A short manual for sNMF
(command-line version)

Eric Frichot
efrichot@gmail.com

November 20, 2013

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

Contents

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

1 Description

Inference of individual admixture coefficients, which is important for population genetic and association studies, is commonly performed using compute-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 admixture 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 genomic 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 admixture coefficients within run-times approximately 10 to 30 times faster than those of ADMIXTURE

[1] Eric Frichot, François Mathieu, Théo Trouillon, Guillaume Bouchard, Olivier François. *Fast Inference of Admixture Coefficients Using Sparse 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 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 sNMF directory.

3 Data format

3.1 input file

The sNMF input file consists of a single genotype file in **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 reference allele. 1 means one copy of reference allele. 2 means two copies of reference allele. 9 means missing data.

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

  ```
  112
  010
  091
  121
  ```

We also maintain C functions to convert from the following formats to geno format.

- **ped** (example.ped)

  The **ped** format has one row for each individual. Each row contains 6 columns of information about the individual, plus two genotype columns for each SNP. Each column can 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 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. 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:

  ```
  ./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 extension .geno.

- **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 line. It is also assumed that the genotypes for a set of individuals are given in the same order of 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:

```
./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 extension .geno.

- **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:

```
./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 ancestrymap 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:

```
./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 **.F**) contains the ancestral genotype frequencies. It is a matrix with $nc \times L$ lines (the number of alleles times the number of SNPs) and K columns (the number of ancestral populations). For each SNP, the first line contains the ancestral frequencies for allele 0, the second line for allele 1, ...

4 Run the programs

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

```
./sNMF -g genotype_file.geno -K number_of_ancestral_populations
```

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

- `-g 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:

- `-p p` is the number of processes that you choose to use if you run the algorithm in a parrallel computer. Be aware that the number of process has to be lower or equal than the number of physical processes available on your computer (default: 1).
- `-i iteration_number` is the max number of iterations in algorithm (default: 200). The algorithm should not go until the max number of iterations. The stopping criterion should depend on the tolerance error only.
- `-a alpha` is the value of the regularization parameter (by default: 101). Results can depend on the value of this parameter, especially for small data sets.
- `-e tolerance` is the tolerance error (by default: 0.0001).
- `-s seed` is the initialization for the random parameter (by default: random).
- `-m ploidy` 1 if haploid, 2 if diploid (default: 2).

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

```
./sNMF -h
```

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

5 Cross-Entropy criterion

We provide two programs that compute a cross entropy score for the data.

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

  ```
  ./createDataSet -g genotype_file.geno
  ```

  The mandatory option is:

  - `-g 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 with a _**I** extension to differentiate 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).
  - `-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 data. The command line format is:

  ```
  ./crossEntropy -g genotype_file.geno -K number_of_ancestral_populations
  ```

The mandatory options are:

- `-g genotype_file.geno` is the path for the genotype file (in .geno format).
- `-K number_of_ancestral_populations` is the number of K 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 (are not mandatory):

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

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 934 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) [2].

6.2 Create a data set with masked data

In the main directory, type:

```
./createDataSet -g examples/panel11.geno
```

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

6.3 Run `sNMF`

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

```
./sNMF -g examples/panel11_I.geno -K 5
```

The results files `examples/panel11_I.Q` `examples/panel11_I.F` have been created.

6.4 Compute the Cross-Entropy criterion

Finally, calculate the cross-entropy criterion:

```
./crossEntropy -g examples/panel_11.geno -K 5
```

With this procedure, we can compute a value for the cross-entropy criterion for each of your analysis. It is a way to choose the best run for different number of ancestral populations (K) and for different values of the regularization parameter (α).

7 Contact

If you need assistance, do not hesitate to send me an email (efrichot@gmail.com or eric.frichot@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 admixture coefficients using sparse 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.