples/panel11_I.5.G
        -i (number max of iterations)    200
        -a (regularization parameter)    0
        -s (seed random init)            11162857829069388553
        -e (tolerance error)             0.0001
        -p (number of processes)         1
        - diploid

Read genotype file examples/panel11_I.geno:               OK.


Main algorithm:
[                                                                        ]
[=================================================]
Number of iterations: 142


Least-square error: 4597146.245722
Write individual ancestry coefficient file examples/panel11_I.5.Q:          OK.
Write ancestral allele frequency coefficient file examples/panel11_I.5.G:   OK.
```

The results files `examples/panel11_I.Q examples/panel11_I.G` have been created. `sNMF` also dislpays the number of iterations and the least-squares error: $||X - QG||_F^2$.

Compute the Cross-Entropy criterion   Finally, calculate the cross-entropy criterion:

```
./bin/crossEntropy -x examples/panel11.geno -K 5
```

log for crossEntropy:

```
summary of the options:

        -n (number of individuals)       934
        -L (number of loci)              10757
        -K (number of ancestral pops)    5
        -x (genotype file)               examples/panel11.geno
        -q (individual admixture)        examples/panel11_I.5.Q
        -g (ancestral frequencies)       examples/panel11_I.5.G
        -i (with masked genotypes)       examples/panel11_I.geno
        - diploid

Cross-Entropy (all data):         0.746568
Cross-Entropy (masked data):      0.756867
```

The `crossEntropy` program displays the cross-entropy calculated on all data and on masked data. The cross-entropy for all data is always lower than the cross-entropy for masked data. The cross-entropy value useful to compare runs is the **cross-entropy for masked data**.

All in once We can also run all three programs in once using the `-c` option:

```
./bin/sNMF -x examples/panel11.geno -K 5 -c
```

6.3 How to choose K

To choose the best value for K, we launch `sNMF` for each value of K from 2 to 10. The figure below displays the cross-entropy value obtained for each value of K. It seems that $K = 6$ is the best value according to the cross-entropy criterion because the cross-entropy does not seem to decrease for K greater than 6. This analysis has for sure to be completed by displaying the admixture barplots associated with each result.

```
for K in 2 3 4 5 6 7 8 9 10
do
        echo K=$K; ./bin/sNMF -x examples/panel11.geno -K $K -c > examples/panel11.$K.log
done
grep "Cross-Entropy (masked data):" examples/panel11.*.log
```
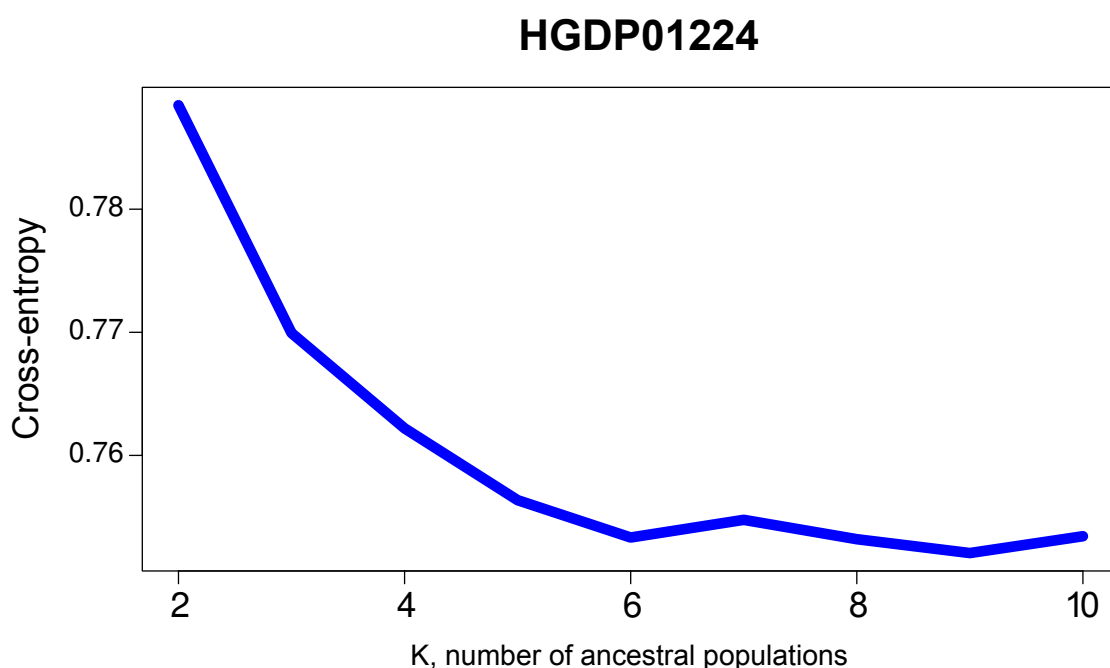

Figure. Values of the cross-entropy criterion for `sNMF` runs (HGDP01224). Values of the cross-entropy criterion for the number of clusters ranging from 2 to 10.

Tips: It is clearly useful to perform several runs of `sNMF` for the same value of K and to choose the run with the smallest value of the cross-entropy criterion.

7 Contact

If you need assistance, do not hesitate to send us an email (efrichot@gmail.com or eric.frichot@imag.fr or olivier.francois@imag.fr). A FAQ (Frequently Asked Questions) section is available on our webpage (ttp://membres-timc.imag.fr/Olivier.Francois/snmf.html). `sNMF` software is still under development. All your comments and feedbacks are more than welcome.

References

[1] Eric Frichot, François Mathieu, Théo Trouillon, Guillaume Bouchard, and Olivier François. Fast inference of individual ancestry coefficients using non-negative matrix factorization algorithms. *Submitted.*

[2] 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.