Documentation for EMALAM

Carola Sophia Heinzel[(1,*)], Peter Pfaffelhuber[(1)]
(1) Department of Mathematical Stochastics, Albert-Ludwigs-University Freiburg, 79104 Freiburg im Breisgau, Germany
(∗) Corresponding author: Carola.Heinzel@stochastik.uni-freiburg.de

August 15, 2024

1 Introduction and Overview

To estimated the ancestry of individuals, we can use the Admixture Model and a maximum likelihood estimator. We proved that this estimator is usually not unique. Hence, EMALAM calculates some other maximum likelihood estimators. The user determines which other estimators EMALAM calculates.

To apply the admixture model, we need some genetic data $x = (x_{i,j,m})_{i=1,\dots,N,j=1,\dots,J,m=1,\dots,M}$ (where $x_{i,j,m} \in \{0,1,2\}$ determines the number of copies of allele j in individual i at locus m). We assume that we deal with diploid individuals, i.e. $\sum_{j=1}^{J} x_{i,j,m} = 2$ and we apply the admixture model in the unsupervised setting. This means that we fix the number of distinct ancestral populations K and aim to infer the IAs $q = (q_{i,k})_{i=1,\dots,N,k=1,\dots,K}$ and the allele frequencies $p = (p_{k,j,m})_{k=1,\dots,K,j=1,\dots,J,m=1,\dots,M}$ from the genetic data. Here, $q_{i,k}$ stands for the part of the genome of individual i from population k (i.e. the IA of individual i in population k) and $p_{k,j,m}$ for the frequencies of allele j in population k at marker m. We write $q_{i,\cdot}$ for a row vector $(q_{i,1}, \dots, q_{i,K})$, $p_{\cdot,j,m} = (p_{1,j,m}, \dots, p_{K,j,m})^\top$. Additionally, we assume that C_x is a constant which only depends on x.

The log-likelihood for p, q, provided the data x is,

$$\ell(q,p|x) = C_x + \frac{1}{2MN} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{j=1}^{J} x_{i,j,m} \log(q_{i,\cdot} p_{\cdot,j,m}). \tag{1.1}$$ eq:L|P

Here, $q_{i,\cdot} p_{\cdot,j,m}$ is a scalar product, i.e. $q_{i,\cdot} p_{\cdot,j,m} = \sum_{k=1}^{K} q_{i,k} p_{k,j,m}$. Apparently, the likelihood (1.1) depends on q, p only via $(q_{i,\cdot} p_{\cdot,j,m})_{i=1,\dots,N,j=1,\dots,J,m=1,\dots,M} = qp$. If S_K is an invertible matrix with K rows, we find (since $\hat{q}\hat{p} = \hat{q} S_K S_K^{-1} \hat{p}$)

$$\ell(\hat{q},\hat{p}|x) = \ell(\underbrace{\hat{q} S_K}_{=:\tilde{q}}, \underbrace{S_K^{-1}\hat{p}}_{=:\tilde{p}} | x) \tag{1.2}$$ P

for all x. Additionally, we have to make sure $\sum_{k=1}^{K} (\hat{q}S_K)_{i,k} = 1, \hat{q}S_K \geq 0$ and $S_K^{-1}\hat{p} \geq 0, \sum_{j=1}^{J} \left(S_K^{-1}\hat{p}\right)_{k,j,m} = 1$. Hence, EMALAM calculates the matrices S_K that are the "most extreme", where the user defines what exactly this means. The final output are the "most extreme" estimated IAs and estimated allele frequencies.

Here, we provide an explanation for using the software EMALAM. Briefly, this software has as an input the estimated IAs and the estimated allele frequencies. To estimate them, we could e.g. use STRUCTURE[5]. Additionally, EMELAM requires some additional information from the reader that is specified in section 3.

2 Explanation how EMALAM works

We use `scipy.minimze` to minimize our objective function. To explain the method, we need some notation.

We write $\sigma(\hat{q})$ for all permutations of the matrix $\hat{q}$ and $\sigma(\hat{q})_k$ for the kth permutation of this matrix. For example, let $\hat{q} = \begin{pmatrix} 0.2 & 0.3 & 0.5 \\ 0.9 & 0.1 & 0 \end{pmatrix}$, i.e. $N = 2, K = 3$. Then,

$$\sigma(\hat{q})_1 = \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.9 & 0 & 0.1 \end{pmatrix}$$
$$\sigma(\hat{q})_2 = \begin{pmatrix} 0.2 & 0.3 & 0.5 \\ 0.9 & 0.1 & 0 \end{pmatrix}$$
$$\sigma(\hat{q})_3 = \begin{pmatrix} 0.3 & 0.2 & 0.5 \\ 0.1 & 0.9 & 0 \end{pmatrix}$$
$$\sigma(\hat{q})_4 = \begin{pmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0 & 0.9 \end{pmatrix}$$
$$\sigma(\hat{q})_5 = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0 & 0.9 & 0.1 \end{pmatrix}$$
$$\sigma(\hat{q})_6 = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0 & 0.1 & 0.9 \end{pmatrix}.$$

Additionally, the matrix $S_K(a)$ is defined as in the corresponding paper, i.e. as in equation (1.2). The a emphasizes that this matrix depends on the parameter a. We define the matrix A and the vector b so that $Aa \leq b$ make sure that the estimated IAs are between 0 and 1 and that they sum up to one, i.e. $\sum_{k=1}^{K} \hat{q}_{i,k} = 1, \hat{q}_{i,k} \in [0, 1]$. Here, the vector a contains the same parameters as the matrix $S_K(a)$, but with an other format.

EMALAM minimizes or maximizes a function $f_{obj}(a)$ with respect to a and the constraints

$C1$ $Aa \leq b$

$C2$ $S_K^{-1}(a)\hat{p} \geq 0$

$C3$ $\sum_{j=1}^{J} \left(S_K^{-1}(a)\hat{p}\right)_{k,j,m} = 1$

$C4$ $|\hat{q}S_K - \hat{q}| \leq |\hat{q}S_K(a) - \sigma(\hat{q})_k| \quad \forall k = 1, \ldots, K.$

Here, condition C1 makes sure that we can interpret $\hat{q}S_K(a)$ as the IAs. Additionally, the conditions C2 and C3 make sure that the estimated allele frequencies are between 0 and 1 and that it holds $\sum_{j=1}^{J} p_{k,j,m} = 1$. Condition C4 is optional and makes sure that the output of EMALAM does not belong to Label Switching, if we consider the most similar IAs. However, please note that the running of EMALAM with this condition C4 takes much longer than without C4. Hence, we recommend to use C4 only for small number of individuals and small number of markers.

3 Application Decisions for the User

information⟩ There are two different decisions that the user can make: The choice for the function that should be maximized and the choice for the measure of similarity to take label switching into account. We explain the choice concerning the function that should be maximized first. Afterwards, we consider the choice concerning the measure of similarity for label switching.

3.1 Choice of the Target Function

Researchers can use different functions to maximize/minimize. The five possibilities have different objective functions $f_{obj}(a)$, where $a = (a_i j)_{i=1,\ldots,K-1, j=1,\ldots,K}$ are the parameters in the matrices S_K. We discuss them in more detail now.

(I) Maximize and minimize the estimated IA for the first individual in the input data of the estimated IAs in every population (Specified by P1 in the code). In this case the user also has to specify the index of the individual. Here, we define

$$f_{obj}(a) = \tilde{q}_{ind,k}.$$

We minimize and maximize every the estimated ancestry of every population for this individual *ind*.

(II) Maximize the admixture of the estimated IAs (Specified by P2 in the code). We maximize the entropy

$$f_{obj}(a) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \tilde{q}_{i,k} \ln(\tilde{q}_{i,k})$$

which consequences the maximal admixture.

(III) Minimize the admixture of the estimated IAs (Specified by P3 in the code). We minimize the entropy. This consequences the minimal admixture.

(IV) Maximize the admixture for a specific population (Specified by P4 in the code). We maximize the admixture in population $k_{spcific}$, i.e. we minimize

$$f_{obj}(a) = -\sum_{i=1}^{N} \tilde{q}_{i,k_{specific}}.$$

This consequences the minimal admixture in population $k_{spcific}$.

(V) Minimize the admixture for a specific population (Specified by P5 in the code). We maximize the admixture in population $k_{spcific}$. This consequences the minimal admixture in this population.

This means that, we have to maximize a function under constraints. This is an often considered problem, see e.g. [4, 2, 3]. Here, we solved this problem with a new method, EMALAM. Specifically, we can apply EMALAM to single individuals or to whole populations as described above to find the most extreme optima.

If `poss` = "P1", EMALAM chooses (I), if `poss` = "P2", EMALAM chooses (II), and if `poss` = "P3", EMALAM chooses (III). Additionally, if `poss` = "P4", EMALAM chooses (IV) with $k_{specific} = 0$ and if `poss` = "P5", EMALAM chooses (V) with $k_{specific}$

3.2 Choice of the Definition for Label Switching

Additionally, we take label switching into account. Therefore, there are different possibilities to define the similarity of the different modes. Here, we use the euclidean norm. This means that we choose the labels for the populations in order to minimize the euclidean norm between the different estimators for the IA. Let us consider a simple example for this.

Let $\hat{q}_{1,1} = 0.4, \hat{q}_{1,2} = 0.6, \hat{p}_{1,1,1} = 0.9, \hat{p}_{2,1,1} = 0.2$ be the output of STRUCTURE for $K = 2, M = 1, N = 1$ (Figure 1). Furthermore, we have the output $\tilde{q}_{1,2} = 0.7, \tilde{q}_{1,1} = 0.3, \tilde{p}_{1,1,1} = 0.6, \tilde{p}_{2,1,1} = 0.2$ for an other run of STRUCTURE. Now, we consider two possibilities to depict these two results , i.e to avoid label switching:

(i) Minimize the differences between the allele frequencies (second column in Figure 1).

(ii) Minimize the differences between the IAs (first column in Figure 1).

EMALAM uses possibility (ii). Specifically, EMALAM uses the assignment of population labels to the estimated IAs and allele frequencies with the smallest euclidean norm between the estimated IAs for the $K!$ different possibilities.

However, it is easy to change this in the function `constraint4` or otherwise, apply pong to take label switching into account.

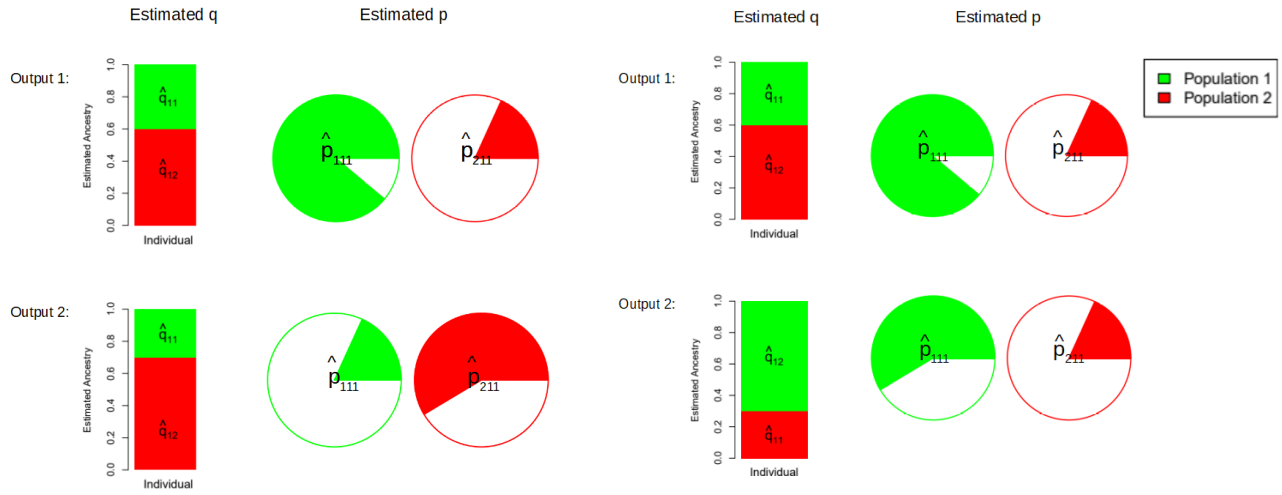

Figure 1: Example Label Switching

⟨ls2⟩

4 Input of EMALAM

There is some information that the user has to provide:

(i) Estimated IAs

(ii) Estimated allele frequencies

(iii) `simi`: Either 1 (takes label switching into account) or 0 (does not take label switching into account, i.e. we have to apply a software as e.g. pong [1] to the output of EMALAM afterwards). The details are described in condition C4. If simi = 1, C4 is an additional constraint of the minimization problem. However, this is only recommended for small number of individuals and markers since the run time is very long for simi = 1.

(iv) `J_m`: Counts the number of different alleles for marker m

(v) `poss`: Definition of the function $f_{opt}(x)$, i.e. of the most extreme values as described above. Here, the user chooses between (I), (II), (III), (IV) or (V). Alternatively, you can also define the function by yourself.

(vi) The names of the output file (including the directory where they should be saved). This is specified in the list `names`. The first entry is the name for the ancestries and the second entry stands for the allele frequencies. If the user chooses `poss = "P1"`, then the output is $2K$ different files, named `names[j]_i`, i = 1, . . . , 2K, j = 0,1.

(vi) $k_{specific}$: Population that are considered in P4 or P5.

We explain the information (i), (ii), (iv) in more detail and start with (i) and (ii).

We provide example files for the input, called q_migtration7_mutation1, p_migtration7_mutation1. They contain the estimated IAs and allele frequencies respectively. Specifically, for the estimated IAs, the file has to contain N rows and K columns. The rows represent the individuals and the columns represent the populations. For the estimated allele frequencies and $J = 2$ the file also contains K columns, but M rows. Every row stands for one marker and every column stands for one population.

The required format can e.g. be created by applying the code in `Extract_q_p.R` to the output of STRUCTURE for $J = 2$. For J arbitrary, they can be extracted with the code `Extract_p_J_arbitrary.R`. Please note that we can exclude the allele frequencies that are the same in every population from the input for EMALAM. Our code does this, if the allele 0 has either the frequency 0 or 1.

Let us consider (iv) in more detail. J_m is a list, where the component m stands for number of different alleles minus one of the m^{th} marker. Let us consider two examples with $M = 2$. If both markers are bi-allelic, $J_m = [1, 1]$. Only bi-allelic markers is the default setting. If the first marker has four different alleles and the second one three, it holds $J_m = [3, 2]$.

5 Interpretation of the Output

We also provide an example output. They are called
test_q1_K3_1.txt,
test_q1_K3_2.txt,
test_q1_K3_3.txt,
test_q1_K3_4.txt,
test_q1_K3_5.txt,
test_q1_K3_6.txt,
test_p1_K3_1.txt,
test_p1_K3_2.txt,
test_p1_K3_3.txt,
test_p1_K3_4.txt,
test_p1_K3_5.txt,
test_p1_K3_6.txt
and have the same format as the input. Additionally, the likelihood is also the same as the one for the input estimators. However, this output is the most extreme one in the sense that the user chose.

Here, the test_q1_K3_1.txt and test_p1_K3_1.txt belong to each other, i.e. have the same likelihood as test_q1_K3_2.txt and test_p1_K3_2.txt.

6 Depiction of the Results

You can use the Code Create_Figures.py to depict the different estimated IAs. On the x-axis, you have the Individuals and on the y-axis, you have the estimated IAs. Figure 2 is an example for the depiction of the results.

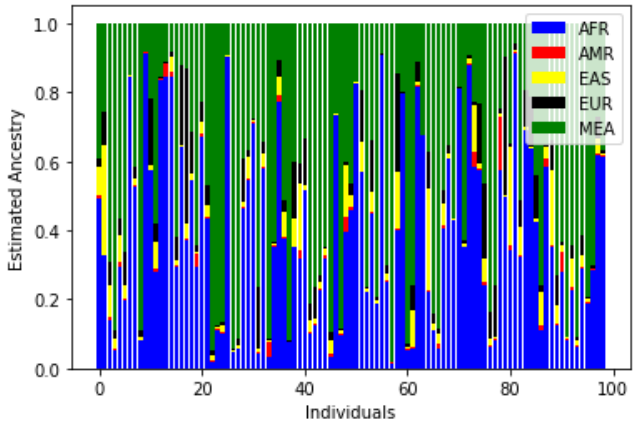

Figure 2: Example Depiction of the estimated IAs

⟨Fig:ex⟩ The list

$$data = [[0.326, 0.004, 0.318, 0.097, 0.254],\\ [0.14, 0.008, 0.094, 0.065, 0.693],\\ [0.053, 0.004, 0.027, 0.027, 0.889],\\ [0.296, 0.01, 0.078, 0.05, 0.567]]$$

is an example for the input of this code. However, we present an other example with more individuals in Figure 2.

References

behr2016 [1] Aaron A Behr, Katherine Z Liu, Gracie Liu-Fang, Priyanka Nakka, and Sohini Ramachandran. Pong: fast analysis and visualization of latent clusters in population genetic data. *Bioinformatics*, 32(18):2817–2823, 2016.

kraft1988 [2] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht-Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

lalee1998 [3] Marucha Lalee, Jorge Nocedal, and Todd Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8(3):682–706, 1998.

powell1994 [4] Michael JD Powell. *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Springer, 1994.

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