exML: Explainable maximum likelihood tool for analysis of ancient DNA samples

Abstract

We propose to demonstrate *exML*, an explainable analysis tool for DNA classification. The underlying problem is to estimate the proportion of different species that contribute to a given genomic dataset. While there are available solutions, they lack explainability. To this end, our proposed solution incorporates novel explanation methods that accompany a Maximum Likelihood analysis in this context. The explanations the demonstrated tool generates include a variant of the commonly used attribution method that assigns influence scores to individual datums based on Shapley values, and a notion of counterfactuals that identify subsets of the input data that together are most influential. In our demonstration, we will focus on the analysis of ancient DNA samples collected from archaeological sites, and show use cases where the explanations generated by *exML* provide insights on otherwise ambiguous classification results.

Keywords: genomics

ACM Reference Format:

. 2022. exML: Explainable maximum likelihood tool for analysis of ancient DNA samples. In *Proceedings of ss (CIKM '22).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The amount of genomic data has been increasing dramatically and steadily ever since the first full human genome was sequenced in 2001 [6], to a state that it is predicted that by the end of 2025, genomes of 60 million humans will be sequenced and analyzed [1]. This allowed research progress that facilitates advances in the fields of healthcare, evolutionary research, and forensic investigations.

One main algorithmic challenge in the field, which will be the focus on our demonstration, is to quantify the contribution of different organisms to a given DNA dataset [5]. Formally, given a set of species S and a dataset of DNA reads D, the goal is to output a distribution vector that assigns for each $s \in S$ a proportion that estimates how abundant s is in D. Though there are existing methods to the problem that generally attain good accuracy, they often perform badly when used to classify small datasets (<100 DNA reads), and lack explainability (see [8]) which may lead to errors and difficulties in resolving ambiguities in the model output.

As a use case, we will focus on tackling this classification problem on data of *ancient DNA of hominins*. Sequencing the DNA of ancient individuals can be used as a virtual time machine to explore our evolutionary history and that of our closest extinct relatives, the Neanderthals and the Denisovans. To date, over 6,000 ancient human genomes have been sequenced from fossilized skeletal remains [3], to assemble a database of labeled reference sequences. The retrieval of DNA from sediments deposited at archaeological sites has recently opened new avenues of research (e.g. [7]), allowing to generate ancient genetic data even in the absence of skeletal remains, by extracting and sequencing the DNA molecules that were deposited in the sediment for up to tens of thousands of years. Yet, this new type of data is inherently more complicated to analyze, requiring the development of new tools, to deal with the challenges of deducing research insights from *small amounts of data*, *corruption of DNA molecules* over tens of thousands of years, and the fact that a single DNA data sample can have DNA sequences *originate from multiple individuals*.

Explainability is a key challenge in this respect as the outputted proportions may, in some cases, be ambiguous. For example, if a system estimates that a certain dataset consists of 50% Neanderthals and 50% Homo sapiens, this result can be interpreted in different ways: **1)** Neanderthals and Homo sapiens lived jointly in a single location. **2)** All reads are ambiguous, leading to the inability of the model to determine the species of origin. **3)** The data originated from another species that is equally distant to both Neanderthals and Homo sapiens. State-of-the-art solutions do not allow to easily differentiate between the three, whereas in our demonstration we will exemplify how the system we describe in the paper, and specifically the versatile explanations it generates, can shed light on the correct way to interpret such output.

We present exML, an explainable DNA analysis system that addresses the above challenges. The system is both attaining state of the art level accuracy in the classification problem, and also provide multiple explanations to the output. The system contains a Maximum Likelihood algorithm for the proportion estimation, and is also equipped with explainability capabilities of multiple flavors, that are on the one hand tailored to the specific settings but may be

CIKM '22, 2022, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00

https://doi.org/XXXXXXX.XXXXXXX

of general interest in the context of explaining Maximum Likelihood systems. We identify two axes of explanations, the first is the type of explanation, namely (a) explanations that are based on quantifying the contribution of individual datums (e.g., DNA reads) or (b) explanations that are based on counterfactuals, which are dataset modification that affect the classification in certain ways. The second axis is whether we explain a classification via data points from the given dataset or via hyper-parameters of the model; the latter are assigned in our setting using a database of *DNA references* which are labeled full genomes of an organism.

In designing each type of explanation, we are inspired by existing methods, but adapt them to our settings. Two kinds of adaptations are in order: First, we need to adapt existing definitions. For instance, in the context of attribution of individual contribution, a standard solution is SHAP [4] which is based on Shapley values. We show that in our context, a modification to Shapley values is needed, as they are based on quantifying the influence of a player on the *result* of a game, when being added to a subset of players, whereas we estimate the influence of adding a DNA read on the *proportions* of species. This requires to take into account that in our case, the influence of adding a single player to a group is expected to be smaller the bigger the group is. Another adaptation we applied is to adjust the explanation computations to our settings. For instance, both Shapley-based and Counterfactual-based explanations involve executing the explained algorithm on multiple subsets of the data, which is typically computationally expensive. In our settings however, we were able to optimize this computation by utilizing the structure of the Maximum Likelihood method.

In this paper, we describe the maximum likelihood algorithm we applied on the classification problem, and show the accuracy it attains on several datasets consisting of simulated and real-world ancient DNA reads of hominids (Homo sapiens, Neanderthals, or Denisovan). We also demonstrate how the different explanations exMLgenerates complement each other and allow users to obtain a holistic view of the analysis results and gain further insights on the data and model. We will describe in details the several demo scenarios we intend to demonstrate, including the datasets that will be analyzed, and will explain how the GUI we created is used to interact with the system.

2 Algorithm for DNA classification

We first describe the general settings, and a maximum likelihood algorithm for the problem. We have in hand a dataset D of DNA reads and a dataset R of labeled DNA references. Each of the references in R is labeled with an item of the group S that intuitively corresponds biological species. Our goal is to estimate the proportions of the different species in D, formally, to output a distribution vector v, such that for each $s \in S$, v_s is the estimated proportion of species s in D. We further have a substitution matrix M s.t $M_{i,j,k}$ is the likelihood of observing the letter j in index k of a DNA read, given that the DNA read is originated from an organism that is closely related to an organism that has the letter i in index k. This matrix is used to estimate the likelihood that an observed DNA read d is related to a reference r, and intuitively addresses the facts that genomes of closely related organisms are very similar, but not identical, and the DNA corruption that can affect the input dataset D. Table 1 summarizes the main notations.

Algorithm 1 is a Maximum Likelihood estimator for this task. It consists of two steps (captured by methods invocation in lines 1 and 2): `PreProcessing`, namely estimating the likelihood of the observed data as a function of the model parameters, and `MaximizeLikelihood`, namely finding parameters to optimize this function. The `PreProcessing` method first aligns every DNA read d to every labeled DNA reference r, to find r_d - sub sequence of r that is most similar to d (line 6). Then, using r_d and M, it calculates $p[r_d]$ - the estimated likelihood to observe the DNA read d in a genome of an organism that is closely related to an organism with genome r (line 7). In lines 9-11, for each species $s \in S$ and read $d \in D$, it calculates $A_{s,d}$ - the average of the $P[r_d]$ values for references labeled with s. The `PreProcessing` method returns the $A_{s,d}$ values, and they are sent as parameters to `MaximizeLikelihood`, that uses them to return the species proportions that maximizes the likelihood of D (lines 15-16), which is the output of the algorithm. Note that the algorithm is highly parallelizable, as the entire loop in lines 4-11 can be executed independently for every DNA read d.

We implemented Algorithm 1 and tested its performance on 4 datasets of ancient human DNA reads collected from sediments, and on simulated datasets of ancient hominin DNA, in which the correct species proportion is known (data is simulated based on real genomes of hominins using the ancient DNA simulator tool published in [7]). For each dataset, we executed Algorithm 1 and calculated the KL divergence [2] between the correct proportions and the output of the algorithm. Figure 1 shows the results, and compares our tool to Kallisto, a method based on k-mers and pseudo alignment that is the state-of-the-art method in classification of ancient DNA (see [7]). Note that when $|D|$ is small (as usually is the case in ancient DNA data), this simple algorithm outperforms state-of-the-art technique, and in larger $|D|$'s it is at least as accurate (though it runs slower, due to the computationally expensive alignment step).

Example 2.1. We generated 2 datasets on which we will exemplify the output of Algorithm 1: D_1 contains 20 DNA reads from Homo sapiens, 20 DNA reads from Denisovans, and 40 DNA reads from regions in the genome that are conserved across all hominin genomes (*i.e.,* non informative regions). D_2 contains 40 DNA reads from Homo sapiens and 40 DNA reads from Denisovans. The average length of a DNA read in both

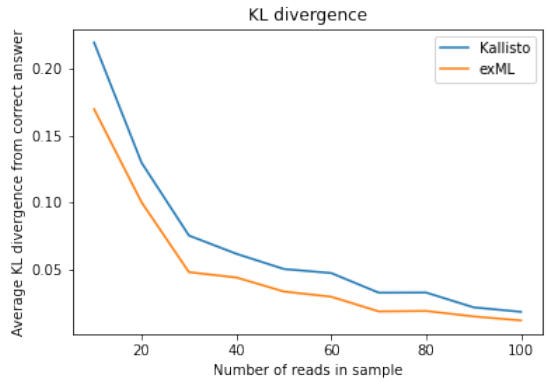

Figure 1. Comparison of exML and kallisto method, we generated 500 samples for each sample size, and calculated the average KL divergence between the output of the different algorithms and the correct answer (smaller KL divergence from correct answer means better accuracy).

DNA read	A word $\in \{A, C, T, G\}^*$
D	Input dataset of DNA reads
S	Set of possible labels (species)
R	Set of DNA references (full genomes)
$Y \in S^{\|R\|}$	Labels for references, $Y_i \in S$ is the label of R_i
$\alpha^* \in [0,1]^{\|S\|}$	Algorithm's output - distribution vector ($\sum_i \alpha_i^* = 1$ and $\alpha_i^* \geq 0$)
R_s	$\{R_i \in R \| Y_i = s\}$
M	Substitution matrix
$v(D) \in [0.1]^{\|S\|}$	Result of running Algorithm 1 on dataset D.

Table 1. Table of notations

datasets is 75. When running Algorithm 1 on these datasets, using the label set $\{Homosapiens, Neanderthal, Denisovan\}$, and a database of labeled genomes of hominin species as references set, the output was $v(D_1) = v(D_2) = (0.5, 0, 0.5)$. This output intuitively means that for both datasets, the estimation of the system is that the proportions that best fit the data are 50% Homo sapiens DNA, no Neanderthal DNA and 50% Denisovan DNA.

3 Explanations

In the explanations our system produces, we employ state-of-the-art techniques for explanations, and adapt them to the particular settings in hand. Specifically, we observe that explanations are relevant for both *reads* and *references*. We next detail both types, describe the algorithms our system utilizes to generate them, and show examples to their usefulness on real world scenarios in analysis of ancient DNA.

Algorithm 1: Maximum likelihood algorithm to estimate proportion of species in genomic dataset

Data: input dataset D, substitution matrix M, label set S, labeled set of DNA references R
Result: Distribution α^* - species proportion estimation

```
A = PreProcessing(D, R, M)
return MaximizeLikelihood(A, D)
Procedure PreProcessing(D, R, M)
    foreach d ∈ D do
        foreach r ∈ R do
            Align d to r to get r_d:=sub sequence of r that is
              most similar to d
            P[r_d] = ∏_{t=1}^{|d|} M_{(r_d[t],d[t],t)}
        end
        foreach s ∈ S do
            A_{s,d} = (∑_{r∈R_s} P[r_d]) / |R_s|
        end
    end
    return all A_{s,d} values
Procedure MaximizeLikelihood(A,D)
    L(D;α) := ∏_{d∈D} ∑_{s∈S} α_s * A_{s,d}
    return α* := argmax_α(L(D;α)) with the constraints
      ∑_i α_i = 1 and α_i ≥ 0  ∀i.
```

3.1 Read-level explanations

A read-level explanation aims to quantify and explain the influence of a single read, or a subset of reads, from the input data set on the output of the algorithm. We present two kinds of explanations at the read-level: attribution-based and Counterfactuals.

3.1.1 Shapley based read attribution.

Definition 3.1. Given a set of DNA reads D, a set of labels S, and an algorithm A that gets D as input and outputs a distribution vector over S, we define a *read-level attribution* for $d \in D$ as a vector $v^d \in R^{|S|}$, such that v_s^d is the influence of d on the proportion A assigns to label $s \in S$.

Example 3.2. As an example for definition 3.1, A can be Algorithm 1 described above, that gets as input a dataset D of DNA reads, S is the set $\{HomoSapiens, Neanderthal, Denisovan\}$ and the explanation of a single DNA read d is a vector $v^d \in R^3$ where v_0 quantifies how d changes the proportion Algorithm 1 assigns to Homo sapiens. For instance, if v_0^d is a positive number, it means that d makes the algorithm tend to increase the estimated proportion of Homo Sapiens.

To generate attribution based read-level explanations, we apply the notion of Shapley values, with modifications to our settings. In the context of cooperative games, Shapley values estimate the contribution of a single player to the result of the game. The Shapley value of a player i is defined as (v is

the value of a game and N is the number of players):

$$\frac{1}{N!}\sum_{G\subseteq[N]\setminus\{i\}}[v(G\cup i)-v(G)]*|G|!*(N-|G|-1)! \quad (1)$$

To use Shapley values in our context, we interpret every DNA read $d \in D$ as a player of a game, and $v(G)$ as the result vector of running Algorithm 1 on a subset $G \subseteq D$. Then, the output of the formula above is a vector that its j'th entry encodes the read's contribution to the estimated proportion of species j. However, to apply this idea for generating read-level explanations, two main adaptations are needed:

Execution on Samples. Shapley value formula requires executing an algorithm on a subset of its input set. When it is used in Machine Learning models, usually this requires the output of a model on a subset of its features, to quantify the contribution of the different features to the result of the model. In general Machine Learning models, it is not feasible to calculate the output of the model only on part of the features, as this requires to train the entire model from scratch (a problem that SHAP solves by using the background distribution to sample values for features it integrates out [4]). However, in our Maximum Likelihood estimator, we actually can run the model on a subset of the DNA reads, by simply rewrite the formula in line 15 of Algorithm 1 so that we plug-in a subset G instead of the entire data set D, and thus calculate what would be the output of the algorithm on the subset G. Furthermore, when we consider multiple subsets G, we only need to execute the expensive `PreProcessing` step once, and re-use the $A_{s,d}$ values it computes in multiple invocations of `MaximizeLikelihood` (A, G). This optimization allows our system to evaluate multiple outputs of the algorithm on different subsets of the input dataset, and to use that information to apply Shapley values formula and attribute the output of the model to the different reads in the dataset.

Scaling Shapley values. Since Algorithm 1 outputs proportions, and not absolute values, the Shapley formula needs to be refined, as illustrated in the following example:

Example 3.3. Consider a sample G for which the ground truth proportion is $(\frac{a}{|G|}, \frac{b}{|G|}, \frac{c}{|G|})$. Further consider a read $i \notin G$ generated from a Homo sapiens. The ground truth for $G \cup \{i\}$ is $(\frac{a+1}{|G|+1}, \frac{b}{|G|+1}, \frac{c}{|G|+1})$.
Thus $v(G) - v(G \cup \{i\}) = (\frac{|G|-a}{|G|^2+|G|}, \frac{-b}{|G|^2+|G|}, \frac{-c}{|G|^2+|G|})$ which is equal to $(\frac{b+c}{|G|^2+|G|}, \frac{-b}{|G|^2+|G|}, \frac{-c}{|G|^2+|G|})$.
Note that $||v(G) - v(G \cup \{i\})||$ decreases approximately linearly as $|G|$ grows (see Figure 2 for empirical example that shows that the bigger the sample, the smaller the influence of adding a single read on the output).

Thus, before applying Shapley values in our context, there is a need to address the fact that the magnitude of the influence of adding a single data point to a set depends on the size of that set, otherwise the explanations are dominated by the smaller samples. We therefore added a scaling parameter to the original Shapley value formula, that takes into account the size of the subset. Specifically, to generate a read-level explanation for read i, exML samples M subsets of D, and outputs the following vector as explanation:

$$\frac{1}{M}\sum_{G\in Samples}[v(G\cup i)-v(G)]*\mathbf{|\,G\,|}. \quad (2)$$

Given a read $d \in D$, the output of this process is an explanation vector v^d that corresponds to definition 3.1, and this is the vector that our system generates as an attribution based read-level explanation.

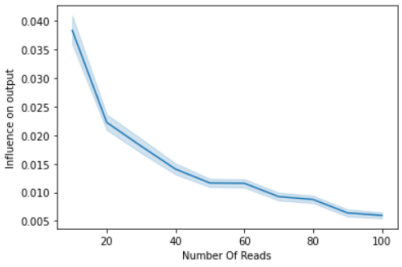

Figure 2. X axis is number of reads in the sample, Y axis is the average magnitude of change that is caused by adding a single read to a sample. The plot shows empirically that the more reads are in a sample, the smaller the influence of adding a single read is.

Example 3.4 (Scaled Shapley values)**.** Figure 3 shows read-level explanations generated by exML using the scaled Shapley values on D_1 and D_2 (as described in example 2.1). As stated above, the output of Algorithm 1 on both datasets is approximately the same, so using merely the output of Algorithm 1, one may not distinguish between them. However, as Figure 3 demonstrates, their different read-level explanations provide insights that direct the user to the correct interpretation of the results. These explanations reflect that in D_1, only the last 40 reads had substantial influence on the output, whereas in D_2, all reads were influential. This implies that the system has more evidence to its output on D_2. In addition, the non-influential reads in D_1 might also be attributed to being originated from a genomically distant organism that is not in the reference set and hence do not influence the output proportion of neither Homo sapiens nor Denisovans.

3.1.2 Read-level counterfactuals.

Given a model M and an input x such that $M(x) = l$, a *counterfactual explanation* is a modified instance y for which $M(y) \neq l$. Intuitively, modifications to x that lead to a different label are indicative of the features that were responsible for the label l being chosen to begin with. We now define and elaborate on the

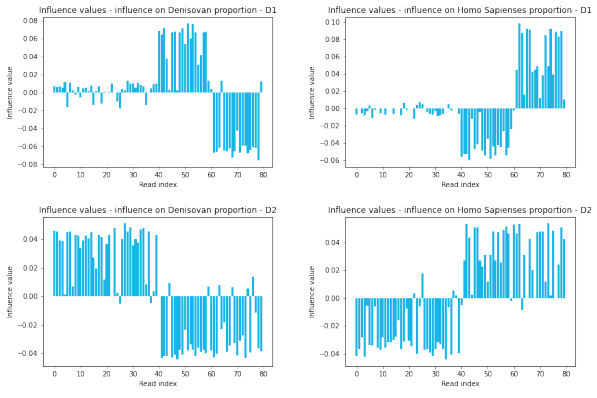

Figure 3. Read-level explanations on D_1 (top), and D_2 (bottom). x axis is the read index, and y axis is the explanation value. Right plots show the explanation values of Homo Sapiens proportions, and Left plots show explanation values on Denisovan proportions.

3 types of counterfactual our system produces. We start by defining the general settings of Maximum Likelihood:

Definition 3.5. Given a dataset D, a set of labels S and a likelihood function $L(\alpha, D) \in ([0,1]^{|S|}, set) \rightarrow [0,1]$, define $ML(D,L) := argmax_{\alpha} L(\alpha, D)$.

Intuitively, the likelihood function gets as input a distribution vector over the labels and a set, and outputs a probability. For example, in our settings, if D is a set of DNA reads, α is a proportion vector, S is the set of known hominin species, and L is the likelihood function of Algorithm 1 on the dataset D, $ML(D,L)$ is the vector that maximizes the likelihood term $L(\alpha, D)$, for example, it can be the vector (0,0,1).

Definition 3.6. Let $H(D,L)$ be $argmax_i(ML(D,L)[i])$ - the index of the maximal value in the vector $ML(D,L)$.

Intuitively $H(D,L)$ is the label with the highest proportion assigned to it. In our example, if $ML(D,L) = (0,0,1)$, so $H(D,L) = 2$ as this is the index that maximizes $ML(D,L)$. In our example this corresponds to saying that the algorithm estimates that the Homo Sapiens species is the most abundant in D.

Definition 3.7. Let *changing dominance subset* be a subset $D' \subseteq D$ such that $H(D,L) \neq H(D \backslash D', L)$.

This is a subset that removing it from the input dataset causes the vector that maximizes the L function to have a different maximizing index. In our example, if there are 2 DNA reads D_1, D_2 that removing them from D will cause the output of Algorithm 1 to be maximal when $\alpha = (1,0,0)$, then the set $\{D_1, D_2\}$ is a dominance changing subset with respect to L and D.

Definition 3.8. minimal or minimum?

Define $CF1(L,D)$ as a subset $D' \subseteq D$ that is a changing dominant subset, and is also minimum subset with that property, meaning that for every subset $T \subseteq D$ that is changing dominance with respect to L, $|D'| \leq |T|$.

In the field of DNA classification, this counter factual is an interesting research insight, as it is pointing out subsets of the data that make the algorithm change the decision, which can promote insights both on these subsets (maybe they are from different species then the others) and on the stability of the output (if a small subset is sufficient to make a substantial change to the output, the algorithm is less stable and trustworthy).

If we assume that L is just a random function we get as black box, even if we assume that we have an $O(1)$ oracle for the dominant species given a subset, getting an output will still take exponential time in the size of D, as we have nothing better to do other than to try all the options of changing D, and stop when we have a change that is causing the dominant species to be different. This is because if L can be any function, it can be a function that its maximizer is always the same, no matter which set it gets as input, except from a single subset on which the maximizer is substantially different. This is clear that there is no way to find that dominance changing subset other than going over all the subsets of D and finding the maximizer of $L(\alpha, G)$, until we encounter a maximizer with a differnet dominant species.

However, since we are interested in the real world scenario, we expect our L function to not be just random, and behave "smoothly". We can assume that small changes in α will cause small changes to the likelihood function, and also that small changes in the dataset will cause small changes in the likelihood function. Furthermore we can assume that removing data points from the dataset has an additive influence on the likelihood, meaning that if removing read a causes the alpha vector tend more to neanderthal, and read b as well, then removing both of them will cause more tendency to neanderthal then removing either of them by its own. As the exact calculation of the subset seems to not even be in NP, we estimate it by greedily removing reads with the highest positive influence on the dominant species, until obtaining a subset D' that meets the above disequality condition. In turn, the influence of individual reads is estimated using their scaled Shapely values. We are interested to show that with some assumptions, this is finding a 2-fold estimation to the minimal changing dominance species. Assume that there is a subset of size X that is changing dominance, we will show that the subset that our algorithm will output will be of size at most 2X.

In every iteration, the algorithm is removing from the dataset the read that is mostly "for" the dominant species proportion, expecting to quickly obtain a dataset in which the dominant species is different. It is easy to see that the set that

Algorithm 2: Calculating counter factual 1

Data: dataset D, and the output of Algorithm 1 on that dataset
Result: subset that is an estimation of cf1
1 Assume w.l.o.g that the dominant species in D is homo sapiens
2 Calculate scaled shapley values for every $d \in D$, define v^d as the vector of scaled shapely value of read d, and v^d_{hs} is the influence of this read on homo sapiens proportion.
3 Start with $D^{'} = \emptyset$, and add to it the read $d^{'}$, which is $argmax_d(V^d_{hs})$
4 Repeat until $H(D \backslash D^{'}, L) \neq Homosapiens$
5 **return** $D^{'}$ as an estimation to the minimal changing dominance subset

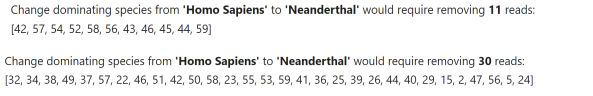

Figure 4. CF1 output on D_3 (top) and D_4 (bottom)

is returned is changing dominance, but we will show that it is also an estimation to the minimal under some assumptions.

CF2 is a minimum subset $D' \subset D$ that changes the model's output by at least ϵ, i.e. $|| \ v(D) - v(D \backslash D') \ ||> \epsilon$. To estimate it, we initialize $D' = \emptyset$, and greedily add to D' a read d that maximizes the term $|| \ \sum_{i \in D'} S(i) + S(d) \ ||$, until $|| \ v(D) - v(D \backslash D') \ ||> \epsilon$. Note that since we are comparing vector norms, d is not necessarily the read with the highest norm of the Shapley vector, but rather one that shifts the decision in the right direction w.r.t. D'.

CF3 is a maximum subset $D'_s \subseteq D$ such that $v(D'_s)[s] \geq \theta$ for each $s \in S$, where $v(D)[s]$ is the estimated proportion that Algorithm 1 assigns to species s when run on D. To estimate CF3 for a species s, we initialize $D'_s = \emptyset$, and greedily add a read d that has the highest $S(d)$ w.r.t. s, until there are no reads for which $v(D'_s \cup d)[s] \geq \theta$.

Example 3.9 (Read-level counter factuals)**.** Figure 4 shows the output of *CF1* on two datasets: D_3 (top figure) has 20 reads from Homo sapiens, and 40 non-informative reads, and D_4 (bottom) has 40 reads from Homo Sapiens and 20 non-informative reads. Though Algorithm 1's output is the same for both datasets ($(1, 0, 0)$), Figure 4 shows that on D_3, CF1 contains 11 reads, whereas in D_4, CF1 contains 30 reads. This conveys to the user that the output on D_4 is more stable, as a larger modification is required for it to change.

3.2 Reference-level explanations

We are interested in generating explanations for references - to quantify and explain to the user how every reference in the labeled references set influenced the output of Algorithm 1. We present Algorithm 3, a sampling based algorithm to calculate explanation vectors for references. The explanations that we compute follow the attribution approach. We define $A^i_{s,d}$ as the value that Algorithm 1 would calculate as $A_{s,d}$, if the references set would not include r_i. In line 1, we run the `PreProcessing` method of Algorithm 1, to get the initial $A_{s,d}$ values. In line 2 we loop over the DNA references. For $A^i_{s,d}$ values where s is the label of r_i, we set the $A^i_{s,d}$ value to be the average that would have been obtained if r_i was not in the references set (line 3); otherwise, we set $A^i_{s,d} = A_{s,d}$ (line 4). In lines 5–9 we go over the samples, and calculate the change that is caused to the output vector of Algorithm 1 by removing r_i. In line 10 we average this influence value over all samples, to get an estimation of how removing reference i influences the output of Algorithm 1. In line 12, we return V, which is a list of all V_i values (V_i is the vector exML outputs as an explanation to reference i, and it intuitively describes how reference i influences the output of Algorithm 1).

Algorithm 3: Calculating reference-level explanations

Data: References set R, input dataset D, Samples K, such that every $K_j \in K$ is a subset of D
Result: V, s.t V_i is the reference-level explanation of R_i
1 Run preProcessing() step of algorithm 1 to get $A_{s,d}$ and $p[i, d]$ values
2 **foreach** $r_i \in R$ **do**
3 For each $A_{s,d}$ such that $s = Y_i$, $A^i_{s,d} := \frac{A_{s,d} * |R_s| - p[r_i,d]}{|R_s| - 1}$
4 For each $A_{s,d}$ such that $s \neq Y_i$, $A^i_{s,d} := A_{s,d}$
5 **foreach** $k_j \in K$ **do**
6 $L^i(k_j; \alpha) := \prod_{d \in k_j} \sum_{s \in S} \alpha_s * A^i_{s,d}$
7 $L(k_j; \alpha) := \prod_{d \in k_j} \sum_{s \in S} \alpha_s * A_{s,d}$
8 $\delta_{i,j} \leftarrow argmax_\alpha(L^i(k_j; \alpha)) - argmax_\alpha(L(k_j; \alpha))$
9 **end**
10 $V_i = \frac{\sum_{j \in |K|} \delta_{i,j}}{|K|}$
11 **end**
12 **return** V

Example 3.10. Figure 5 shows the reference explanations generated by our system on two datasets, D_5 (left figure), and D_6 (right figure). Each dataset contains 50 reads generated from Neanderthals, 50 from Homo sapiens, and 50 from Denisovans. However, D_5's Neanderthals are from Neanderthal KX198082, and D_6's Neanderthals are from Neanderthal Altai KC879692. On both datasets, Algorithm 1

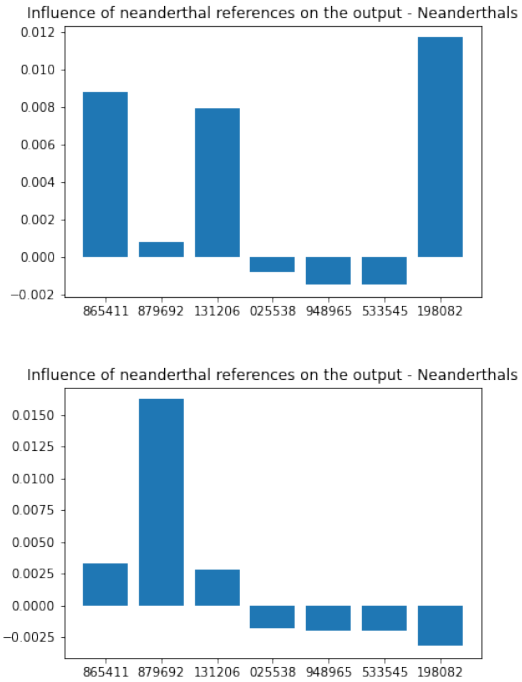

Figure 5. Reference explanations on D_5 (top) and D_6 (bottom). X axis is the reference index, and y axis is the influence of that reference on the proportion of Neanderthals in the output of algorithm 1.

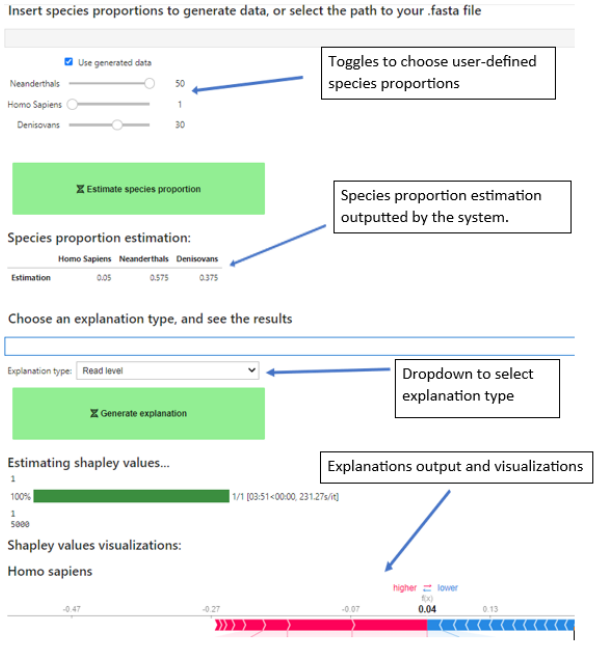

Figure 6. The GUI that is used to interact with the system

outputs the same estimation $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, but the reference explanations can help identify the specific sub group of Neanderthals that contributed to the dataset, as in both explanations, the reference that has the highest positive influence on the Neanderthal proportion is the one from which the data was actually generated.

4 Implementation

To make interactions with the system more user friendly, we created a Graphic User Interface (displayed in Figure 6). The GUI allows the user to either input a dataset of DNA reads, or to generate a dataset with user-defined species proportions (using the toggles on the top of the screen). Then, the user can execute Algorithm 1 on the generated/inputted dataset (using the "estimate proportions" button on the top section of the GUI), and view the proportion estimation that the system assigns to the dataset. Afterwards, using the drop down on the bottom part of the GUI, the user can select any explanation from the explanations that were described throughout this paper (read-level, reference-level, counter factuals), and have the system calculate the desired explanation and display visualizations that accompany it.

We implemented *exML* using python, while utilizing the following tools: *biopython* for alignments and parsing of DNA data, *joblib* for parallel computation, *SHAP* for generating visualizations and *Jupyter Widgets* for the GUI. Reference genomes were downloaded from the *NCBI database*, and simulated ancient DNA data was generated using a python script published in [7].

The source code, GUI, and explanations of how to load the GUI and relevant data sets can be found here together with instructions to execute demo scenarios and a notebook that shows results and some basic interactions and explorations with the real world data.

5 Demo scenarios

In our demonstration, we will use the GUI to walk the audience through the usage examples we described throughout this paper, focusing on the different explanations and the insights that can be drawn from them.

step 1 - data loading. We will show the audience how to use the GUI to generate the datasets for the different demo scenarios (datasets described in examples 3.4, 3.5,3.6). This will be done by either using the toggles in the top of the GUI, or by selecting a dataset from an available pool of datasets we will provide.

step 2 - running the maximum likelihood algorithm. We will run the maximum likelihood algorithm on the different datasets generated in step 1, and let the audience observe the output of the system - the estimation for the distribution of species proportions. We will show how the output is similar to the ground truth proportions, and exemplify how

the output can be ambiguous, to demonstrate the need for explanations in this context.

step 3 - read level explanations. we will walk the audience through **example 3.4** described above. After running the algorithm and observing the proportion estimation on the datasets from example 3.4, the audience will observe that the output on the datasets is ambiguous, and understand the need for the explanation to resolve the ambiguity. Hence, we will use the bottom part of the GUI to *run the read-level explanations on both datasets*, and allow the audience to view the outputs and related visualizations, and to experience how the insights that can be drawn from the explanations are helping to resolve the ambiguity and guide the user to the correct way to interpret the output, as described in example 3.4 in this text (the audience will observe Figure 3).

step 4 - reference level explanations. We will show similar in details demonstration of other example 3.5 to illustrate the usage and insigths drawn from reference level explanations, including the dataset generation, algorithm execution, and explanations execution (the audience will observe Figure 5 and understand how it helps to analyze which sub species of Neanderthal contributed more to each dataset).

step 5 - counter factuals. We will walk the audience through example 3.9, to illustrate how the counter factuals explanations are useful when trying to analyze the stability of the algorithm.

step 6 - free exploration. We we will encourage the audience to freely explore the system by using the GUI to analyze generated datasets with different proportions, or to analyze a specific dataset from a pool of datasets we will provide. To demonstrate the genericness of our method, we will also show the audience how the system attains similar levels of accuracy and explainability on *data of different organisms*, such as dogs, cats, or horses.

References

[1] Ewan Birney, Jessica Vamathevan, and Peter Goodhand. 2017. Genomics in healthcare: GA4GH looks to 2022. *BioRxiv* (2017), 203554.

[2] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[3] Yichen Liu, Xiaowei Mao, Johannes Krause, and Qiaomei Fu. 2021. Insights into human history from the first decade of ancient human genomics. *Science* 373, 6562 (2021), 1479–1484.

[4] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[5] Alice Carolyn McHardy, Héctor García Martín, Aristotelis Tsirigos, Philip Hugenholtz, and Isidore Rigoutsos. 2007. Accurate phylogenetic classification of variable-length DNA fragments. *Nature methods* 4, 1 (2007), 63–72.

[6] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. 2001. The sequence of the human genome. *science* 291, 5507 (2001), 1304–1351.

[7] Benjamin Vernot, Elena I Zavala, Asier Gómez-Olivencia, Zenobia Jacobs, Viviane Slon, Fabrizio Mafessoni, Frédéric Romagné, Alice Pearson, Martin Petr, Nohemi Sala, et al. 2021. Unearthing Neanderthal population history using nuclear and mitochondrial DNA from cave sediments. *Science* 372, 6542 (2021), eabf1667.

[8] David S Watson. 2021. Interpretable machine learning for genomics. *Human genetics* (2021), 1–15.