Explainable artificial intelligence in forensics: realistic explanations for number of contributor predictions of DNA profiles

Marthe Veldhuis

TU Delft

Netherlands Forensic Institute
Ministry of Security and Justice

Explainable artificial intelligence in forensics: realistic explanations for number of contributor predictions of DNA profiles

by

Marthe Veldhuis

in partial fulfilment of the requirements for the degree of

Master of Science
in Computer Science - Data Science and Technology Track

at Delft University of Technology,
to be defended publicly on Thursday June 17th, 2021 at 13:00.

Student number: 4395778
Project duration: November, 2020 – June, 2021
Thesis committee:

Dr. T. Abeel,	TU Delft, supervisor, committee chair
S. Ariëns,	Netherlands Forensics Institute, daily supervisor
Dr. C.C.S. Liem,	TU Delft, external committee member

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Netherlands Forensic Institute
Ministry of Security and Justice

Preface

The current report on explainable artificial intelligence in forensics is the result of my thesis project to obtain the Master of Science degree in Computer Science; particularly, for the Data Science track. This project has been a collaboration between the Bioinformatics lab at Delft University of Technology and the Netherlands Forensics Institute (NFI). As such, I had the wonderful opportunity to work on the research and development of explanations for number of contributor predictions of DNA profiles. From the beginning of my master's degree, I had the goal to work on implementations with real-world significance. Combining scientific research with practical goals has made this project especially fulfilling to me.

This document is structured into two parts; the first is a scientific paper written for the journal *Forensic Science International: Genetics* demonstrating the main results; and additional chapters that provide context of the project including methodology, experiments that did not make it into the final product, and surveys with users. The chapters follow a mostly chronological order. For readers that are unfamiliar with the forensics context of this problem, I recommend first consulting chapter 2 before reading the paper.

I would like to express my gratitude towards my supervisors from the TU Delft and NFI: Thomas Abeel and Simone Ariëns, for their guidance throughout this process. Furthermore, I would like to thank Corina Benschop for her involvement and feedback, and Rolf Ypma and Cynthia Liem for taking an interest in my work. Lastly, I am happy to have received support from the people that are dearest to me.

Marthe Veldhuis
The Hague, June 2021

Explainable artificial intelligence in forensics: realistic explanations for number of contributor predictions of DNA profiles

Marthe Veldhuis[a,b,*], msveldhuis96@gmail.com

Simone Ariëns[b], s.ariens@nfi.nl

Rolf Ypma[b], PhD., r.ypma@nfi.nl

Thomas Abeel[a], PhD., t.abeel@tudelft.nl

Corina C.G. Benschop[c], PhD., c.benschop@nfi.nl

* Corresponding author

[a] Delft University of Technology, Mekelweg 5, 2628 CD Delft, The Netherlands

[b] Netherlands Forensic Institute, Division of Digital and Biometric Traces, Laan van Ypenburg 6, 2497GB The Hague, The Netherlands.

[c] Netherlands Forensic Institute, Division of Biological Traces, Laan van Ypenburg 6, 2497GB The Hague, The Netherlands.

Explainable artificial intelligence in forensics: realistic explanations for number of contributor predictions of DNA profiles

Abstract

Using machine learning to determine the number of contributors (NOC) in short tandem repeat (STR) mixture DNA profiles has been shown to obtain good accuracy. However, the models used so far are not transparent to users as they only output a prediction without any reasoning for that conclusion. Therefore, we leverage techniques from the field of eXplainable artificial intelligence (XAI) to help users understand why specific predictions are made. Where previous attempts at explainability for NOC estimation have relied upon using simpler, transparent models that achieve lower accuracy, we use techniques that can be applied to any machine learning model. Our explanations incorporate SHAP values and a counterfactual example for each prediction into a single visualization. Existing methods for generating counterfactuals focus on uncorrelated features. This makes these methods inappropriate for the highly correlated features derived from STR data for NOC estimation, as these techniques can generate examples with impossible feature value combinations. For this reason, we constructed a new counterfactual method, Realistic Counterfactuals (ReCo), which generates realistic counterfactual explanations for correlated data. We show that ReCo outperforms some state-of-the-art methods on traditional metrics, as well as on a novel realism score. A user evaluation of the visualization demonstrates the positive opinions of end-users, which is ultimately the most appropriate metric in assessing explanations for real-world settings.

1. Introduction

1.1 Number of contributor estimation

Deriving the Number of Contributors (NOC) from Short Tandem Repeat (STR) profiles is a challenging task due to occluding factors such as allele sharing between donors, or allelic drop out [1-9]. This becomes increasingly difficult when the number of contributors rises. However, most probabilistic genotyping software that is used for weight of evidence calculations does require the NOC to be entered by the user [10, 11], which can influence the height of the likelihood ratio [2, 11-16].

Valuable steps have been made to develop methods that can more accurately predict the NOC than relying on the Maximum Allele Count (MAC)-method which involves taking the locus with the most alleles, dividing by two and rounding up [17]. The improvement mainly corresponds with incorporating more information such as for example the Total Allele Count (TAC), peak heights, drop out and stutter rates, the distribution of allele counts, and population

allele frequency [3, 5, 8, 9]. Others use more complex techniques like Bayesian networks [4]. From the multitude of models to estimate the NOC, machine learning models have shown to outperform standard methods on both accuracy and speed [13, 18-20]. However, machine learning algorithms are often considered to be *black-boxes* [21-28], as the predictions they produce are made based on generalization from training data, but the exact mechanism is not easily understood. It is important for DNA experts to know which factors the algorithm or *model* used to make a prediction. In this way, the experts can decide whether or not to trust the outcome. Perhaps the model considered some information that the expert missed, or even made a decision on information that should not be relevant to determine the NOC. By delivering this transparency, predictions can be made more understandable and more informed decisions can be made.

A method using a decision tree was presented as a more transparent way to use machine learning to estimate the NOC [29]. However, using a simple model such as a decision tree leads to less accurate predictions; they reported a decrease in accuracy of over 10% as compared to a random forest model. The method also relies heavily on filtering of artefacts, for which another decision tree is used. Furthermore, the data used in this study is also derived from a small number of donors, which means that there is little diversity and less complexity in the data. If more complex data is used, the performance of a simple model decreases even further. More complex predictors are more suited to handle such data.

It seems that there exists a trade-off between accuracy and transparency. However, none of the previously mentioned techniques have explored the field of eXplainable Artificial Intelligence (XAI). XAI has emerged to provide explanations for any type of machine learning model, since users want to know *why* a certain prediction is made [21-28]. The European Commission recently underlined the importance of explainability in a proposal for rules on AI systems in higher-risk settings such as law [30]. Though NOC estimation does not directly cause decisions without the involvement of human experts, these experts should be well-informed about the system that they might let influence their decision. We aim to provide some basic insight into XAI before diving in to how it can be used in the application of NOC estimation.

1.2 eXplainable Artificial Intelligence (XAI)

Machine learning models roughly fall into two categories when it comes to how interpretable they are; transparent- and black-box models [21, 23-28, 31]. With transparent models, one can derive the exact steps taken to arrive from input features to an output within reasonable time [24, 25]. A decision tree could be considered a transparent model, since it shows each decision made for any input to reach a prediction. It starts at the top with the root node, and splits off to different branches based on conditions specified in each node, until a

leaf node is reached which represents a prediction. This transparency is limited by the size, the complexity and components of the algorithm. In the example of the decision tree, it cannot be too large, make decisions based on complicated conditions, or use variables that are not easily understood [24, 25, 32]. If all of these conditions are violated, the model becomes a black-box. It then requires post-hoc explanations, which are generated after the underlying model has been optimized.

To achieve an explanation, we can choose to leverage some structures of the model, or create a model-agnostic explanation [23, 24, 27, 28]. The decision tree example is therefore a model-specific explanation, since it utilizes the structure of the tree to serve as the explanation. *Model-agnostic* explanations do not make any assumptions of the type of model they are explaining and thus can be applied to any machine learning model.

It is also important to determine the scope of the explanation. Either they refer to the entire model and its data such as the decision tree example (global), or to specific parts of it (local) [21, 23-28, 31]. A *local* explanation has the advantage that only information about the current decision is shown. In this way, an explanation can be more compact and simpler than attempting to portray the entire model. Conversely, the complete model could be more complex, as the explanation only contains a subset of the entire prediction space.

For NOC estimation, DNA experts look at one prediction at a time and would like the most accurate description of how a single profile is processed. As we mentioned earlier, more complex machine learning models perform best, so it might be difficult to explain the entire decision-space of the algorithm. Instead, local explanations of how individual DNA profiles are predicted seems more fit. A model-agnostic approach is preferable, since the application of machine learning models to this problem is still in development [13, 18-20]. Local, model-agnostic explanations are generally one of two types; *feature importance* or *counterfactuals*.

Feature importance methods highlight the values of the input that have driven the model to make a certain prediction [21, 23-25, 27, 31]. This effectively answers the question *"Why did the model predict A?"*. An established method for arriving at such explanations is SHAP. SHAP calculates Shapley values that show how much certain input features have contributed to a prediction, in comparison to the average prediction [33]. These Shapley values have a solid background in game theory to produce consistent explanations. For the exact method and techniques used to calculate the Shapley values, we refer to Lundberg et al. (2017) [33]. SHAP has been implemented in real-life cases such as predicting hypoxia based on clinical data [34], and predicting the most fitting eye-surgery type [35]. They seem to have obtained valuable information about which factors the ML models based their prediction on.

Counterfactual explanations are example data points which have a different prediction from the input data point [21, 23, 24, 27]. From such a counterfactual the audience could derive how the original instance could have been predicted differently if certain input features had different

values. As such, they answer the question *"Why did the model not predict B?"*. This way of reasoning is underpinned by the social sciences to be effective, as humans seek contrastive explanations [21, 22]. This field is in active development and no method has been proven to work well for all sorts of applications. With this paper, we present a counterfactual method that is suitable for the NOC prediction domain. As such, counterfactuals will be covered in more detail.

1.3 Counterfactuals

A counterfactual is an example instance that is similar to the instance we want to explain, but has a different prediction [36-51]. The differences in feature values between the input and counterfactual can give the user an impression about the local decision space of the model. More formally, a counterfactual can be described as follows [38]:

> "*The model predicted outcome* y *because input instance* x *had values* $\{x_0, x_1, \ldots, x_n\}$*. If instead instance* x *had values* $\{x'_0, x'_1, \ldots, x'_n\}$*, and all other values had remained constant, the model would have predicted outcome* y*'. "*

This alternative outcome is often referred to as the *target* of the counterfactual [36-51]. To help the user relate this new prediction as a possibility for the original input, the counterfactual must be similar to the input. To find the most suitable counterfactual, there needs to be a definition of what 'similar' entails. Most commonly, this is measured by the distance from the input to the counterfactual [37-43, 45-48, 52]. Though some methods use L_2 or Euclidean distance [43, 47], L_1 or Manhattan distance appears to be the measure of choice as it does not blow outlier distances out of proportion as L_2 distance tends to do [37, 38, 40-42, 46, 48]. This is because with Euclidean distance, the differences in feature values are squared, while Manhattan distance takes the absolute differences. Alternatively, or additionally, similarity of a counterfactual can be measured by the number of differences in feature values in comparison to the input [36, 37, 40, 42, 44-48, 50, 51], sometimes referred to as L_0 distance.

In summary, the consensus is that a counterfactual should be:

- Valid: it has the *target* outcome.
- Proximal: it has minimum *distance* to the input.
- Sparse: it has minimum *feature differences* with regards to the input.

There are more aspects to optimize such as presenting a diverse set of counterfactuals [37, 39-42, 48], or providing counterfactuals that are actionable; meaning that the changes can be acted upon to reach that alternative outcome [37, 39, 40, 43, 44, 51]. This is useful when

the input features can be changed in the future, for example by raising your income for a loan application.

To generate counterfactuals, they can either be chosen from the training data [45, 53], or can be artificially sampled [37-42, 46, 47, 49-51]. The main advantage of presenting a training data point, is that it is a real-life example. It is therefore inherently realistic. However, training sets can be thinly populated, which means that the most similar counterfactual might still be widely different from the input that you are comparing to. The sampling-based approaches usually do not suffer from this problem. They either create a dense area of sampled data [38, 39, 41, 46], or take the input and perturb its feature values until a different outcome is reached [37, 40, 42, 47, 49-51]. While most tackle sampling by randomly changing feature values from the input instance or the training data [38-40, 46, 47, 49, 51], some take a more sophisticated route by using a genetic algorithm [37, 41, 42, 50]. Genetic algorithms generate instances from a starting 'population' such as the training data, or the input instance. These are then 'evolved' through crossover, mutation and selection. Crossover refers to combining feature values from two individuals, while mutation randomly changes an arbitrary feature value. By selection, only the samples with the best fitness score are kept. This fitness score is usually defined by the distance to the input. Some approaches also consider other objectives, such as sparsity [37, 40, 46]. This is usually implicitly incorporated by the previously mentioned methods that start from the input instance and adjust features one at a time until the target prediction is reached. This will keep the counterfactuals inherently sparse. Others have implemented sparsity as a constraint [44-46]. The difficulty with the latter is how to define beforehand how many differences between the input and counterfactual are allowed or even plausible. This can vary strongly between various domains, datasets, and users. Another approach is to edit produced counterfactuals back to the input instance until the target prediction no longer holds [40]. The risk here is that there is no guarantee the counterfactual can be made more similar to the input.

One method has identified that the search for counterfactuals can be tackled by using Multi-Objective Optimization (MOO), such that several scores can be optimized simultaneously [37]. In this way, the multiple objectives do not have to be enforced through summing them together, adding constraints or filtering steps, but can be included to find a Pareto optimum set of solutions. This set consists of instances with different trade-offs between the scores, and are non-dominated. What this entails is that for each of these instances in the set, there exists no better alternative; there cannot be an improvement for one objective, without decreasing the score for another objective.

Some approaches have leveraged the power of SHAP values to create counterfactuals [36, 49]. By only changing the features from the input instance that have negative SHAP values that work against the target prediction B, a counterfactual could be found [49]. This approach suffers from the fact that by only changing features with negative SHAP values, they limit the

range of possible feature changes and therefore produce counterfactuals that are generally further away. In a similar approach, features with the highest SHAP values for the predicted class A were iteratively set to zero, until the target class is reached [36].

An aspect of generating counterfactuals with sampling-based methods that is largely overlooked or handled poorly is realism. As the samples are often generated by randomly changing feature values, or by combining instances, they might be infeasible. For example, a generated instance in the context of loan applications might be a 20-year-old person with 15 years of working experience as an ideal candidate for a loan. This example obviously does not represent a real-life situation. A counterfactual example must be a plausible data point to make the user see its real-life value. Note that this issue does not frequently occur with counterfactuals derived from the training data, which are inherently realistic.

There have been some attempts to create plausible counterfactuals. These mostly rely on the assumption that features are independent. For example, to give a general impression of the relation to the training data, the distance to the closest training data point can be measured [37]. By taking this score into account, found counterfactuals are generally closer to the training data. It is also possible to look for counterfactuals that lie in dense, connected areas of the training data [43]. This ensures that the query instance can be transformed to take on the target output, which is relevant for actionable settings. Similar to these approaches, the range of feature values can be limited [39, 41]. Either based on the training data, or inputted by the user. When considering our previous example, there are most likely plenty of 20-year-old people, and also people with 15 years of working experience in the training data. However, the issue with this example is that age and working experience are correlated, and the combination of the feature values is highly unlikely. None of the previously discussed techniques take correlation into account.

Some efforts have been made to handle correlated data, though these mostly leave the responsibility to the user. For example, the user can supply causal graphs between features to model certain feature correlations [40]. These graphs are then applied to filter the already-generated counterfactuals to remove any that do not comply. This could mean that no counterfactuals remain, as the filtering happens after the generative process is completed. An implementation called GeCo has shown promise by limiting feature combinations to the ones made in the training data, though again the user needs to supply each of these relations manually [42].

One method derives counterfactuals from training instances, which relies on the assumption that there are inherently sparse counterfactuals in the training set [45]. As they point out themselves, this will most likely fail on more real-life datasets as there are often more feature differences than they deem fit (< 2).

Though several studies have brought up the issue there should be a way to handle correlated features [39, 40, 54, 55], no method has been published that inherently adapted this in a way that is viable for real-life data, without the need to manually model feature relationships. To the best of our knowledge, we are the first to develop a method that is intrinsically suitable for real-life datasets with correlated features.

Finally, improving the visual presentation of counterfactuals is regarded helpful to the users. Most counterfactual methods for tabular data present the comparison of the input and counterfactual in a table [37, 40-42, 44, 45, 47, 53, 56]. This does not clearly communicate the magnitude of the feature value differences between these instances without the user having to do arithmetic. Similarly, this effect is also apparent in explanations from a conversational statement or natural language [48, 57, 58]. With a visual approach, these magnitudes can be communicated better. [59]. Though some previous visualizations were developed for counterfactuals [46, 51], it was unclear for which audience these were fit and how well they worked for those users. Furthermore, no visualization has incorporated feature attributions with counterfactuals, which could be beneficial to form a complete picture of the prediction [51, 59].

1.4 Contribution

With this paper, we aim to demonstrate the value of XAI to the field of forensic science by applying it to a real-world use case. We generate explanations for individual predictions of the Number of Contributors (NOC) to a DNA profile, which can be applied to any type of machine learning model. These explanations consist of SHAP values and a counterfactual example in a compound visualization which we have found to be the first explanation that unifies these techniques. We also implemented a new method for finding realistic counterfactuals (ReCo), which to the best of our knowledge is the first technique that automatically handles correlated data, while creating sparse counterfactuals. Lastly, we have created a new realism metric that scores counterfactuals on the plausibility of their feature combinations.

2. Materials and methods

2.1 Data analysis and sampling

The used dataset originates from a previous study that the Netherlands Forensics Institute (NFI) performed [18]. It initially consisted of 590 PowerPlex® Fusion 6C (PPF6C) profiles, either from a single donor, or from a mixture of up to 5 donors. The mixtures were formed from 1174 different single donors that were mixed in various proportions and using various amounts of DNA to create profiles that are representative of real casework. The ground-truth NOC was therefore available. Each profile x was translated into 19 features consisting of allele counts,

allele frequencies and peak heights such that $x = \{x_1, \ldots, x_{19}\}$. These are all numeric variables which can be found in more detail in Supplementary Table 1.

The original dataset was expanded with 5000 samples to ensure a higher density of samples in the feature space. In a development version of the statistical library DNAStatistX [60], realistic profiles can be generated by using the same model that is used for calculating weights of evidence. Note that DNAStatistX implements an algorithm to calculate the Maximum Likelihood Estimate which is largely based on the source code of the probabilistic genotyping system EuroForMix [12]. This program was used to generate factors such as peak height, degradation, and mixture proportions within ranges derived from the original dataset. Note that elevated stutter peaks were not simulated. However, the probability of drop-in was set quite high at 0.05 by which the simulated DNA profiles could include additional peaks, not belonging to one of the donors, as can occur under casework circumstances. In Supplementary Table 2, the exact parameters can be found. With these parameters in place, the genotypes are generated randomly based on Dutch population frequencies [61]. To ensure that all donors have at least some of their alleles observed in the generated profile, we chose to set the requirement that each donor must have an LR of at least 1000 when computed using DNAStatistX.

The advantage of sampling before applying any XAI technique is that the profiles are generated, not the derived features. In this way, validated software is used to generate as plausible as possible profiles from which features can be calculated afterwards. The features used are strongly correlated (see Supplementary Figure 5), which would make sampling in a later step more difficult.

Once the features were derived from the sampled data, about half of them appeared to have been drawn from a different distribution as compared to the original dataset of 590 instances (see Supplementary Figures 3 and 4, and Supplementary Table 3). For instance, the TAC and MAC values of the sampled data appear to be slightly higher, implying neater, easier to interpret data. On the other hand, the variation in allele counts and peak heights is larger, adding more diversity in the data. Because of these discrepancies, we tested the value of the simulated data in a benchmarking study, which demonstrated that the model actually performs better once trained on the combined dataset of the original and simulated datasets together (see Supplementary Table 4 and Supplementary Figures 8 and 9, in comparison to Supplementary Table 5 and Supplementary Figures 6 and 7).

2.2 Machine learning model

Originally, the estimation of the NOC was treated as a classification problem by the NFI with the RFC19 model such that $f(x) = y$ where x is an input profile consisting of the 19 features [18]. The model f is a random forest classifier (titled RFC19), which produces an

output y with five categories such that $y \in \mathbb{N} \mid 1 \leq y \leq 5$. They obtained a test accuracy of 82.5%.

Since the outputs of the model are ordinal, the problem could benefit from being tackled with a regression model. After a short benchmarking study with a default random forest regressor (see Supplementary Figures 6 - 9 and Supplementary Table 4), we concluded that a regression model has the potential to achieve more accurate predictions. The regression model in combination with the larger dataset even improved performance on the profiles that originated from the original dataset (see Supplementary Table 6). This shows that the model performs well on real profiles, and not just on the simulated instances.

Explanations can also benefit from using regression, as the classification approach ignores the ordinal relation between the outputs, which is apparent to the user. With regression the output contains decimals such that $y \in \mathbb{R} \mid 1 \leq y \leq 5$. In this study, for the purpose of introducing XAI to a NOC machine learning model, we chose to continue with the regression model trained on the combined dataset (RFR19_merged) though the XAI method will be applicable independent of the type of machine learning model.

2.3 Explanation goals

The most pertinent case for which explanations of the machine learning model are helpful, is when the DNA expert comes to a different conclusion than the model. It can be unclear why this discrepancy exists when only the model output is available. It is possible that the user missed some information that the model based its decision on, or perhaps the model made its prediction based on the wrong factors. Presenting explanations that provide a sense of thresholding values and how close the decision is, can help the expert make a more well-informed decision on which result to trust. Explanations can also be informative in straightforward cases as a confirmation of the user's own estimation of the NOC, by presenting general information on the model's focus. These scenarios line up nicely with the two questions that a good explanation of a single prediction should answer [21, 38, 39]:

1. *What were the main reasons for the model to reach the current prediction?*
2. *With which feature changes could the model have arrived at a different prediction?*

To answer these two questions, we consulted a study that has identified which types of explanations work best for which types of questions [62]. From their definitions, the experts would like to have "WH-X" and "WH-NOT-Y" questions answered, which correspond to questions 1 and 2 listed above. These can be answered by an explanation of what features of the profile contribute most to the prediction, and a counterfactual explanation demonstrating what changes in this profile would lead to a different prediction.

To answer question 1, we determined that the use of SHAP values would be sufficient to give an impression of feature importance. We acknowledge that all perturbation-based feature importance methods arbitrarily split the impact of correlated features on the model [63]. The result of this issue is that the importance values for correlated contributing features are underestimated, in contrast to if their importance was left undivided. However, since the main goal of these values is to give an impression of the contributing factors to a prediction, the exact values are not a priority. This part of the explanation is to observe a general sense of which features contributed to the prediction in which direction. For this purpose, we deem SHAP adequate. For the second question, we have developed a new counterfactual technique, for which we performed a more in-depth analysis of the requirements.

2.4 Desiderata counterfactual explanations

To develop the most suitable counterfactual method, we derived a list of desiderata that it must accommodate. These requirements originated from the factors discussed in section 1.3, in combination with the explanation goals we defined in section 2.3. All desiderata are discussed below.

- Model-agnostic — Can be applied to any model
- Interactive — Target output can be chosen by the user
- Valid — Target output must always be reached
- Sparse — Minimal feature differences between input and counterfactual
- Proximal — Minimal distance between input and counterfactual
- Realistic — Plausible combinations of feature values in counterfactual

As we are looking to continue development on the machine learning model, a *model-agnostic* explanation method is preferred. In this way, the same explanations can be generated regardless of the underlying algorithm. We assume to have access to the predictions of the model.

Most existing methods assume a binary prediction problem, meaning the target output is the opposite of the current prediction. In our problem however, the range of possible values is 1-5. It is not always straightforward to pick the next-best option; different users determine different ranges of possibilities. We therefore let the user pick the target through an *interactive* prompt.

It should be possible to generate a counterfactual for any input. If the closest counterfactual example seems incomparable to the input profile, that shows a limitation of the dataset. This is not inherently bad; it could even provide the user some insight in how the model works. We have designed counterfactual targets to be integers between 1-5 to match directly with the

NOC that DNA experts have to report. Since the current model uses regression, we consider instances with a rounded-off prediction that match the target to be *valid* counterfactuals.

Sparsity is encouraged to prevent users from experiencing cognitive overload. We know that humans pick explanations in a biased way [22], meaning that if many options are available, only a few will be selected. The number of different feature values between the input and counterfactual can be counted using L_0 norm as shown in Equation 1.

$$\mathrm{fd}(x, x') = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[x_i \neq x'_i] \quad (1)$$

Where n represents the number of features, x the profile to be explained, and x' the counterfactual profile.

For the distance between the input and counterfactual, we first analyzed the underlying data. The choice of distance should be catered towards the problem [38]. As our dataset has outliers, and most features are not normally distributed (see Supplementary Figures 3 and 4), L_1 distance is more appropriate. With L_2 distance, outliers can get blown out of proportion. Though many counterfactual methods scale distance by the Median Absolute Deviation (MAD) [38, 40, 45, 46, 48], this is not appropriate for the current dataset because not all features are normally distributed. If a feature with much larger variation than the MAD were to be scaled this way, the distance score would be dominated by that feature. Therefore, we scale with each feature's range to minimize the influence of different ranges, variations, and distributions [37, 39]. This is more robust for unscaled and unnormalized features with lots of outliers, which is the case in this dataset. This distance measure is shown in Equation 2.

$$\mathrm{d}(x, x') = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\hat{R}_i} |x_i - x'_i| \quad (2)$$

Where $\hat{R}_i$ represents the range of the i-th feature, n the number of features, x the profile to be explained, and x' the counterfactual profile. It has an additional property that $0 \leq d(x, x') \leq 1$. It can also be used alongside categorical variables by replacing $\frac{1}{\hat{R}_i}|x_i - x'_i|$ with $\mathbb{I}[x_i \neq x'_i]$.

Since none of the features of a DNA profile can be changed to reach an alternative prediction, actionability is not a goal of this method. Similarly, we do not strive to present a set of diverse counterfactuals as diversity is often encouraged for a similar purpose as actionability; to provide a user with multiple routes to reach the different outcome [38, 42, 48].

Moreover, presenting multiple, possibly contradicting examples does not seem like a user-friendly introduction to counterfactual explanations.

The desiderata discussed so far have been covered consistently in the literature. For realism, there is not such a proper definition. Within the problem of NOC estimation, it is essential to present the user with data that is plausible. None of the sampling methods discussed in section 1.3, are automatically suitable for datasets with correlated features, as they would produce unlikely feature combinations. For example, a TAC of 150 is impossible in combination with a MAC of 2 for this kit (from which 23 loci are used in this study), even though these are both normal feature values when looking at the feature distributions. Other approaches that place constraints on the sampled data are too time-consuming, as modelling the relationships between features is not trivial. Since the features might also change in the future, it is not practical to invest time in modelling their relations currently. Instead, we turn towards the training data which inherently consists of the most realistic instances to use. We therefore regard this a good starting point for our explanations.

2.5 Realistic Counterfactuals (ReCo)

To fulfil all previously defined desiderata, we developed an algorithm called Realistic Counterfactuals (ReCo). Instead of generating data and then filtering instances that are infeasible with respect to the training data, ReCo starts with the training instances and forms them into sparser counterfactuals. ReCo therefore consists of two parts: First, the most suitable counterfactual training instance is found. Second, that counterfactual training instance is made sparser by applying a filter.

Finding the most suitable counterfactual training instance: From the input profile x and its prediction $f(x) = y$, where f can be any machine learning model, the user defines a target prediction $y' \neq y$. ReCo then finds all instances x^* from the training data with the target prediction $f(x^*) = y'$. This prediction must match with their ground truth NOC so that no incorrect predictions are presented as examples.

ReCo then finds the set of non-dominated instances with regards to sparsity and distance. By minimizing both objectives simultaneously, the obtained Pareto set of counterfactuals has optimal trade-offs between the two scores [37]. As we intend to present a single counterfactual, we select the median instance x' from this set which balances the two scores best as can be seen in Equation 3.

$$x' = med\left\{min_{x^*}\left(\mathrm{fd}(x, x^*),\ \mathrm{d}(x, x^*)\right)\right\} \quad (3)$$

Where $\mathrm{fd}(x, x^*)$ and $\mathrm{d}(x, x^*)$ are defined in Equation 1 and 2 respectively. Additional objectives could be added if deemed important in the future, and the selection from the set can be adjusted if a certain score is preferred over another. Objectives can also be compared without any normalization as is required with for example a weighted sum where balancing scores is dependent on their variance and mean [64, 65].

The counterfactual instance x' is part of the training data, making it a realistic data point to present. However, such an instance has the following disadvantages:

- Lack of sparsity: the training instance has many different feature values as compared to the profile we want to explain.
- Lack of relevance: not all of these differences are informative to arrive at the target prediction.

ReCo tackles both of these issues by applying a filter to the found counterfactual instance, selecting only the most relevant feature value changes.

Filtering the counterfactual training instance: Filtering is defined by the following five steps. Table 1 shows a practical example.

1. Start by finding the set of features that have different values between the input x_i and the counterfactual x'_i. The size of this set can be a maximum of n, the number of features of which an instance consists. In Table 1, there are three features in this set.

$$\text{differences} = \{ \forall i \in \mathbb{N} \wedge 1 \leq i \leq n \mid x_i \neq x'_i \} \quad (4)$$

2. Compute the SHAP values for both the input instance and the counterfactual instance, per feature in differences. Subtract the SHAP values of the input instance from the SHAP values of the counterfactual instance. This set is then sorted by the elements' magnitudes. This gives us an impression of which changes in feature values from the input instance to the counterfactual instance have impacted the change in prediction the most. The biggest positive or negative SHAP changes have likely made the most impact on the change in prediction. In Table 1, the SHAP change of Feature 1 is largest, while it is the smallest for Feature 3.

$$\text{SHAP\_change} = \{ \text{SHAP}(x'_i) - \text{SHAP}(x_i) \mid i \in \text{differences} \} \quad (5)$$

3. To make the counterfactual instance sparser as compared to the input instance, we need to remove the irrelevant feature differences. If the prediction goes down from the

input to the counterfactual, or becomes more negative, we expect the features with negative SHAP change to be most relevant. On the other hand, positive SHAP changes are defined to be misaligned with the change in prediction in this case. This is listed in the bottom row of Table 1; the change in Feature 2 is misaligned. We also include very small SHAP changes such as for Feature 3. These feature differences are most likely not relevant to help reach the counterfactual prediction, and could therefore possibly be filtered from the counterfactual instance.

$$\text{misaligned} = \begin{cases} \{\, i \mid \text{SHAP\_change}_i > -\epsilon \,\}, & \text{if } f(x') - f(x) < 0 \\ \{\, i \mid \text{SHAP\_change}_i < \epsilon \,\}, & \text{otherwise} \end{cases} \quad (6)$$

4. The next step is to check if the feature differences with misaligned SHAP change can be removed. 'Removing' in this context means that the feature value of the counterfactual x'_i is replaced with the feature value of the input instance x_i. If the prediction of this filtered counterfactual $f((x' \setminus \{x'_i\}) \cup \{x_i\})$ stays the same as the target y', it is labelled as irrelevant_diff.

$$\text{irrelevant\_diff} = \{\, i \in \text{misaligned} \mid f((x' \setminus \{x'_i\}) \cup \{x_i\}) = y' \,\} \quad (7)$$

5. Once removing the next feature difference causes a different outcome than the target prediction, filtering stops. All irrelevant features differences are filtered from the counterfactual so that the final counterfactual is defined as:

$$\text{counterfactual} = x' \setminus \{x'_i\} \cup \{x_i\} \mid i \in \text{irrelevant\_diff} \quad (8)$$

	Feature 1	**Feature 2**	**Feature 3**
SHAP value in input	0.300	-0.200	0
SHAP value in counterfactual	0	-0.150	-0.001
SHAP change	-0.300	+0.050	-0.001
Candidate to be filtered from counterfactual?	No	Yes	Yes

Table 1: Example of how a counterfactual is filtered. The input instance has a prediction of 4, and the counterfactual has a prediction of 3. Therefore, the direction of the change in prediction is negative. Features 1-3 are the features that differ between the input and counterfactual. Their SHAP values are calculated for both the input and the counterfactual. For Feature 1, the SHAP change is negative, matching the direction of the change in prediction. In contrast, the SHAP change in Feature 2 is positive, and the SHAP change in Feature 3 is small. These last two differences in feature values are therefore likely not relevant to the counterfactual prediction, and thus are candidates to be filtered.

Note that even though we directly use SHAP values to determine whether or not a feature value can be 'removed', we are aware that these SHAP values can be underestimated for

correlated features. However, ReCo mainly relies on the direction of the SHAP value, so whether it positively or negatively contributes to the prediction. The SHAP values will not become negative while the true value is positive. Therefore, these inaccuracies are not as important to our method. Still, if a feature value difference is marked to be irrelevant though it was impactful for the model, ReCo always checks the prediction before removing it from the counterfactual.

Although the current implementation of ReCo is used for regression, it can be used for classification as well. In this case, we do not have to consider the direction of the change in prediction, we only determine if the change in feature value corresponds to more positive SHAP values for the counterfactual class. If that is the case, the counterfactual feature value is kept, otherwise the input value could remain.

2.6 Realism score

We present a novel realism score which can be used to evaluate counterfactuals. This score assesses whether a generated counterfactual has feasible combinations of feature values in relation to the training data. It is calculated as follows:

1. When the dataset is loaded, a list is generated for each feature that ranks all other variables according to their correlation with the feature.
2. When a counterfactual is found, each feature that has a different value than the original instance is assessed. We will refer to this feature under investigation as F_{diff}.
    a. The feature's top correlated variable F_{corr} is looked up from the list in step 1.
    b. We check that the value $F_{diff} = f_{diff}$ in combination with the value $F_{corr} = f_{corr}$ exists in the training data. If so, add 1 to the realism score. If not, add 0.
    c. If F_{corr} was also part of the set of features that differs between the original and the counterfactual instance, we return to step a. and pick the next most correlated feature with F_{diff} to be F_{corr}. In this way, the score is always grounded in the values of a real instance.
    d. The total realism score is normalized by dividing by the number of features that were scored.

Please refer to Figure 1 for an example. In this case, instances only consist of a TAC and a MAC value. The counterfactual only has a different TAC value from the original instance, so we need to check if that generated TAC value is plausible. The most highly-correlated feature to the TAC is the MAC. We assess if the combination of TAC = 30 (from the counterfactual) with MAC = 6 (from the original) exists in the training data. Since it does not exist, the realism score is incremented by 0. The MAC feature is not part of the differences between the

counterfactual and the original, so the algorithm terminates. The final realism score for this counterfactual is 0.

Original	Counterfactual
MAC = 6	MAC = 6
TAC = 98	TAC = 30

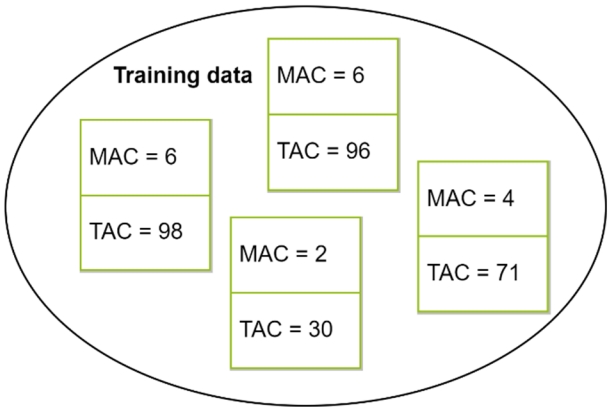


Figure 1: Example of a counterfactual that receives a realism score of 0; the proposed counterfactual contains a feature combination that does not occur in the training data.

2.7 Set-up quantitative evaluation ReCo

To determine the quality of ReCo, we have assessed it on the metrics defined by the desiderata described in section 2.4. As our method is model-agnostic and valid by design, and interactivity is a built-in feature, we chose to focus on the three remaining metrics of sparsity, proximity and realism. Sparsity and proximity seem to be a standard for evaluation of counterfactuals [36, 37, 39, 40, 42, 44-46], whereas the metric for realism is not as clearly defined. Proximity to the training data is often used as a score of realism, though we argue that our realism metric defined in section 2.6 reflects this purpose better. We will present both of these for comparison. To re-iterate:

- Sparsity concerns the number of feature differences between the input and counterfactual and is measured using L_0 distance (Equation 1).
- Proximity relates to the distance between the input and counterfactual. We measure this according to range-normalized L_1 distance (Equation 2).
- Proximity to the training data is the distance between the counterfactual and the closest training instance. We measure this according to range-normalized L_1 distance (Equation 2).
- Realism measures if the feature combinations of the counterfactual are present in the training data. It is measured according to the realism score proposed in section 2.6.

These metrics are used to compare ReCo against three other counterfactual methods [40]. As constraints, we have chosen methods that are model-agnostic, suitable for regression, and

suitable for numeric tabular data. *WhatIf* is our own implementation of Google's What-If tool for searching the closest counterfactual from the training data [53]. We implemented it with our range-normalized L_1 distance. *DiCE random* is a sampling approach that generates counterfactuals from the input by randomly sampling its feature values [40]. It starts from the input instance, and randomly picks a feature to be given a sampled value until the target prediction is reached. For this implementation, we used default parameters and set the target prediction between $\text{target} - 0.5$ and $\text{target} + 0.4$. The algorithm automatically takes the minimum and maximum values of each feature into account. Lastly, we compare with *DiCE genetic* which implementation is similar to GeCo [42], as it generates counterfactuals from using a genetic algorithm. The algorithm starts from training instances with the target prediction, and *evolves* them to form new samples. When generating a new instance, two training instances are used as its *parents*. This means that for each feature, it can either take the value of instance 1, instance 2 (*crossover*), or a random value is assigned (*mutation*). Through selection of the best instances with respect to sparsity and proximity, a counterfactual is found. We also used the default implementation for DiCE genetic.

For all these methods, the target is set to the second most likely prediction so that the process runs automatically. For instance, if the test prediction is 3.2, the counterfactual target is set to 4.

2.8 Set-up visualization

We incorporated both SHAP values, and the counterfactual example generated by ReCo into a single figure so that the user can understand the main reasons for the original prediction, along with how a different outcome could have been achieved. We believe this is the first visualization to unify counterfactuals with SHAP values. The following requirements were considered from conferring with the consulted DNA experts in addition to some desiderata already expressed in the literature [66, 67].

First of all, the visualization is *consistent*. Each profile is presented in the same format; SHAP values on the left and a counterfactual example on the right. The same features will always be on the same location as well. This consistency helps users reach some level of familiarity with the visualization over time as it allows for comparison between profiles.

The feature values are also plotted on a normalized scale to get a visual representation of how large a value is compared to the range of possible values. For normalization, we used a quantile transform as this maps all feature values between 0-1 while spreading out the most frequent values [68]. As we have observed in Supplementary Figures 3 and 4, many features have a skewed distribution and contain outliers. For the visualization, that would make some values difficult to distinguish since the scale is warped by outliers. The quantile transform

smooths the relationship between the observations, making the variation between the more common values more evident.

Secondly, the explanation is *contextualized* with informative text about the current prediction, and the conditions of the two parts of the explanation. In this way, the user understands for which conditions the explanation holds. By encoding the two separate explanations with different color palettes, a distinction is made between the SHAP values and the counterfactual. Only the counterfactual differences will be shown with arrows as they indicate changes. The used color palettes are specifically chosen to be *accessible* as they are distinguishable to the color-blind [69].

Lastly, some *interactivity* is introduced by letting the user choose the counterfactual target.

2.9 Set-up user study

It was important to evaluate the visualization from the perspective of the end-users, an aspect often brushed over in XAI studies [56, 59]. The explanation was specifically designed for DNA experts within the context of NOC estimation, so we invited DNA experts from the NFI to participate in a user study.

We did not use this survey to see if users can more accurately determine the NOC as this is the experts' initial introduction to any XAI implementation, and as such require more training and experience to properly use it as a decision-making tool. The data on which the explanation is based is also not fully understandable as many of the features are too abstract for users to see how they relate to NOC estimation. Instead, the evaluation was set up around two simpler aspects; the first was to see if users can gain insight into the predictions of the model, and by extension, if that information helps regulate the users' trust. The second aspect concerned how user-friendly the explanation is.

For the exercise on trust, we selected two exemplary profiles for two use-cases. Profile 1 was fairly simple for the model to predict, where we intended the explanation to increase trust in the model prediction. Profile 2 was difficult for the model, leading to an erroneous prediction. In this case the explanation was meant to make the user doubt the model prediction. We measured trust with two questions:

1. Which number(s) of contributors do users consider?
2. Do users think that the prediction is correct?

As a baseline, we ask these questions when users are only presented with the prediction. Then we ask them once again after a state-of-the-art explanation, and once after our visualization. In this way, we compare against readily available explanations. If users trust the prediction more after seeing the explanation, we expect them to be able to pinpoint the NOC more, and think the prediction is (more) correct. For profile 1, we compared our visualization against a SHAP force plot [70]. As SHAP is designed for users to understand "why a model

makes a certain prediction", we deemed it fit for the goal of increasing trust. For profile 2, we compared our visualization against a counterfactual table, as this representation is common for counterfactuals [37, 40-42, 44, 45, 47, 53, 56]. As counterfactuals show how a different prediction can be reached, it can decrease trust in the original prediction if that change seems small. To keep the survey short, we did not compare the visualization against SHAP and the counterfactual (CF) table for both profiles.

Before these questions were asked, all explanations were introduced with a video, visualization and bullet points to ensure that the participants understand the presented information. With a qualification test, we checked that the participants had completed the introduction.

Within the section about user-preference, we asked users to pick their favorite explanation based on three aspects: *ease of use* (how easily users could find the relevant information), *appeal* (how nice users thought it was to use), and *completeness* (how well users could form a total picture of the prediction). The aim was to determine if the participants had an absolute preference for any of the explanations they had seen.

3. Results and discussion

This work presents two distinct products; a new counterfactual method ReCo, and a visualization combining the results from ReCo with SHAP values. We show the results of the objective evaluation of ReCo, after which we present the visualization and the corresponding user study results. As both of these aspects specifically target NOC estimation, only the dataset described in section 2.1 was used for the evaluation.

3.1 Quantitative evaluation ReCo

The obtained scores on the test data for the four methods can be found in Figure 2.

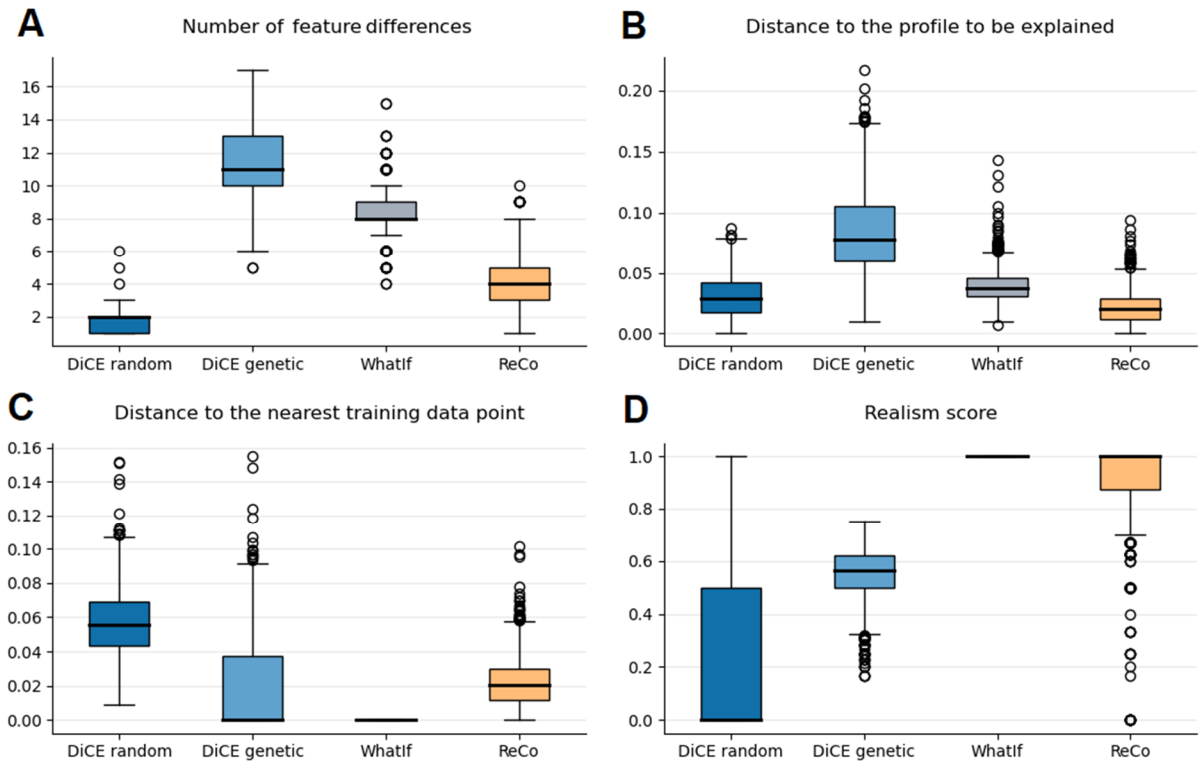


Figure 2: Quantitative evaluation of ReCo in comparison to WhatIf, DiCE random and DiCE genetic on four different metrics; the number of feature differences (A), the distance to the input (B), the distance to the training data (C), and realism (D).

The WhatIf method could be seen as a baseline, using only existing training examples as counterfactuals. Its realism score and distance to the training data are therefore perfect, but it suffers from many feature differences and a higher distance score due to the sparsity of the training data.

While DiCE random performs best in terms of the number of feature differences, and quite well on distance, it performs poorly on realism and is the furthest away from the training data. This is because DiCE random starts from the original instance, and perturbs a random feature until the target prediction is reached. This strategy helps keep the number of feature differences and the overall distance score low, but does not in any way account for the relations between the features. This makes this method inappropriate for our dataset.

An improvement can be seen when the genetic version is used (DiCE genetic); the median realism score almost hits 0.6, and the distance to the training data is practically zero. We can attribute these better scores to the combination of profiles from the training data. However, this crossover still simply combines the feature values of two instances, which can create unlikely feature combinations. Mutation has a similar effect. It is interesting to see that this algorithm leads to significantly larger distances and more feature differences. It could be that by combining training instances, the newly formed amalgamation becomes more generalized for

the target prediction and as such, moves further away from the input. One final aspect to note about both DiCE techniques is that they failed to generate a counterfactual for about 2% of the test inputs, thereby failing our desideratum for validity.

ReCo seems to score relatively well on all four metrics. As the method first finds the closest and most sparse training instance, this is an inherently realistic starting point. Because both sparsity and distance are optimized, in contrast to WhatIf, which only minimizes the distance, the obtained data points might already be sparser. Then by filtering, these two scores go down further whenever it is possible. The reason that we can filter so many differences without moving too far away from the training data and producing unlikely feature combinations, could be explained by a number of factors. First of all, the filter removes small or counterintuitive differences that are likely insignificant to the model. These limited differences will not cause the counterfactual to move too far away from the training data. Secondly, the features that are filtered could have little discriminatory power between the original and target output. This could be because their values are similar for instances of the original and target prediction in the training data. For example, if for both the original and the target NOC, the median of a feature in the training data is equal, it possibly has little discriminatory power between the two outcomes. As a final remark, we note that there are some outliers that score low on realism. When more feature differences are filtered, the likelihood increases that the values from those input features do not match with the leftover counterfactual feature values. Also note that the current realism metric is strict; it does not check if a feature value is close to known combinations in the training data, the values must match 100%. It might be interesting to see if adding a tolerance to this metric creates a more nuanced score, but we leave this for future work.

3.2 Visualization

The visualization for the explanation of a single DNA profile prediction is depicted in Figure 3.

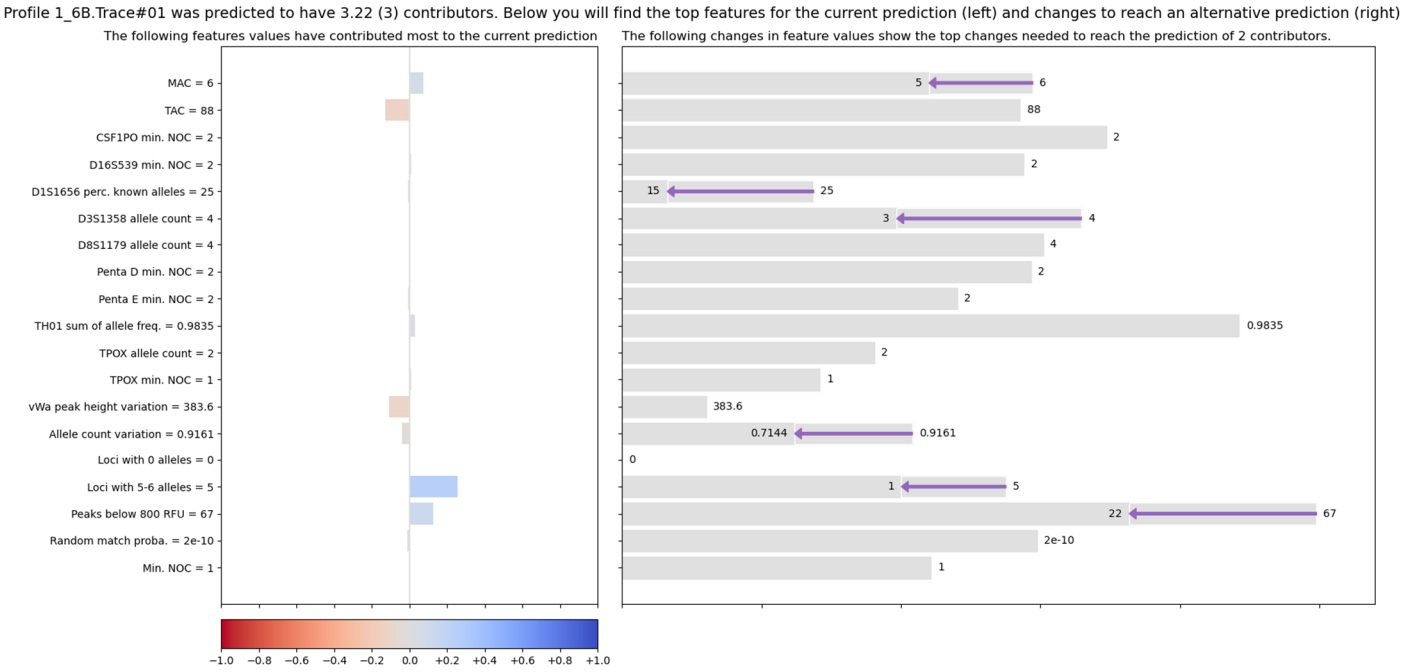


Figure 3: Visualization for the explanation of a profile with 3 contributors, that was correctly predicted to have 3 contributors (profile 1 in the user study). Its feature values are listed on the left and plotted on the right. SHAP values are depicted on the left with red and blue bars, and a counterfactual example generated by ReCo for a prediction of 2 contributors is shown on the right with arrows.

The top line informs the user about the current profile and what the model's prediction is. The decimal output of the regression model can be used to give the user an impression about the certainty of the prediction; any value ending in .49 or lower will be rounded down; any value ending in .50 or higher will be rounded up. The top line further includes a summary of what information can be found in the figure. On the left-hand side, all 19 features and their values as defined by this profile are listed. These same feature values also appear in the right section as normalized grey bars, aligned with the feature values on the left.

The SHAP values are visible in the left section; red bars mean that the feature values pushed the prediction down, while blue bars represent feature values that pushed the prediction up. Note that SHAP starts from a base prediction of 3 contributors; this is the mean outcome value out of the 1-5 range. Starting from a prediction of 3, adding the SHAP values together forms the current prediction of 3.22. In this case, there are twelve feature values influencing the decision, though only about six or seven are clearly visible. We intentionally only added the SHAP value legend at the bottom as we do not want the users to focus on the exact values, but on the direction and relative size instead as it is possible that the exact values are underestimated due to the correlations between features. For this prediction, the model noticed this profile's higher values of *MAC*, *loci with 5 and 6 alleles*, and *peaks below 800 RFU* (the stochastic threshold that applies to the data in this study) as indicators for more

contributors. More alleles per loci indeed imply more donors, and lots of low peaks indicate a profile with less quantity of information which can be more prevalent with higher-order mixtures. In contrast, the TAC and peak height variation at locus vWA have lower values that typically occur in lower-order mixtures.

To generate a counterfactual explanation for this profile, we have set the target at 2 contributors. As often a minimum NOC is reported, it might be interesting to be able to rule out this option, and instead go with the prediction of 3 contributors. Within the application, the user can normally first explore the factual explanation consisting of the features and SHAP values before choosing a counterfactual target. The counterfactual that ReCo has found for this explanation has six lower feature values as denoted by the purple arrows. If any features would need to increase their value, the arrow would be olive-colored. The arrows demonstrate all the changes that are required to reach the target prediction. Three of the arrows relate to the three feature values that we discovered were pushing the prediction up (MAC, loci with 5 and 6 alleles, and peaks below 800 RFU). By adjusting these values, along with the other three feature values, a lower prediction can be achieved. It seems that to reach this target of 2 contributors, many features need to change, and by a large extent. This can provide an indication that the model is fairly certain that the NOC is not 2.

3.3 User study results

The survey was quite extensive since it includes introductions for three types of explanations. As such, we expected that the response would be limited. In total, 7 useable answers were collected from DNA experts of several age groups from 18 to 54. One response had to be eliminated as they failed the qualification tests. Because of the limited size of the group, we treated the obtained results as a subjective collection of the participants' opinions. The results of the first exercise about trust can be found in Figure 4. It presents if users gained or lost trust in the prediction after seeing the two explanations for profile 1 and 2.

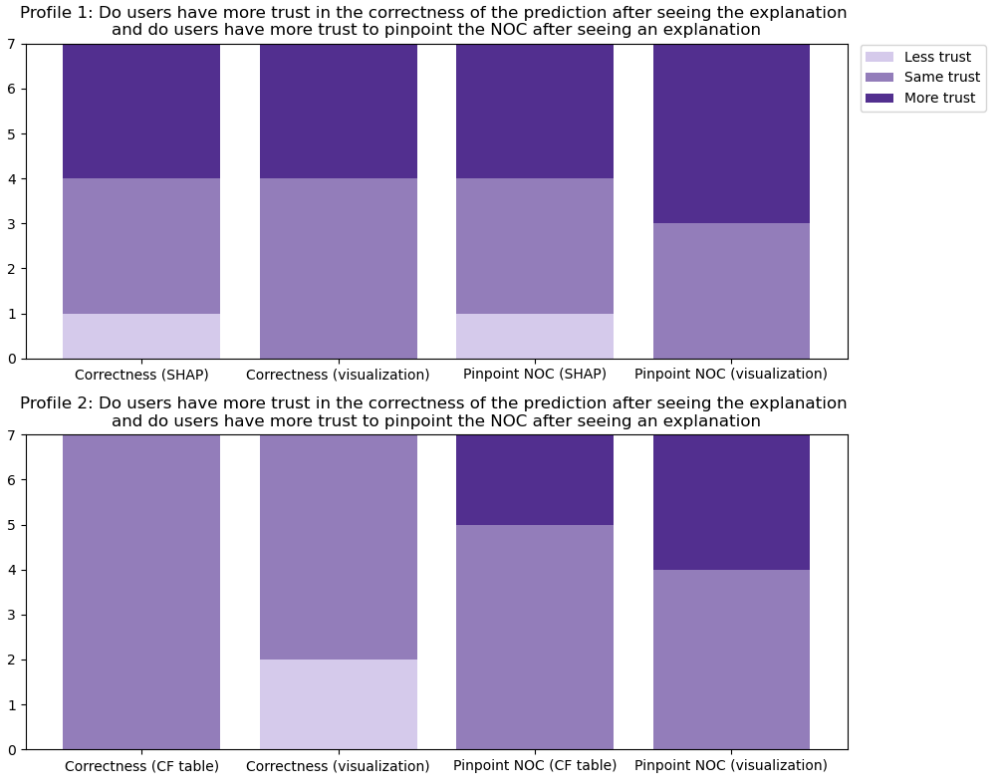


Figure 4: Results from the user study trust exercise. For profile 1 (top), it shows the influence of seeing a SHAP explanation in comparison to our visualization, on trust in the correctness of the model prediction, and on the users trust to pinpoint the NOC. For profile 2 (bottom), it shows the influence of seeing a CF table explanation in comparison to our visualization, on trust in the correctness of the model prediction, and on the users trust to pinpoint the NOC.

When the model is fairly certain about the prediction (profile 1, see Figure 3), seeing any explanation makes some users (3/7) both gain more trust in the prediction, and enables them to pinpoint the NOC better. This can be partly attributed to the fact that the feature values of the profile were first presented with the explanations. As such, a few participants (2/7) became more certain of a certain NOC because of the feature values, not because of what the SHAP force plot was trying to communicate. SHAP can induce some confusion seeing that 1 user gained less trust in the correctness of the prediction and considered a wider range of contributors. The way that the bars of the SHAP force plot work against each other, was not intuitive for some users (2/7) as they expressed difficulty with understanding it. For the visualization, most users (4/7) noted that a lot of change was required to reach the prediction of 2 contributors, and therefore dropped this outcome from consideration. One user thought that the visualization presented similar information to reach a prediction of 2 contributors as they would have thought, thereby increasing their trust in this explanation.

When the model is uncertain or incorrect (profile 2, see Figure 5), the CF table had no effect on how users perceived the correctness of the prediction, while the visualization made

some users (2/7) trust the prediction less. From the additional textual input, a lot of users did mention that they started to doubt the prediction (5/7), but not all of them changed their answer. The remarks that participants made with the visualization related most frequently to the fact that only minor changes are required to change the prediction to 3; changing the TAC from 98 to 96. DNA experts would not make a different decision depending on such a small difference in TAC value, they always use ranges. As such, the experts began to doubt whether or not the model made a correct decision. One participant even noted that for a TAC of 98, there can be 2 artefact peaks and that therefore they thought the prediction was incorrect. The visualization in general made users more confident to pinpoint the NOC as they considered less options than with the SHAP or CF table explanations (more trust to pinpoint NOC in Figure 4).

In short, it seems that our visualization provides some insight into the model, which influences how users view and trust the prediction. The users might even feel more equipped to make a narrower estimation of the NOC. Our visualization seems to be less confusing than a SHAP force plot, and more informative than presenting a counterfactual in a table. Note that because the study was limited, these results are an indication of the participants opinions and might vary once repeated with a larger group of people. Since many of the features are quite difficult to understand at this point, we did not evaluate on how well the explanation can help experts in determining the NOC. Such an evaluation can be done in the future when the model and features have been developed further.

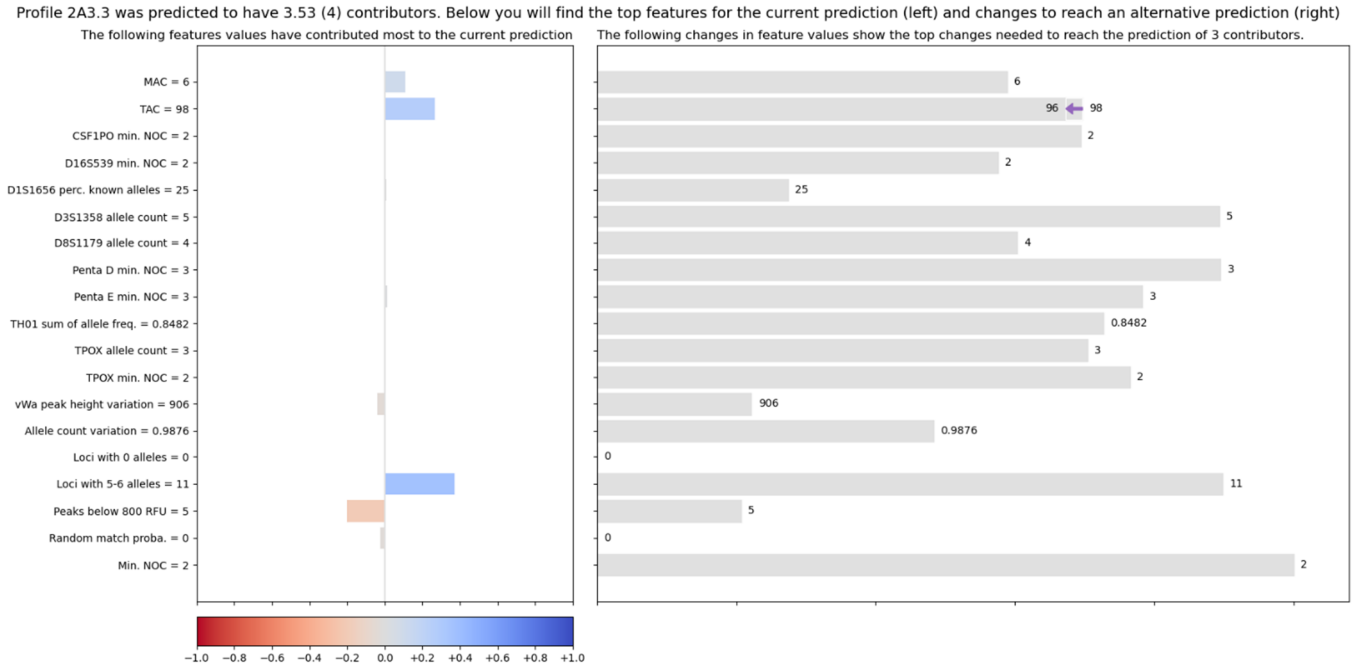


Figure 5: Visualization for the explanation of a profile with 3 contributors, that was incorrectly predicted to have 4 contributors (profile 2 in the user study). Its feature values are listed on the left and plotted on the right. SHAP values are depicted on the left with red and blue bars, and a counterfactual example generated by ReCo for a prediction of 3 contributors is shown on the right with arrows.

The results of the second task about user preferences can be found in Table 2. Our compound visualization scored the best out of the three options, though some users had a preference for SHAP for its ease of use. The experts who preferred our visualization, mostly chose it because of its visual representation, the amount of available information and because the information was easy to find.

	Ease of use	**Appeal**	**Completeness**
SHAP force plot	2	2	1
Counterfactual table	0	0	1
Compound visualization	5	5	5

Table 2: Results of user preferences. The numbers represent how many users selected each type of explanation they preferred in terms of ease of use, appeal and completeness.

3.4 Future work

As the DNA experts we consulted have indicated, the features on which the explanation are based are still difficult to comprehend. Some of the variables also encode redundant information such as *[locus] min. NOC* which is the same as *[locus] allele count* divided by 2 and rounded down. It seems that the features can be further investigated on redundancy, perhaps re-designed and expanded upon. For one, to ensure that they are understandable to users on how they relate to the NOC estimation task, and secondly that they are as informative to the machine learning models as possible.

It might benefit the NOC estimation problem to develop multiple binary models that differentiate between just two options; one for 1 or 2 contributors; one for 2 or 3; etc. This could create more specialized models, and thus more specific explanations. We refer to an implementation of such a structure for selecting the most suitable eye-surgery option for a patient [35].

Another direction of interest is to further develop the proposed realism metric. For example, by introducing some matching tolerance with values from the training data, or by comparing more feature combinations than with the top correlated variable. It could also be incorporated into the fitness function of a genetic sampling algorithm. In this way, the algorithm can optimize on generating counterfactuals with realistic feature combinations as well.

4. Conclusion

This study describes an implementation of XAI for predictions of the number of contributors of DNA profiles which can be applied to any type of machine learning model. The explanation consists of SHAP values and a counterfactual example incorporated into a compound

visualization, which has been evaluated by a small group of DNA experts. From their observations, it seems that the visualization provides some insight into the predictions of the model. We further present a method for finding realistic counterfactuals, called ReCo. ReCo creates a counterfactual by first obtaining the most suitable training instance, and then filtering the irrelevant feature value differences between this instance and the input. This produces examples that have fewer feature differences than by using training examples, and are more plausible than counterfactuals generated by sampling-based approaches. To the best of our knowledge, ReCo is the first method that can handle correlated data automatically, while still creating sparse counterfactuals. Additionally, a realism metric was defined that scores how plausible counterfactuals are in terms of their feature combinations.

Finally, we hope that this study encourages other implementations of machine learning to incorporate an XAI-component, especially when the users of such models are not familiar with the underlying concepts of machine learning.

Acknowledgements

We are thankful to Jerry Hoogenboom for generating the 5000 samples, Jennifer van der Linden for getting us up to speed with the data and model, the NFI group BiS for participating in the user studies, and Jason van Breukelen for providing general feedback.

References

[1] M.D. Coble, J.A. Bright, J.S. Buckleton, J.M. Curran, Uncertainty in the number of contributors in the proposed new CODIS set, Forensic Science International: Genetics 19 (2015) 207-211.

[2] C.C.G. Benschop, H. Haned, L. Jeurissen, P.D. Gill, T. Sijen, The effect of varying the number of contributors on likelihood ratios for complex DNA mixtures, Forensic Science International: Genetics 19 (2015) 92-99.

[3] H. Haned, L. Pène, J.R. Lobry, A.B. Dufour, D. Pontier, Estimating the Number of Contributors to Forensic DNA Mixtures: Does Maximum Likelihood Perform Better Than Maximum Allele Count?, Journal of Forensic Sciences 56(1) (2011) 23-28.

[4] A. Biedermann, S. Bozza, K. Konis, F. Taroni, Inference about the number of contributors to a DNA mixture: Comparative analyses of a Bayesian network approach and the maximum allele count method, Forensic Science International: Genetics 6(6) (2012) 689-696.

[5] D.R. Paoletti, D.E. Krane, T.E. Doom, M. Raymer, Inferring the Number of Contributors to Mixed DNA Profiles, IEEE/ACM Transactions on Computational Biology and Bioinformatics 9(1) (2012) 113-122.

[6] B.A. Young, K.B. Gettings, B. McCord, P.M. Vallone, Estimating number of contributors in massively parallel sequencing data of STR loci, Forensic Science International: Genetics 38 (2019) 15-22.

[7] C.M. Grgicak, S. Karkar, X. Yearwood-Garcia, L.E. Alfonse, K.R. Duffy, D.S. Lun, A large-scale validation of NOCIt's a posteriori probability of the number of contributors and its integration into forensic interpretation pipelines, Forensic Science International: Genetics 47 (2020).

[8] H. Swaminathan, C.M. Grgicak, M. Medard, D.S. Lun, NOCIt: A computational method to infer the number of contributors to DNA samples analyzed by STR genotyping, Forensic Science International: Genetics 16 (2015) 172-180.
[9] C. Benschop, A. Backx, T. Sijen, Automated estimation of the number of contributors in autosomal STR profiles, Forensic Science International: Genetics Supplement Series 7 (2019).
[10] M.D. Coble, J.-A. Bright, Probabilistic genotyping software: An overview, Forensic Science International: Genetics 38 (2019) 219-224.
[11] D. Taylor, J.-A. Bright, J. Buckleton, Interpreting forensic DNA profiling evidence without specifying the number of contributors, Forensic Science International: Genetics 13 (2014) 269-280.
[12] Ø. Bleka, G. Storvik, P. Gill, EuroForMix: An open source software based on a continuous model to evaluate STR DNA profiles from a mixture of contributors with artefacts, Forensic Science International: Genetics 21 (2016) 35-44.
[13] C.C.G. Benschop, J. Hoogenboom, F. Bargeman, P. Hovers, M. Slagter, J. van der Linden, R. Parag, D. Kruise, K. Drobnic, G. Klucevsek, W. Parson, B. Berger, F.X. Laurent, M. Faivre, A. Ulus, P. Schneider, M. Bogus, A.L.J. Kneppers, T. Sijen, Multi-laboratory validation of DNAxs including the statistical library DNAStatistX, Forensic Science International: Genetics 49 (2020) 102390.
[14] C.C.G. Benschop, A. Nijveld, F.E. Duijs, T. Sijen, An assessment of the performance of the probabilistic genotyping software EuroForMix: Trends in likelihood ratios and analysis of Type I & II errors, Forensic Science International: Genetics 42 (2019) 31-38.
[15] T. Bille, S. Weitz, J.S. Buckleton, J.-A. Bright, Interpreting a major component from a mixed DNA profile with an unknown number of minor contributors, Forensic Science International: Genetics 40 (2019) 150-159.
[16] J.S. Buckleton, J.-A. Bright, K. Cheng, H. Kelly, D.A. Taylor, The effect of varying the number of contributors in the prosecution and alternate propositions, Forensic Science International: Genetics 38 (2019) 225-231.
[17] T.M. Clayton, J.P. Whitaker, R. Sparkes, P. Gill, Analysis and interpretation of mixed forensic stains using DNA STR profiling, Forensic Science International 91(1) (1998) 55-70.
[18] C.C.G. Benschop, J. van der Linden, J. Hoogenboom, R. Ypma, H. Haned, Automated estimation of the number of contributors in autosomal short tandem repeat profiles using a machine learning approach, Forensic Science International: Genetics 43 (2019) 102150.
[19] M.A. Marciano, J.D. Adelman, Developmental validation of PACE™: Automated artifact identification and contributor estimation for use with GlobalFiler™ and PowerPlex® fusion 6c generated data, Forensic Science International: Genetics 43 (2019).
[20] M. Kruijver, H. Kelly, K. Cheng, M.-H. Lin, J. Morawitz, L. Russell, J. Buckleton, J.-A. Bright, Estimating the number of contributors to a DNA profile using decision trees, Forensic Science International: Genetics 50 (2021) 102407.
[21] B. Mittelstadt, C. Russell, S. Wachter, Explaining Explanations in AI, 2018.
[22] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1-38.
[23] D.V. Carvalho, E.M. Pereira, J.S. Cardoso, Machine learning interpretability: A survey on methods and metrics, Electronics (Switzerland) 8(8) (2019).
[24] A. Barredo Arrieta, N. Diaz-Rodriguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion 58 (2020) 82-115.
[25] Z.C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery, Queue 16(3) (2018).
[26] L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining Explanations: An Overview of Interpretability of Machine Learning, 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), 2018, pp. 80-89.
[27] A. Adadi, M. Berrada, Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE Access 6 (2018) 52138-52160.

[28] M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, Communications of the ACM 63(1) (2020) 68-77.
[29] M. Kruijver, H. Kelly, K. Cheng, M.-H. Lin, J. Morawitz, L. Russell, J. Buckleton, J.-A. Bright, Estimating the number of contributors to a DNA profile using decision trees, Forensic Science International: Genetics.
[30] E. Commision, Fostering a European approach to Artificial Intelligence, 2021.
[31] W.J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, Proceedings of the National Academy of Sciences of the United States of America 116(44) (2019) 22071-22080.
[32] R.R. Fernández, I. Martín de Diego, V. Aceña, A. Fernández-Isabel, J.M. Moguerza, Random forest explainability using counterfactual sets, Information Fusion 63 (2020) 196-207.
[33] S. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, 2017.
[34] S.M. Lundberg, B. Nair, M.S. Vavilala, M. Horibe, M.J. Eisses, T. Adams, D.E. Liston, D.K. Low, S.F. Newman, J. Kim, S.I. Lee, Explainable machine-learning predictions for the prevention of hypoxaemia during surgery, Nat Biomed Eng 2(10) (2018) 749-760.
[35] T.K. Yoo, I.H. Ryu, H. Choi, J.K. Kim, I.S. Lee, J.S. Kim, G. Lee, T.H. Rim, Explainable Machine Learning Approach as a Tool to Understand Factors Used to Select the Refractive Surgery Technique on the Expert Level, Transl Vis Sci Technol 9(2) (2020) 8.
[36] Y. Ramon, D. Martens, F. Provost, T. Evgeniou, A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C, Advances in Data Analysis and Classification 14(4) (2020) 801-819.
[37] S. Dandl, C. Molnar, M. Binder, B. Bischl, Multi-Objective Counterfactual Explanations, in: T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, H. Trautmann (Eds.) Parallel Problem Solving from Nature – PPSN XVI, Springer International Publishing, Cham, 2020, pp. 448-469.
[38] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR, Harvard journal of law & technology 31 (2018) 841-887.
[39] A.-H. Karimi, G. Barthe, B. Balle, I. Valera, Model-agnostic counterfactual explanations for consequential decisions, International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 895-905.
[40] R.K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, 2020, pp. 607-617.
[41] S. Sharma, J. Henderson, J. Ghosh, CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models, 2020, pp. 166-172.
[42] M. Schleich, Z. Geng, Y. Zhang, D. Suciu, GeCo: Quality Counterfactual Explanations in Real Time, 2021.
[43] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. Bie, P. Flach, FACE: Feasible and Actionable Counterfactual Explanations, 2020.
[44] J. Moore, N. Hammerla, C. Watkins, Explaining deep learning models with constrained adversarial examples, 2019, pp. 43-56.
[45] M. Keane, B. Smyth, Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI), 2020.
[46] R.M. Grath, L. Costabello, C.L. Van, P. Sweeney, F. Kamiab, Z. Shen, F. Lécué, Interpretable Credit Application Predictions With Counterfactual Explanations, ArXiv abs/1811.05245 (2018).
[47] A. White, A. Garcez, Measurable Counterfactual Local Explanations for Any Classifier, ECAI, 2020.
[48] C. Russell, Efficient search for diverse coherent explanations, 2019, pp. 20-28.
[49] S. Rathi, Generating Counterfactual and Contrastive Explanations using SHAP, 2019.
[50] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and Counterfactual Explanations for Black Box Decision Making, IEEE Intelligent Systems 34(6) (2019) 14-23.
[51] O. Gomez, S. Holter, J. Yuan, E. Bertini, ViCE, 2020, pp. 531-535.
[52] K. Sokol, P. Flach, Desiderata for interpretability: Explaining decision tree predictions with counterfactuals, 2019, pp. 10035-10036.

[53] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, J. Wilson, The What-If Tool: Interactive Probing of Machine Learning Models, IEEE Transactions on Visualization and Computer Graphics 26(1) (2020) 56-65.
[54] S. Barocas, A.D. Selbst, M. Raghavan, The hidden assumptions behind counterfactual explanations and principal reasons, 2020, pp. 80-89.
[55] L. Bertossi, Score-Based Explanations in Data Management and Machine Learning, 2020, pp. 17-31.
[56] A. Adhikari, D.M.J. Tax, R. Satta, M. Faeth, LEAFAGE: Example-based and Feature importance-based Explanations for Black-box ML models, IEEE International Conference on Fuzzy Systems, 2019.
[57] K. Sokol, P. Flach, Conversational Explanations of Machine Learning Predictions Through Class-contrastive Counterfactual Statements, 2018, pp. 5785-5786.
[58] K. Sokol, P. Flach, One Explanation Does Not Fit All: The Promise of Interactive Explanations for Machine Learning Transparency, KI - Kunstliche Intelligenz 34(2) (2020) 235-250.
[59] S. Verma, J.P. Dickerson, K. Hines, Counterfactual Explanations for Machine Learning: A Review, ArXiv abs/2010.10596 (2020).
[60] C.C.G. Benschop, J. Hoogenboom, P. Hovers, M. Slagter, D. Kruise, R. Parag, K. Steensma, K. Slooten, J.H.A. Nagel, P. Dieltjes, V. van Marion, H. van Paassen, J. de Jong, C. Creeten, T. Sijen, A.L.J. Kneppers, DNAxs/DNAStatistX: Development and validation of a software suite for the data management and probabilistic interpretation of DNA profiles, Forensic Sci Int Genet 42 (2019) 81-89.
[61] A.A. Westen, T. Kraaijenbrink, E.A. Robles de Medina, J. Harteveld, P. Willemse, S.B. Zuniga, K.J. van der Gaag, N.E.C. Weiler, J. Warnaar, M. Kayser, T. Sijen, P. de Knijff, Comparing six commercial autosomal STR kits in a large Dutch population sample, Forensic Sci Int Genet 10 (2014) 55-63.
[62] A.R. Akula, S. Todorovic, J.Y. Chai, S. Zhu, Natural Language Interaction with Explainable AI Models, CVPR Workshops, 2019.
[63] C. Molnar, G. Konig, J. Herbinger, T. Freiesleben, S. Dandl, C.A. Scholbeck, G. Casalicchio, M. Grosse-Wentrup, B. Bischl, Pitfalls to Avoid when Interpreting Machine Learning Models, ArXiv abs/2007.04131 (2020).
[64] G. Chiandussi, M. Codegone, S. Ferrero, F.E. Varesio, Comparison of multi-objective optimization methodologies for engineering applications, Computers & Mathematics with Applications 63(5) (2012) 912-942.
[65] N. Gunantara, A review of multi-objective optimization: Methods and its applications, Cogent Engineering 5(1) (2018) 1502242.
[66] K. Sokol, P. Flach, Explainability fact sheets: A framework for systematic assessment of explainable approaches, FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 56-67.
[67] K. Sokol, P. Flach, Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety, 2019.
[68] s.-l. developers, sklearn.preprocessing.QuantileTransformer. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html>, 2020 (accessed 25-05-2021.).
[69] P. Kovesi, Good Colour Maps: How to Design Them, ArXiv abs/1509.03700 (2015).
[70] S.M. Lundberg, B. Nair, M.S. Vavilala, M. Horibe, M.J. Eisses, T. Adams, D.E. Liston, D.K.-W. Low, S.-F. Newman, J. Kim, S.-I. Lee, Explainable machine-learning predictions for the prevention of hypoxaemia during surgery, Nature Biomedical Engineering 2(10) (2018) 749-760.
[71] M.G. KENDALL, A NEW MEASURE OF RANK CORRELATION, Biometrika 30(1-2) (1938) 81-93.

Supplementary Material 1: data analysis and sampling

The original features and their descriptions can be found in Supplementary Table 1 [18]. The feature names were edited to be more descriptive and consistent as they are presented to users.

Original feature name	New feature name	Description
MAC	MAC	Maximum Allele Count
TAC	TAC	Total Allele Count
MinNOC_CSF1PO	CSF1PO min. NOC	Minimal NOC based on locus CSF1PO (allele count at locus CSF1PO / 2, rounded up)
MinNOC_D16S539	D16S539 min. NOC	Minimal NOC based on locus D16S539 (allele count at locus D16S539 / 2, rounded up)
PercAF_D1S1656	D1S1656 perc. known alleles	Number of alleles at locus D1S1656 as a percentage of all known alleles at D1S1656 in the allele frequency file
AlleleCount_D3S1358	D3S1358 allele count	Allele count at locus D3S135
AlleleCount_D8S1179	D8S1179 allele count	Allele count at locus D8S1179
MinNOC_Penta D	Penta D min. NOC	Minimal NOC based on locus Penta D (allele count at locus Penta D / 2, rounded up)
MinNOC_Penta E	Penta E min. NOC	Minimal NOC based on locus Penta E (allele count at locus Penta E / 2, rounded up)
SumAF_TH01	TH01 sum of allele freq.	Sum of frequencies of the alleles at TH01 defined in the allele frequency file
AlleleCount_TPOX	TPOX allele count	Allele count at locus TPOX
MinNOC_TPOX	TPOX min. NOC	Minimal NOC based on locus TPOX (allele count at locus TPOX / 2, rounded up)
stdHeight_vWA	vWA peak height variation	Standard deviation of peak heights at locus vWA(average variation from the mean peak height at locus vWa)
stdAllele	Allele count variation	Standard deviation of the number of alleles per locus (average variation from the mean number of alleles per locus)
MAC0	Loci with 0 alleles	Number of loci with 0 alleles
MAC5-6	Loci with 5-6 alleles	Number of loci with 5 or 6 alleles
peaksBelowRFU	Peaks below 800 RFU	Number of peaks below the stochastic threshold of 800 RFU
MatchProbability	Random match proba.	Probability of a random Dutch person matching to this DNA profile
MinNOC	Min. NOC	Minimal NOC (locus with lowest allele count / 2, rounded up)

Supplementary Table 1: Overview of the 19 features and their descriptions from the original RFC19 model [18]. New feature names were created to be more consistent with their definitions and each other.

Additional data was sampled to handle the sparsity of the original dataset. Supplementary Table 2 details the parameters used to generate 5000 mixture profiles using a development

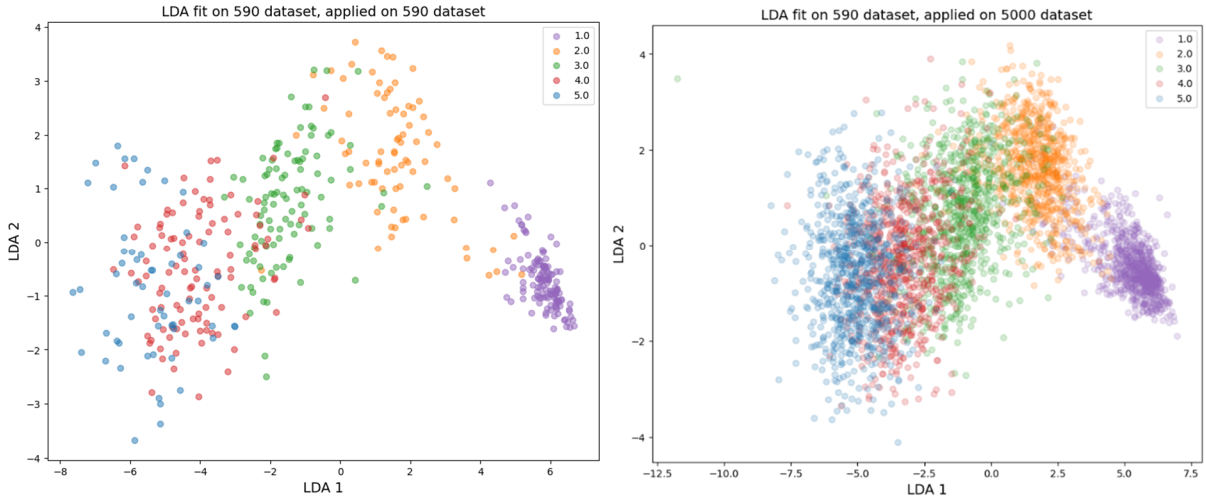
version of DNAStatistX. For each number of contributors (1-5), 1000 profiles were generated. The used STR kit is PowerPlex Fusion 6C™ (PPF6C, Promega) with dye-specific detection thresholds as used by default at the NFI. Dutch population frequency data was used [61]. After generating a profile, LRs were calculated using each donor in a mixture as the person of interest under H1. Only mixtures for which all donors reached a minimum LR of 1000 were included in the dataset. LR calculations were performed using the true NOC under the propositions, using theta correction of 0.01 and using the kit settings for PPF6C as implemented in DNAStatistX.

Parameter	**Value**
Drop-in prC	0.05
Drop-in lambda	0.01
Average peak heights	$U(100, 20000)$
Variation coefficient peak heights	$U(0.1, 1.0)$
Degradation	$U(0.4, 1.1)$

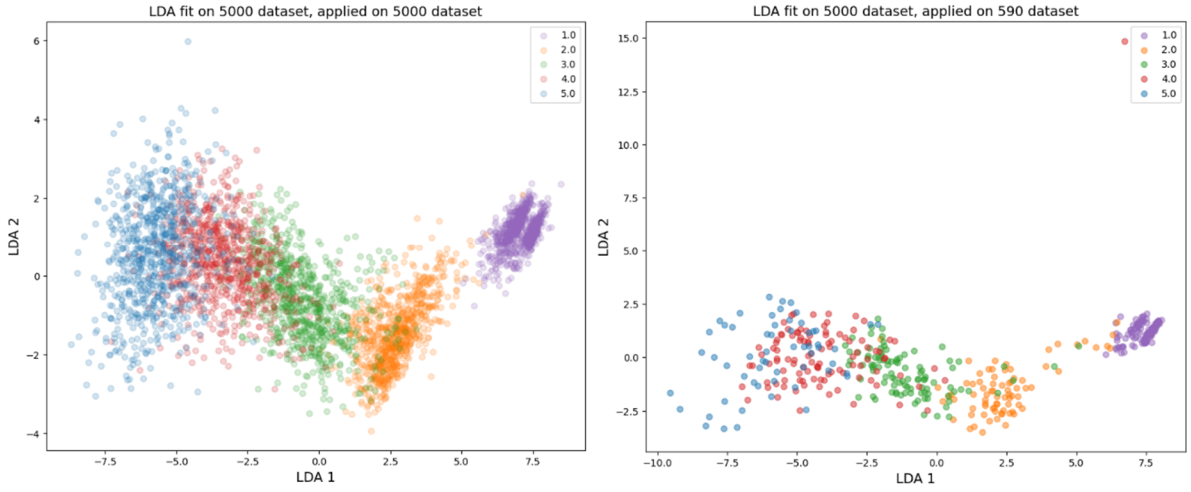
Supplementary Table 2: Sampling parameters as used in the development version of DNAStatistX for simulation of the 5000 mixture profiles.

The following section shows a comparative data analysis of the original dataset and the sampled dataset.

By fitting LDA on the original dataset of 590 samples (from here on referred to as “590-dataset”) and applying it to the dataset of the 5000 sampled instances (from here on referred to as “5000-dataset”) as shown in Supplementary Figure 1, it appears that the 590-dataset captures a lot of the variance that is also present in the 5000-dataset. The spread of the 5000 samples is broad over the two LDA dimensions. However, looking at Supplementary Figure 2, it seems that by fitting LDA on the 5000-dataset, only LDA 1 captures a good spread of the 590-dataset. LDA 2 does not contain much differential information for the 590-dataset. This is to be expected as the 5000-dataset is artificially created and therefore might contain less unexpected variation which is present in the 590-dataset. The one outlier is a 4-person mixture that has a high TAC of 138 which is the highest TAC value in the dataset. In the LDA for the 590-dataset it is in the middle of the red cluster.

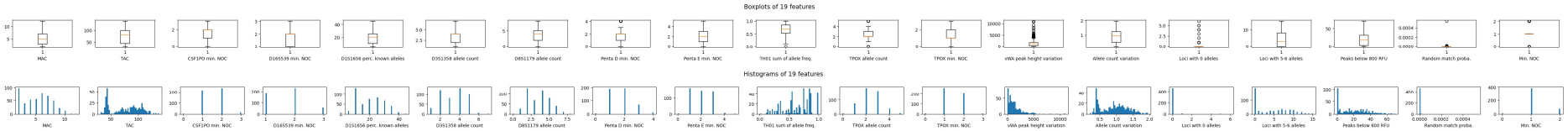


Supplementary Figure 1: LDA fit on the 590-dataset, applied to both that same dataset and to the 5000-dataset. DNA profiles consisting of one, two, three, four, or five donors are presented as purple, orange, green, red and blue circles, respectively.

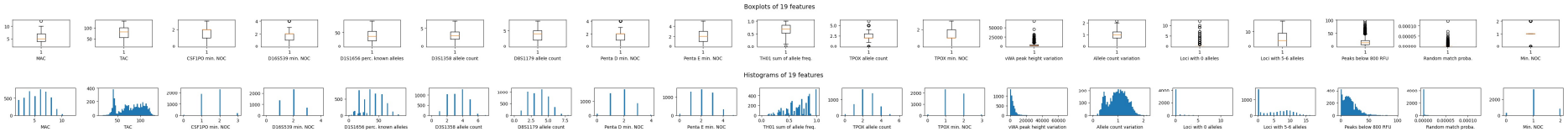


Supplementary Figure 2: LDA fit on the 5000-dataset, applied to both that same dataset and to the 590-dataset. DNA profiles consisting of one, two, three, four, or five donors are presented as purple, orange, green, red and blue circles, respectively.

Plotting the 19 features in boxplots and histograms show that they are mainly not normally distributed and contain many outliers. This is true for both datasets as can be seen in Supplementary Figures 3 and 4.



Supplementary Figure 3: Boxplots and histograms for each of the 19 features in the training set of the 590-dataset.



Supplementary Figure 4: Boxplots and histograms for each of the 19 features in the training set of the 5000-dataset.

Ideally all 19 features should follow comparable distributions for both datasets. From visual inspection, it is clear that this is not the case for all these features. By using the two-sided Kolomogorov_Smirnov (KS) statistic[1], it was determined that 8 features do not appear to be drawn from the same distribution. The results of this statistic are listed in Supplementary Table 3, where a large KS statistic or small p-value (less than 1.0e-2) corresponds to rejecting the null hypothesis assuming the samples were drawn from the same distribution. For discrete variables, a different KS statistic implementation was used[2].

	KS statistic	p-value
MAC	**0.12**	**3.0e-7**
TAC	**0.09**	**1.2e-4**
CSF1PO min. NOC	0.04	3.3e-1
D16S539 min. NOC	**0.09**	**3.3e-4**
D1S1656 perc. known alleles	**0.63**	**4.7e-200**
D3S1358 allele count	0.06	4.8e-2
D8S1179 allele count	0.06	2.5e-2
Penta D min. NOC	0.04	3.4e-1
Penta E min. NOC	0.06	3.3e-2
TH01 sum of allele freq.	0.07	1.9e-2
TPOX allele count	0.05	9.6e-2
TPOX min. NOC	0.02	9.9e-1
vWA peak height variation	**0.42**	**5.7e-83**
Allele count variation	**0.13**	**7.2e-8**
Loci with 0 alleles	0.05	1.6e-1
Loci with 5-6 alleles	0.05	9.0e-2
Peaks below 800 RFU	**0.19**	**5.7e-18**
Random match proba.	**0.10**	**9.4e-5**
Min. NOC	0.05	1.6e-1

Supplementary Table 3: KS statistic results for the 19 features comparing the 590- and 5000-datasets. Bold features have a large statistic value or small p-value, and therefore the null hypothesis is rejected. This means that these features appear to be drawn from different distributions.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks_2samp.html
[2] https://rdrr.io/cran/dgof/man/ks.test.html

For analyzing feature correlations, we applied Kendall rank correlation coefficient. It is suitable for features that are not normally distributed (as is assumed for Pearson), and is more robust to outliers [63, 71]. Most features are highly correlated as can be seen in Supplementary Figure 5. A slight decrease in correlation can be observed between the 590- and 5000-dataset, though most values still lie above 0.4.

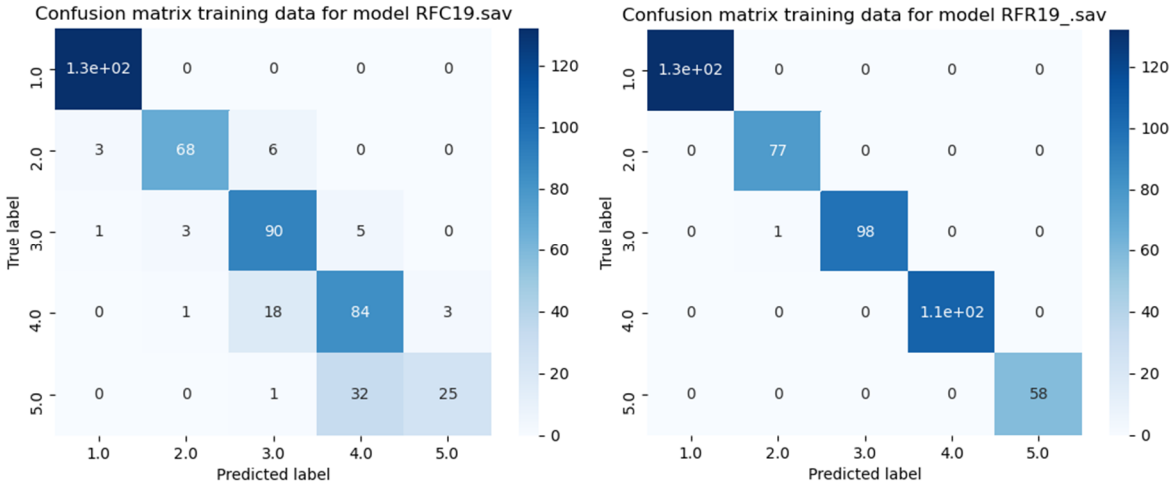
Supplementary Figure 5: Kendall feature correlations of the 19 features in the training set of the 590-dataset (left) and 5000-dataset (right).

In a short benchmarking study, we explored if A) regression can possibly outperform classification for NOC estimation and B) training on the new merged 5590-dataset has benefits for performance for NOC estimation.

Supplementary Table 4 shows which models were used to compare regression and classification, while Supplementary Figures 6 and 7 list the results on the training- and test data respectively.

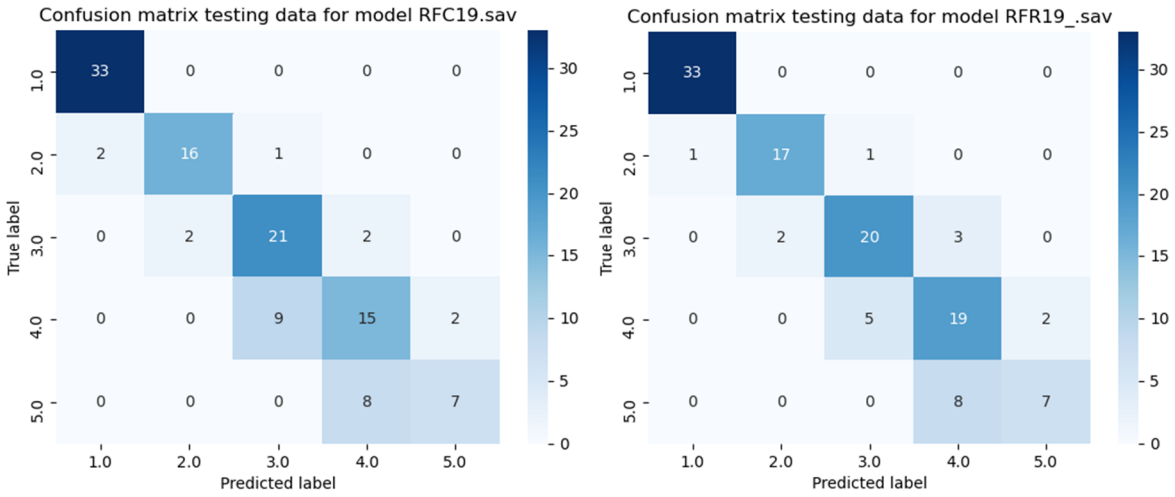
	RFC19 model	**RFR19 model**
Model type	Random forest classifier[3]	Random forest regressor[4]
Model parameters	As described in [18]	Defaults
Dataset used	590-dataset	590-dataset

Supplementary Table 4: Models and model parameters used for a short benchmarking to comparing the original model with a default regressor, on the original 590-dataset.



Supplementary Figure 6: Confusion matrices of the RFC19 (left) and RFR19 (right) models applied on the training data from the 590-dataset. The regression model performs almost perfectly (99% accuracy), where the classifier performs a bit worse (85%), and predicts some instances more than one class off.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.htmls



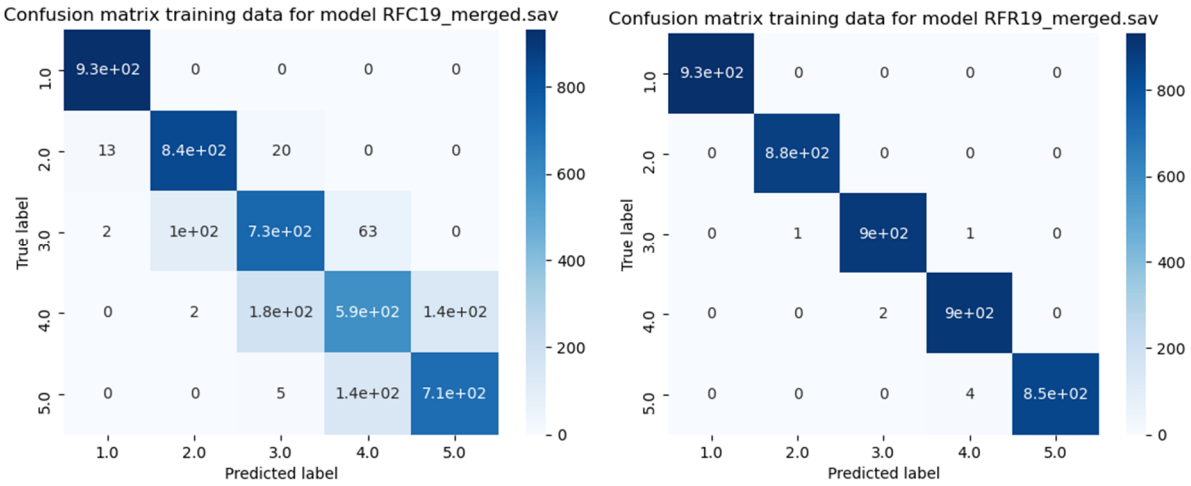
Supplementary Figure 7: Confusion matrices of the RFC19 (left) and RFR19 (right) models applied on the test data from the 590-dataset. The regressor obtains similar or better results than the classifier (81% accuracy versus 78%). Especially for profiles of 4 donors, there is consistent better performance from the regression model. For profiles of 2, 3, and 5 donors, the number of correct predictions is on average the same for the classifier and the regressor.

The larger discrepancy between the train and test performance for the regressor is due to overfitting. The default parameters of the regressor in comparison to the optimized classification parameters are more tuned towards larger datasets. These results show promise that a regression model could outperform a classification model once put through more rigorous training.

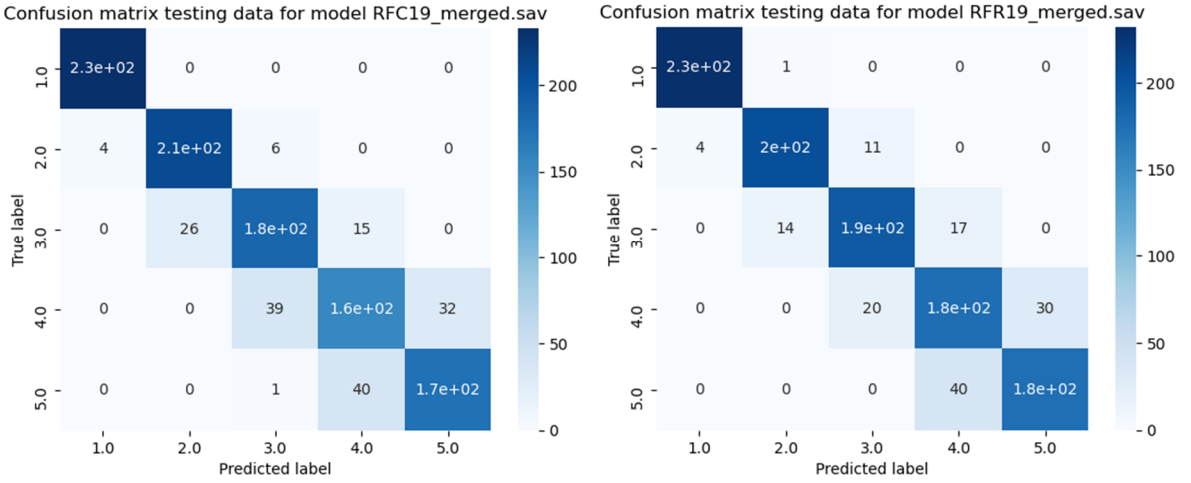
Supplementary Table 5 shows which models and datasets were used to test the merged 5590-dataset, while Supplementary Figures 8 and 9 list the results on the training- and test data respectively.

	RFC19_merged model	**RFR19_merged model**
Model type	Random forest classifier	Random forest regressor
Model parameters	As described in [18]	Defaults
Dataset used	5590-dataset (merged 590- and 5000-datasets)	5590-dataset (merged 590- and 5000-datasets)

Supplementary Table 5: Models and model parameters used for a short benchmarking study to explore if the new merged 5590-dataset has benefits for performance on the original 590-dataset.



Supplementary Figure 8: Confusion matrices of the RFC19_merged (left) and RFR19_merged (right) models on the training data from the 5590-dataset.



Supplementary Figure 9: Confusion matrices of the RFC19_merged (left) and RFR19_merged (right) models on the test data from the 5590-dataset. Accuracy now lies at about 85% for classification, and 88% for regression. Improvement lies mainly for predicting 3 and 4 contributors.

Additional analysis summarized in Supplementary Table 6 shows that the regression model trained on the 5590-dataset performs about equally well on samples originating from the 590- and 5000-dataset, while the classification model performs slightly worse on samples from the 590-dataset. The overall performance of the regression model is also slightly better, showing its potential for future applications.

	RFC19_merged	**RFR19_merged**
Total test accuracy	85%	88%
Test accuracy on samples from 590-dataset	82%	86%
Test accuracy on samples from 5000-dataset	86%	88%

Supplementary Table 6: Performance of the RFC19_merged and RFR19_merged models on the test data from the 5590-dataset. We compare the accuracy on samples that originate from the original 590-dataset and the 5000-dataset separately.

Contents additional chapters

1. Proposal and methodology

1.1 Problem description

One of the steps in DNA profile interpretation is determining the Number Of Contributors (NOC) to a DNA sample. The Netherlands Forensics Institute (NFI) previously developed a machine learning model to predict the NOC of a DNA sample based on features derived from Short Tandem Repeat (STR) data [1]. This random forest classifier uses 19 statistical features derived from the STR data which outperforms more conventional methods for NOC estimation. However, the only output that DNA experts can consult is the range of probabilities for each NOC. When the expert analyzes a profile and comes to a different result than the machine learning model outputs, it is challenging to determine what is correct. No information about how the model came to this conclusion is provided, therefore not allowing experts to use this support tool optimally for their decision making. This decision can be of importance in weight of evidence calculations [2].

With the addition of eXplainable Artificial Intelligence (XAI), the NFI aimed to improve the value of their prediction model for experts in determining the NOC. XAI has been recognized as a tool to help humans understand the *why* of outcomes in Machine Learning (ML) applications [3-7]. Many of such methods have been developed to understand the factors that influence certain decisions made by ML applications, which is what the NFI was looking for as well. For instance, if the NOC model predicts a different outcome than the expert had in mind, the expert can consult explanations of the model. In this way, the expert can make an informed decision to stick with their own conclusion if the model does not seem to have learned the correct distinctions, or choose the predicted value if the model makes a good case.

1.2 Main related works

Explanations have a certain scope; they can be applied to a single prediction, or to the entire ML model. This distinction is defined as local- or global explanations [3-7]. As experts are evaluating the individual predictions of a ML model, they are concerned with local explanations. Besides scope, explanation techniques can either be optimized for certain ML models, or be developed to work for any type of ML model. These are called model-specific or model-agnostic respectively [3-7]. Since the NFI has plans to keep optimizing the ML model for determining the NOC, we focused on model-agnostic methods. There exist roughly two directions of generating local, model-agnostic explanations.

The first is techniques such as SHAP, which has been established as providing effective explanations in the form of the top input features that have driven the model to making a certain prediction [7]. This effectively answers the question *"Why did the model predict A?".* Some research has implemented SHAP to real-life cases such as predicting hypoxia based on clinical data [8], and predicting the most fitting eye-surgery type [9]. They seem to have obtained valuable information about which factors these ML models used to make decisions.

The second direction of explanations is a more recent research direction, which answers the question *"Why did the model not predict B?"*. This type of explanation is called a counterfactual, showing how the instance could have been predicted differently if certain input features were different [8, 9]. This way of reasoning is underpinned by the social sciences to be effective, as humans seek contrastive explanations [10]. Since this technique is new, numerous methods are being developed, yet none has particularly risen to the top as with SHAP [11].

The literature on generating explanations underpins the value of creating explanations that are catered towards a specific problem, as their effectiveness is highly sensitive to the audience they are presented

to [10]. Therefore, it was a good idea to explore which techniques exist, if they could be applied to this problem, and how they could be adapted to produce the best result.

1.3 Research questions

In this study, we aimed to generate local, model-agnostic explanations for ML models that predict the number of contributors. To achieve this, we had to identify the existing techniques for generating local explanations and the types of assumptions they make on the underlying data. In this way, we could decide which methods were applicable to the specific dataset that we have available.

How can we generate informative model-agnostic local explanations for predictions of the number of contributors (NOC)?

1. What information do experts look at when determining the NOC?
2. What purpose does an explanation of the NOC machine learning model serve?
3. What does the NOC machine learning problem look like?
4. Which types of local explanations could work for this problem?
5. How can an explanation be presented to the users?
6. How can local explanation techniques be adapted to be applied to this problem?
7. How can we evaluate the generated explanations from a machine learning perspective?
8. How can we evaluate the generated explanations from a user perspective?

1.4 Methodology and planning

Originally, the thesis was planned in a linear fashion where the phases of development would be processed one by one. After considerations about maintaining quality, it was decided that an Agile approach would be more beneficial. In this way, quality could be monitored by using short cycles which include all phases of development in a period of three weeks each. The cycles could focus more on certain stages of the process as time progresses. For instance, cycles were more research-heavy at the earlier stages, and more evaluation-heavy at the end. At the end of each period, reflection helped adjust the course of action. Figure 1 shows the planning in the form of a Gantt chart. Table 1 contains an overview of which research questions were answered in which cycles, and what sections of the thesis contain the corresponding answers.

The first cycle was designed to establish a baseline from both a user perspective as from a technological view. From the users we surveyed their usual workflow; how they use the machine learning model; and what is missing. Similarly, we applied some popular explanation techniques to the data as a baseline for the explanations. The data and machine learning model were analyzed. With this sprint, questions 1, 2, and 3 were answered, and a start with question 4 was made.

Cycle 2 was mostly concerned with exploration of several techniques such as Anchors, SHAP, counterfactuals and how these could be combined to further answer question 4. To answer question 5, the visualization was also mainly developed during this time as it was clear that the profile feature values, in combination with a counterfactual would need to be presented to the user.

From previous cycles, it was clear that a combination of SHAP values and a counterfactual could work well. As no counterfactual method was currently fully suitable for this problem, we worked on creating our own implementation, working towards the answer of question 6. This was also when extra data was sampled.

In cycle 4, the counterfactual method was finalized as well as the total visualization, answering questions 5 and 6. The objective evaluation functions were implemented during this time as well for question 7.

The final sprint consisted of running the objective evaluations against the state of the art, while also creating a user study to answer question 8.

	Cycle 1: **baseline**	Cycle 2: **exploration**	Cycle 3: **implementation**	Cycle 4: **integration**	Cycle 5: **evaluation**
Question 1	P 1.1, Ch 2, 3				
Question 2	P 2.3, 2.4				
Question 3	P 2.1, 2.2				
Question 4	P 1.2, 1.3, Ch 3	P 1.3, Ch 4			
Question 5		P 2.8	P 2.8	P 2.8, 3.2	
Question 6			P 2.5	P 2.5, 2.6	
Question 7				P 2.7	P 2.7, 3.1
Question 8					P 2.9, 3.3, Ch 5

Table 1: Overview of which research questions were answered in which cycles and in which sections of the thesis these can be found.

1.4.1 Risks

The main risks of this project related to the techniques for explanations and user study. Some mitigation steps were defined as follows:

1. The explanation techniques are difficult to implement, slowing down the progress.
    a) Implement any techniques with available code first.
    b) Ask for help from NFI supervisor / colleagues.
2. No users are available to participate in the user study.
    a) Ask feedback from Corina Benschop to get at least one expert evaluation.
    b) Create a substitute task and ask for feedback from colleagues at the NFI.

Phases		9-11-2020	16-11-2020	23-11-2020	30-11-2020	7-12-2020	14-12-2020	21-12-2020	28-12-2020	4-1-2021	11-1-2021	18-1-2021	25-1-2021	1-2-2021	8-2-2021	15-2-2021	22-2-2021
1. Identify problem	Q2							**Christmas**	**holiday**					**Spring**			
2. Define research questions	Q3													**holiday**			
3. Run cycles: research - design -	Identify problem																
implementation - experiment - evaluation	Create baseline																
4. Evaluate final product with users	Define research questions																
5. Write thesis	Run cycles									cycle 1 (demo: 21-1)			cycle 2 (demo: 11-2)			cycle 3 (demo: 4-3)	
	Update meeting supervisors						**16-12**							**01-02**			
	Send committee form											**22-1**					

Cycle 1: Baseline		1-3-2021	8-3-2021	15-3-2021	22-3-2021	29-3-2021	5-4-2021	12-4-2021	19-4-2021	26-4-2021	3-5-2021	10-5-2021	17-5-2021	24-5-2021	31-5-2021	7-6-2021	14-6-2021
Cycle 2: Exploration	Q3					**2-4 to 5-4 Easter**				**27-4**	**5-5**	**13-5 - 14-5**					
Cycle 3: Implementation	Q4																
Cycle 4: Integration	Run cycles		cycle 4 (demo: 25-3)			cycle 5 (demo: 15-4)											
Cycle 5: Evaluation	Evaluate final product																
	Write thesis																
	Prepare defense																
	Update meeting supervisors	**03-03**					**07-04**						**17-05**				
	Midterm meeting	**03-03**															
	Send green light form + draft									**30-04**							
	Final submission														**03-06**		
	Final defense																**17-06**

Figure 1: Gantt-chart showing the general planning of the thesis. The phases on the left show the general phases of the project; starting from identifying the problem and the research questions, then moving into cycles of development. The themes of the 5 cycles are listed below, these were adjusted during the process. Some extra time was allocated for the final evaluation of the product with users and finally writing the thesis. All phases and cycles are marked in purple and also listed inside the left-most column of the chart. Milestones and meetings are shown in blue, and holidays in red.

2. DNA Mixture interpretation

Experts can use DNA evidence to determine if certain people were involved in a crime by comparing the suspect DNA, victim DNA and other DNA samples to the evidence found at a crime scene. This interpretation becomes more difficult when the DNA profile consists of evidence from multiple people since information might overlap, or not every person contributed as much material. Even though software exists for analyzing this evidence, it is required that the expert inputs how many people contributed to the sample [12]. This chapter explains how to interpret a specific type of DNA profile, and gives a quick impression on how the number of contributors can be determined and which factors might influence that process.

2.1 Short Tandem Repeat (STR) profiles

In forensic work, DNA evidence is often analyzed using *Short Tandem Repeat (STR)* profiles. These STRs are specific tracks of repeated short DNA sequences of about two to six base pairs long that have been proven to show high variability between individuals in how many times the sequence repeats [13]. Most of these parts of the DNA or loci have been defined by CODIS (Combined DNA Index System), the United States national DNA database. We can capture the STR within three steps. First, a Polymerase Chain Reaction (PCR) is used to amplify the available DNA and label the target loci. Secondly, with Capillary Electrophoresis (CE), the fragments are separated based on size and label. Third and finally, this output is translated with software to an electropherogram. We will not go into detail about how these methods work specifically, as mainly the results are of interest. In Figure 2, we see a simplified result which the electropherogram can produce for locus TH01. The y-axis shows the amount of information found in Relative Fluorescent Units (RFU), which is how the machine counts the quantity of DNA found. The top of the x-axis shows the number of base pairs, a measurement for the entire sequence found. Most importantly, we see two peaks, representing two alleles on this locus. These alleles are characterized by the number of repeats of the STR for locus TH01, which is [AATG]. On the right of Figure 2, we see the DNA sequence for six and eight repeats.

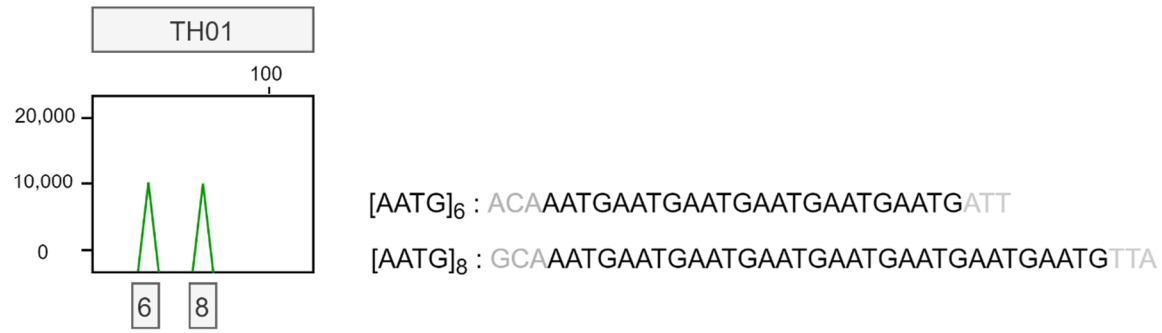


Figure 2: Simplified electropherogram result for locus TH01 showing two alleles with six and eight repeats each. The repeat sequence is shown on the right with arbitrary flanking regions that do not represent reality.

One individual can have two different alleles for a single locus; one inherited from the mother, and one from the father. It is also possible that a person inherited the same allele from both of their parents, this means that they are homozygous at that locus. The peak will then be twice as large. We will now get into more detail of how to derive the number of contributors from an STR profile.

2.2 Estimating the Number of Contributors (NOC)

The first step of DNA STR profile interpretation is to determine whether a sample has originated from a single source, or if the sample is a mixture [14]. This is often easily discerned by checking whether or not there are loci with more than two alleles present. As we saw in Figure 2, a single person can

contribute a maximum of two alleles per locus, so profiles with more alleles are usually considered a mixture. The next step is to determine the NOC. This step is necessary for DNA analysis software to calculate the *weight of the evidence* found [15]. An incorrect NOC can have an effect on this weight of evidence [2]. In extreme cases it could make the difference between support for the proposition that a person of interest (POI) contributed to the evidence or the support for the alternative, that an unknown person, unrelated to the POI, contributed to the evidence.

Determining the exact number of contributors can be challenging. There are several obscuring factors that could make an expert underestimate the number of donors, especially when the number of donors increases [14, 16]. While the two left-most pictures in Figure 3 are quite simple to interpret, the two on the right can be somewhat ambiguous because of the factors described below.

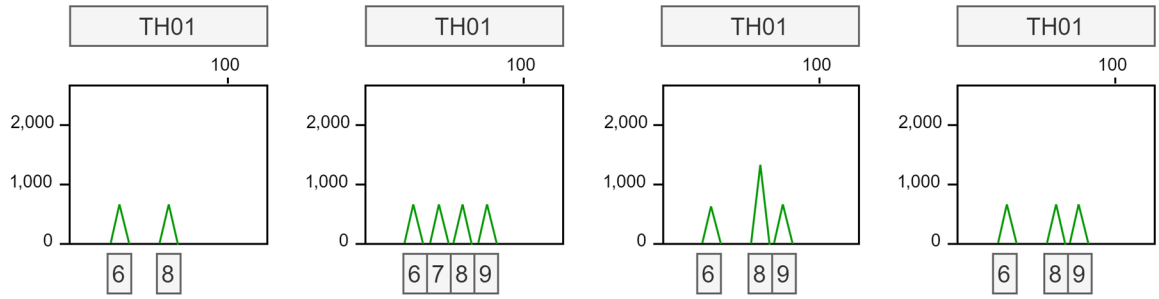


Figure 3: Four simplified electropherogram results for locus TH01. From left to right: Example of a simple single donor profile; Example of a simple 2-person mixture profile; Example of a 2-person mixture profile with allele sharing or a homozygous allele, peak 8 is twice as high compared to peak 6 and 9; Example of a 2-person mixture profile with drop-out, one peak has likely not been detected.

- Allele sharing: If two donors have the same allele at a locus, this is called allele sharing. It frequently occurs when donors are relatives, since siblings share a lot of DNA. It might be difficult to distinguish if an allele is shared between donors, or if a single donor simply is homozygous for this allele; in both cases, the peak height for that allele is higher. This can be seen in the third picture from the left of Figure 3; allele 8 is twice as high as alleles 6 and 9.
- Allele drop-out: If the DNA was degraded, for example due to sunlight, some parts of the DNA might not be present in the sample to measure. It is also possible that the amount of available DNA is so small, that the alleles fall below a certain detection threshold. Because of this low quality or quantity of DNA, some allele fragments might not show up in the profile at all, which is called drop-out. This can be seen in the right-most picture of Figure 3; only 3 alleles are found.

These factors can decrease the number of alleles found in a certain profile, which could lead to an underestimation of the number of contributors. There are also factors that could lead to an overestimation of alleles present in a sample, and thus an overestimation of the NOC [14]. The two right-most images in Figure 4 demonstrate these phenomena which are also described below.

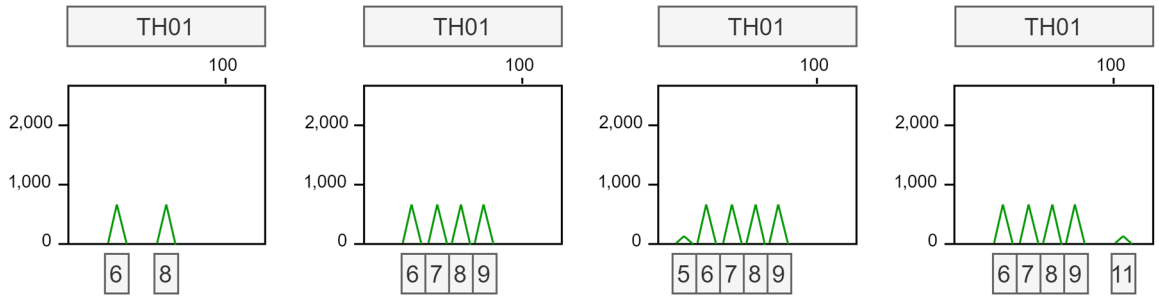


Figure 4: Four simplified electropherogram results for locus TH01. From left to right: Example of a simple single donor profile; Example of a simple 2-person mixture profile; Example of a 2-person mixture profile with a stutter peak at allele 5 caused by the folding of an STR with 6 repeats; Example of a 2-person mixture profile with a noise peak at location 11 caused by an error in reading or low-level contamination event.

- Stutter: During the process of measuring the STRs, a STR fragment can accidentally fold over itself. This could cause the electropherogram to measure this strand to have one repeat fewer, since the folded part of the fragment is not correctly measured. In this way, a small stutter peak is found in the profile just before the valid peak. In Figure 4, this is shown by the small peak at allele 5, caused by the folding of the STR with 6 repeats. Note that also +1, -2, or even complex +/-0.5 stutter peaks can occur.
- Allele drop-in or other noise: The measuring process is not perfect, so some random noise or low-level contamination might show up in the electropherogram, that does not contain any information about the DNA. In Figure 4, we can see that the rightmost image has a small peak at allele 11. Since it is not close to another allele, it is likely not a stutter peak.

Stutter peaks and noise are often filtered out using certain thresholds in the profile analysis software. As a result, some DNA information might also be lost if there is little material available. All of the previously discussed events are more troublesome and prevalent when there is little information available (low template).

The simplest method to get an estimate of the NOC is by using the Maximum Allele Count (MAC)-method [14, 17]. By taking the locus with the most alleles present, dividing that number by two, and rounding up, we can get an idea of the minimum NOC. Though this method is simple, it does not take into account the factors of allele sharing, drop-out, etc. Therefore, the performance is quite poor with 3 or more contributors, when there is a lot of allele sharing, or when the quality of the profile is low [18, 19]. When assessing mixtures between 2-5 contributors, depending on the dataset, the MAC can obtain correct predictions for about 60-70% of samples [1, 19, 20].

Besides the MAC, the Total Allele Count (TAC) can also give an indication for the NOC. The TAC is a count of all the found alleles of all loci, which gives a more general overview of the profile as a whole [14]. A combination of these two measures can give a better impression of the entire profile [14, 20].

3. Survey on mixture interpretation and explanation types

From the background information, we obtained a good grasp of how the NOC can be determined. This survey was then run to verify that the experts at the NFI had a similar workflow, thought process and looked at similar data. There were 12 responses in total.

3.1 Set-up

The survey was structured according to three main questions:

1. What is the normal workflow of experts when estimating the NOC?
2. What type of explanation do experts prefer to help them make a decision (feature importance or counterfactual)?
3. What type of data do experts prefer to help them make a decision (features or peak information)?

Question 1 verifies the workflow of the experts to see if any information was overlooked. Questions 2 and 3 relate to the possible types of explanations that could be implemented. For explaining single predictions in a model-agnostic fashion, there are two main approaches that work well for tabular data; feature importance methods and counterfactual explanations (see Appendix A). To confirm that these are valuable for this specific problem, this survey contrasted these types of explanations. The type of data that is presented to the user is also important. Currently, the machine learning models crafted by the NFI are based on features concerning summary statistics of the profile (such as the TAC and the MAC). However, it would also be possible to train (deep) models on the peak information to create predictions. This information concerns the peak location and size for all loci per profile. With this question, we intended to find any preference regarding the data.

3.2 Question 1: workflow

This question describes an average workflow as interpreted by the literature. The users were asked to write any missing steps in the workflow. In summary:

- Inspect general information about the profile (peak heights, TAC, MAC, NOC tool prediction);
- Check the locus with the MAC to see if all peaks can be explained with the expected number of donors;
- Check for stutter peaks / extra peaks from another donor.

3.2.1 Analysis answers question 1

The following remarks were reported to be missing from the workflow:

- Check the number of peaks below the detection threshold (6x). This gives an indication of the DNA quality (1x) / the amount of dropout (3x).
- Experts can often not make a reliable choice between 4 or 5 donors based on the information (1x).
- Locus SE33 (1x).
- None (3x).

In summary, missing information concerned the number of peaks below the detection threshold which is not available to the machine learning model, so this information could not be incorporated. The remark about 4 or 5 donors demonstrates the difficulty of making decisions with more donors. Locus SE33 has the largest variety of alleles, which is why one user finds it more informative. The remarks suggest that **more and/or different features could be useful for the machine learning model in the future.**

3.3 Question 2: feature importance or counterfactual explanations

This question describes two types of explanations for the same prediction of a profile. Option A are SHAP values, while option B is a counterfactual explanation. The users were asked to choose which explanation would be most helpful to make a decision between two NOC values (4 or 5). They could also pick both options. The question is shown in Figure 5 and the results are presented in Figure 6.

Figure 5: Question 2 with option A showing feature importance values and option B a counterfactual explanation.

3.3.1 Analysis answers question 2

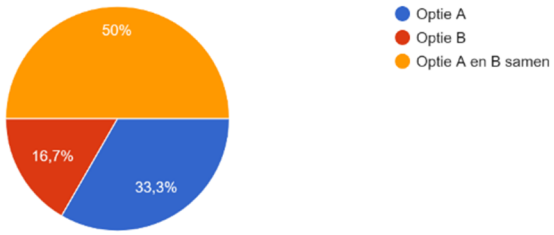


Figure 6: Pie chart of answers to question 2.

Motivation for choosing A:

- You want to know why the model predicted its result (2x).
- Easier to understand (1x).
- Option B is also good, but a visual explanation is better (1x).
- Option B is also good, but can the model know if the expert is interested in 4 or 6? (2x).

Motivation for choosing B:

- Is more specific for comparing one to another (2x), which is relevant for criminal case investigations (1x), option A is more background information.
- Option A and B together is also a good option.

Motivation for choosing A and B:

- Option B can provide very specific information (1x) (e.g., if the allele count on one locus were lower to get a different NOC, it could be explained by stutter).
- Option B is relevant when you came to a different NOC than the tool outputs (2x).

- Option A tells you why it came to its result in the first place (4x).
- Option B tells you where the threshold values lie (1x).
- Option B tells you if the predictions were close together (1x).
- More convincing (1x).
- Combination of information makes the decision complete (1x).

In summary, **most users liked the combination of explanations to form a complete picture**. People that picked one option, often also mentioned they liked the other as well. **They enjoyed the general information of the feature attributions, and the specific values of the counterfactuals.** The counterfactual seemed to provide extra information such as giving an impression of the threshold values and how close the decision is.

Since option A had a visualization, as opposed to option B, it could have induced some presentation bias as seen in one of the responses.

3.4 Question 3: features or peak information

This question describes two counterfactual explanations based on different types of data. Option A consists of the features that are currently used by the machine learning models. These are mainly summary statistics that describe aspects of the profile. Option B shows information about peak heights. The users were asked to choose which explanation would be most helpful to make a decision between two NOC values (4 or 5). The question can be found in Figure 7 and its answers in Figure 8.

A: Hieronder staat hoe dit profiel zou moeten veranderen zodat het model een NOC van 4 zou voorspellen in plaats van 5. Er is te zien wat de feature waardes van de TAC en AC5-6 hadden moeten zijn in vergelijking met het originele profiel om tot een NOC van 4 te komen. Hetzelfde profiel met een TAC-waarde van 100 i.p.v. 115, en een AC5-6-waarde van 8 i.p.v. 13.

TAC: verlaag van 115 naar 100

AC5-6: verlaag van 13 naar 8

B: Hieronder staat hoe dit profiel zou moeten veranderen zodat het model een NOC van 4 zou voorspellen in plaats van 5. Er is te zien wat de 2 piekwaardes van de locus TH01 hadden moeten zijn in vergelijking met het originele profiel om tot een NOC van 4 te komen. Hetzelfde profiel met een piekhoogte op allel 5 van 0 i.p.v. 3500, en een piekhoogte op allel 6 van 0 i.p.v. 1800.

Locus TH01 allel 5 piekhoogte: verlaag van 3500 naar 0

Locus TH01 allel 6 piekhoogte: verlaag van 1800 naar 0

Figure 7: Question 3 with option A showing an explanation based on features and option B showing an explanation using peak height information.

3.4.1 Analysis answers question 3

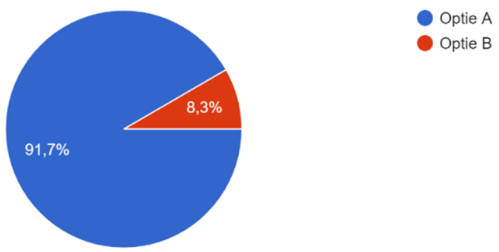


Figure 8: Pie chart of answers for question 3.

Motivations for option A:

- When you change the number of peaks or peak heights, you essentially change the TAC / MAC / other features (1x).
- Changing peak heights / removing peaks at a certain locus does not make sense (2x).
- Option B seems too trivial, it sounds like the prediction is based on only one locus (2x).
- Prefer to look at TAC / MAC (profile-level), not the peak heights at one locus (3x).
- Option B would require a lot of research into individual loci, option A is enough (1x).
- Option A gives more information than the expert can see, whereas option B the expert probably already noticed (2x).
- Peak heights are not stable for the PPF6C kit (1x).

Motivations for option B:

- Option B could be useful to characterize the imbalance between peak heights. They also mention that they would like to see information such as stutter levels, drop-ins, TAC, MAC, and mention that a combination would be ideal.

In summary, **every participant sees the value in profile-wide features**. Most people mentioned that they mainly **consider the profile as a whole, and would not consider peak heights at a single locus as informative**. On top of that, one expert found that the peak heights in the used kit are not stable so making a decision on that information might not be a good idea. Features also give new information, whereas the peak heights are already available to the expert. The one person who picked option B mentioned that there could be value in the ratio between peak heights, while also expressing their interest in the features. It could be interesting to **encode this ratio into a new feature.**

Because option B only adjusted peaks at one locus, a lot of experts expressed that they would never base their decision on a single locus and were therefore a bit puzzled. It would have been better to have presented multiple changes at different loci to mitigate this confusion.

3.5 Conclusion

The workflow matched well with our expectations, there were no unexpected answers. Regarding the explanations, it was interesting to see that the experts found most value in the combination of the two types. Where feature attributions give a general impression of the prediction, the counterfactual provides more specific information showing where the threshold of the prediction lies. It was surprising to see that almost nobody found the peak data informative to base the explanation on. This could be attributed to the fact that we only presented information about a single locus, or because this peak information is already available to the experts when they analyze a profile, therefore not resulting in new insights.

4. Experiments with various XAI techniques

From the user survey and an additional brainstorming session with the DNA experts, we determined that there were two questions that required an answer:

1. *What were the main reasons for the model to reach the current prediction?*
2. *With which feature changes could the model have arrived at a different prediction?*

It seemed that for question 1, a general overview of feature importance was adequate. Then to answer question 2, a specific example was fitting. In this way, there is both general information about the current prediction, as well as a specific example of a close different prediction. For both questions 1 and 2, we wanted a technique that is model-agnostic and provides explanations per prediction (local).

The following sections show several experiments that ultimately led to the requirements and implementation found in the paper. These experiments show how several techniques were considered.

4.1 SHAP

SHAP is one of the most established feature importance methods with a solid theoretical foundation [21]. From question 2 of the user survey, we received positive feedback on the general information that the SHAP values can provide about a prediction, which is why we wanted to incorporate it into our explanation. As we were on the verge of deciding between classification and regression, we wanted to explore what information SHAP values can provide in both settings, and which version the users prefer. SHAP values can be presented in various ways, but we mainly wanted to focus on the information that it conveys. Therefore, we used one of the simplest visualizations that the package offers, *force plots*.

4.1.1 SHAP for multi-class classification

Figure 9 shows an example of the explanation for a 4-person mixture profile, which is predicted as such by the original classification model with a probability of 0.67. Note that this figure only shows the explanation for class '4', with red bars representing feature values that make this prediction more certain, and blue bars representing feature values that make this prediction less certain. The force plot is two-dimensional, so the red and blue bars can only represent two directions such as "more certain" and "less certain". If the expert were to explore the full range of the prediction, that means they would need to look at five separate force plots.

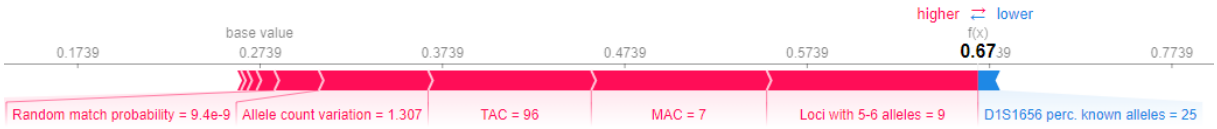


Figure 9: SHAP force plot for classification.

The experts liked that this explanation gives them access to the probability of the prediction, and they enjoyed the overall visualization. However, they would not want to look at multiple figures because that takes too much time and effort to understand and compare. They also thought it was confusing that the same feature values can contribute positively to multiple classes. In essence, they do not want to extrapolate the relevant information themselves.

4.1.2 SHAP for regression

By using SHAP force plots in the context of a regression model, we obtain a more concise report. For example, in Figure 10 the same profile as in Figure 9 is predicted to have an NOC of 3.96. The red bars

now represent the feature values that push the prediction “up”. If there were blue bars, these would push the prediction “down”. The two-dimensional scale can now represent the entire range of the outputs from 1-5, meaning that only one figure is needed to show all possible values of the output.

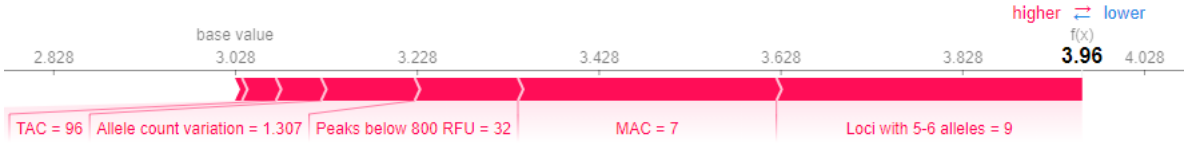


Figure 10: SHAP force plot for regression.

Users generally liked this version better, because they would only need to consult one image and it provides a full summary of what is going on. They also mentioned that handling the NOC problem with regression felt more natural than multi-class classification, as the output is on an ordinal scale. With a bit of explaining, the users could understand that a prediction of 3.96 is more certain than for example 3.55. We also demonstrated that if many red and blue bars are working against each other, that might also indicate the uncertainty of the model.

4.1.3 Conclusion

Users seemed to prefer SHAP values within the context of regression since they only have to analyze one consistent image which represents the entire range of output values. With some coaching, we believe the experts will be able to interpret the SHAP values well enough to provide them some insight into the prediction.

4.2 Anchors

From the initial user study, we found that experts valued the specific values of counterfactuals. For that reason, Anchors seemed like a good method to generate the *factual* side of the explanation. This is because Anchors define feature value ranges, that “anchor” a prediction in a certain region of the feature space [22]. If an instance has feature values that fall within these ranges, we say that the Anchor *holds*. Then, if the Anchor holds, the output can be predicted with high probability. This means that the features not included in the Anchor can vary and the prediction will stay the same. Another advantage from this approach was that these Anchors can be presented in a visually attractive way on the base visualization that we already had designed (see Figure 11). The idea was to combine both Anchors with counterfactuals to incorporate factual- and counterfactual information into one picture.

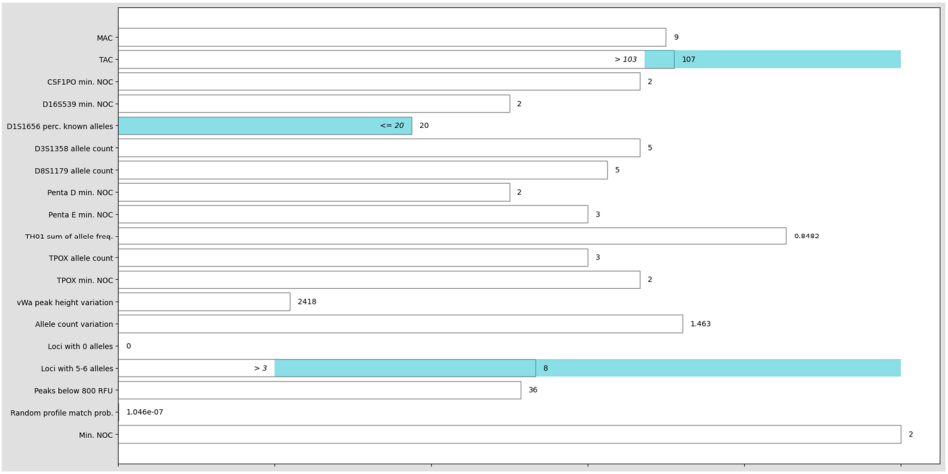


Figure 11: Anchors visualization. The light blue shaded values represent feature ranges for which the same prediction holds.

From the definition, it also seemed that Anchors could be used to create sparse counterfactuals from. Instead of showing the differences in feature values between the input and counterfactual, we could present the differences in their Anchors. As Anchors only consist of a handful of rules, this would create a sparser result.

4.2.1 Matching the counterfactual Anchor

The first experiment explored the idea whether valid counterfactuals could be produced when derived from these differences in Anchors. We used the following steps for some initial experimentation:

1. Generate an Anchor for the input (left side of Table 2).
2. Find the closest training point with a target prediction to be the counterfactual.
3. Generate an Anchor for that counterfactual (right side of Table 2).
4. Change the input to match the counterfactual Anchor (Table 3).
5. Check that the prediction changes to the target.

Anchor for the input instance	Anchor for the counterfactual instance
*The model will predict **3 contributors 96%** of the time when ALL the following rules are true:* - *TAC <= 79.00* - *Loci with 5-6 alleles > 1.00* - *D8S1179 allele count <= 3.00* *These rules hold for the original data with a probability of 0.02*	*The model will predict **2 contributors 97%** of the time when ALL the following rules are true:* - *Allele count std. <= 0.93* - *Random match probability <= 0.00* - *D8S1179 allele count <= 2.00* *These rules hold for the original data with a probability of 0.02*

Table 2: Two Anchors for Experiment 1 that led to a successful result.

	Input value	New value
Allele count std.	1.03	0.70 (<= 0.93)
D8S1179 allele count	3.00	2.00 (<= 2.00)
Prediction	**3.0**	**2.0**

Table 3: Changing the input features that do not match the counterfactual Anchor caused the prediction to be **2.0**.

As can be seen in Table 3, the prediction did change after matching the counterfactual Anchor. However, problems occurred when:

- The input instance already fit the Anchor of the counterfactual.
- Changing the input to fit the Anchor of the counterfactual did not lead to a change in prediction. An example of this is shown in Table 4 and Table 5.

Anchor input	Anchor counterfactual
*The model will predict **1 contributors 93%** of the time when ALL the following rules are true:* - *TPOX min. NOC <= 1.00* - *Allele count std. <= 0.65* *These rules hold for the original data with a probability of 0.29.*	*The model will predict **2 contributors 100%** of the time when ALL the following rules are true:* - *Allele count std. <= 0.83* - *TAC <= 79.00* - *vWa peak height std. > 3024.43* *These rules hold for the original data with a probability of 0.01.*

Table 4: Two Anchors for Experiment 1 that did not lead to a successful result.

	Input value	New value
vWa peak height std.	108	5000 (> 3024)
Prediction	**1.0**	**1.0**

Table 5: Changing the input features that do not match the counterfactual Anchor caused the prediction to stay **1.0.**

So even though the Anchors for the counterfactual prediction were fulfilled, the prediction stayed the same. Perhaps it was caused by the fact that the Anchors for the input also still held. Therefore, we conducted a different experiment.

4.2.2 Mismatching the input Anchor

The procedure is similar to the first experiment with the exception of step 4; here we also want to ensure that the instance no longer matches the input Anchor. We used the same profile as in Table 5.

1. Generate an Anchor for the input (left side of Table 4).
2. Find the closest training point with a target prediction to be the counterfactual.
3. Generate an Anchor for that counterfactual (right side of Table 4).
4. Change the input to match the counterfactual Anchor **and to no longer match the input Anchor** (Table 6).
5. Check that the prediction changes to the target.

	Input value	New value
vWa peak height std.	108	5000 (> 3024)
Allele count std.	0.46	**0.79 (<= 0.83 but not <= 0.65**)
Prediction	**1.0**	**2.0**

Table 6: Changing the input features that do not match the counterfactual Anchor, and did match the original Anchor caused the prediction to change to **2.0**.

However, it apparently matters how much this value is changed. In Table 7, we see that it does not always work.

	Input value	New value
vWa peak height std.	108	5000 (> 3024)
Allele count std.	0.46	**0.70** (<= 0.83 but not <= 0.65)
Prediction	**1.0**	**1.0**

Table 7: By changing the value of Allele count std. slightly less that in the previous attempt, the prediction stayed at **1.0**.

Perhaps this has to do with how Anchors are binned; all data is discretized before Anchors are generated. However, looking at the bins for feature Allele count std., 0.79 and 0.70 do not belong to a different bin.

- 0:'Allele count std. <= 0.38'
- 1:'0.38 < Allele count std. <= 0.45'
- 2:'0.45 < Allele count std. <= 0.65'
- **3:'0.65 < Allele count std. <= 0.83'**
- 4:'0.83 < Allele count std. <= 0.93'
- 5:'0.93 < Allele count std. <= 1.05'
- 6:'1.05 < Allele count std. <= 1.17'
- 7:'1.17 < Allele count std. <= 1.28'
- 8:'1.28 < Allele count std. <= 1.44'
- 9:'Allele count std. > 1.44'

This seemed strange; the Anchor on the right side of Table 4 states that it will predict 2 contributors 100% of the time when the stated rules are true. This started a suspicion that this does not mean that

all other feature values can have any random value. Considering this same counterfactual Anchor from Table 4, setting all other feature values to zero results in a prediction of 1.0. The same prediction apparently does not apply when arbitrary values for the features not included in the Anchor's rules are chosen.

4.2.3 Analysis of experiments

From the experiments, some information had to be uncovered in some more detail.

First of all, though it is described that when an Anchor holds "changes to the rest of the feature values of the instance do not matter" [22], we have seen that is not true for *all* of the feature space. Within the more formal description of the method, one can derive that an Anchor only holds for a sampled subspace around the input instance.

Secondly, this subspace is generated by randomly sampling all feature values not included in the Anchor, which we only uncovered by analyzing the source code. This process likely cannot generate realistic data points for our dataset of highly correlated features. The method description does not clearly specify how sampling is done or how they determine what is still considered local. It is therefore quite difficult to determine for which part of the data the explanation holds. This provides the explanation as to why setting all feature values excluded from the Anchor to zero did not work; an instance where most of the feature values are zero is not part of the local neighborhood of a normal input instance. The precision and coverage that they denote with each Anchor are also based on this perturbation space. Looking at the counterfactual Anchor in Table 4, it holds with a probability of 0.01 for the sampled data. If that Anchor holds, then the prediction of 2 contributors is 100% certain. This does not mean that it applies in the same way for the original data. This could explain why the counterfactual Anchors did not work out:

- The Anchors for seemingly similar instances did not fall into the same *local neighborhood*.
- The sampled data does not represent the dataset accurately, especially since they use *random sampling*, and therefore the specific values of the Anchors do not hold for the dataset.
- Our approach of changing the input instance to match the counterfactual Anchor creates an instance that is unrealistic and/or does not match the perturbed data.

Lastly, Anchors do not inform the user about the most *influential features* for the current prediction which was the initial goal of using Anchors; feature importance with the addition of ranges. The rules that are included in the Anchor are generated stochastically until a certain precision value is reached. This means that the features included in the Anchor are not necessarily the most important ones, but important enough to reach a certain precision. In this way, major contributing features could be excluded from the explanation.

4.2.4 User interpretation

We presented the picture in Figure 11 to the DNA experts in a brainstorming session about the visualization and asked how they interpreted it. Most users saw the Anchors as the most important features for the current prediction. As we determined, this is not the case due to the stochasticity with which the features are added to be part of an Anchor. This stochasticity has another undesirable effect; if the same profile is explained multiple times in a row, different Anchors will be shown. This does not help with user interpretation. Other users have even interpreted that the Anchors represent the only features that *could* be varied, while others should remain the same. Lastly, the precision of the Anchor was mostly misinterpreted as the certainty of the model's current prediction.

After attempting to explain how Anchors are generated, we noticed that the idea of a local neighborhood cannot be translated to layperson-terms. Since the perturbation space cannot be communicated to the user, the context of this explanation is unclear.

The one positive remark about Anchors was the fact that it shows a range. One expert mentioned that in the example of Figure 11, if more alleles were present in the profile by for instance lowering the detection threshold, the TAC would increase. However, as the Anchor specifies that any TAC > 103 holds for this prediction, the same output would occur. This could also be solved by simply inputting a different value into the model, and seeing if the same prediction holds.

4.2.5 Conclusion

The current implementation of Anchors is not applicable to this problem because:

- Anchors do not represent the most important feature values that have led the model to the current prediction, even though users will interpret it as such.
- Anchors can differ between multiple runs, which is confusing for users.
- Anchors do not hold for a real dataset with correlated features because the random sampling process does not generate realistic data.
- Anchors only hold for a local neighborhood, which cannot be communicated clearly to users.

4.3 Counterfactuals

At first, a base counterfactual was implemented. This was simply the closest counterfactual instance from the training data. This instantly uncovered one of the fundamental problems with presenting an instance from the training data; not all of the presented differences are relevant, some might even counteract the prediction. For example, in one of the instances it showed to *increase* the TAC to *decrease* the NOC. This is counterintuitive since more alleles correspond to more contributors. To solve this issue, we considered several options before landing at the final ReCo implementation.

4.3.1 Existing solutions for counterintuitive counterfactuals

One way to mitigate counterintuitive examples, is to show multiple (diverse) profiles so that a more generalized picture is painted to the user [23-25]. However, in this way the users are burdened with having to extrapolate which information is relevant by themselves.

Similarly, by showing the distribution of multiple profiles you can get the same effect [26]. However, distributions are not informative to users that are not familiar with them [10] . They can also occlude local effects that are not visible from a global perspective.

Finally, a suggestion was made to enhance sparsity in training data counterfactuals by introducing a matching tolerance [27]. This means that if the difference between a feature value from the input and the counterfactual is small, for example less than 5%, they would be considered equal. However, we argue that this might overlook the exact threshold values on which a model makes a decision. This could thus lead to missing the target prediction.

4.3.2 Generalized counterfactuals

To counteract the previous issue related to presenting an example data point from the training data, we brainstormed some ideas about how to generalize the counterfactuals.

The first idea was to **cluster** the training points, and then present the median profile of the closest cluster with respect to the input as a counterfactual. This would have the benefit that since it is the median of the cluster, it probably has feature values that are more in consensus with this group of instances, therefore presenting a more generalized counterfactual. However, since there are 19

features per profile, counter-intuitive feature values can still occur. It is also not clear how to choose the extra parameters that come with clustering (such as when to stop), to obtain an optimal result. Lastly, since the counterfactual is in the middle of a cluster, it is likely quite different from the input.

Inspired by the clustering idea, we implemented a **distance kernel** approach. What this entails is that a new instance is generated by taking the average of all training data points, weighted by (1 - their distance to the input profile). In this way, feature values of multiple instances are summarized, but the values of closer instances are incorporated more. This mitigates a lot of the issues with the clustering approach, such as complicated optimization and the distance to the original instance. As such, we implemented the distance kernel and compared the results on our training data to those we obtained from the base counterfactuals. These results are summarized in Table 8.

Metric	**Score distance kernel**	**Score closest training point**
Mean number of feature differences	13 (/19)	8 (/19)
Mean distance to the input	0.123 (/1)	0.039 (/1)

Table 8: Two metrics measured on the distance kernel implementation of counterfactuals, in comparison to the baseline counterfactual. These measurements were performed on the training data.

We can see that the distance kernel already performs worse than the baseline, since both distance to the input and the number of feature differences between the input and counterfactual are high. What we have obtained with the distance kernel is a new data point, which is a summary of multiple other instances. Because it incorporates information from many instances, it is going to have lots of differences in comparison to our input. These many differences also translate into a larger distance.

In conclusion, generalized counterfactuals stray too far from the concept of a counterfactual explanation which is to find an *example* which is the most similar to the input. By presenting a data point that is an amalgamation of other instances, we lose the quirks of what makes an example profile unique. Generalized counterfactuals also stray too far from the input in a literal sense; the distance and number of feature changes is larger than the baseline counterfactual. Presenting a counterfactual profile that has similar peculiarities as the input, even though those do not fit the average, is preferred. The counterintuitive differences thus had to be solved from a different perspective.

4.3.3 Multi-objective counterfactuals

As a way to find sparser training data counterfactuals, we determined that it could be beneficial to select these instances not only based on distance, but on the number of feature differences as well. Minimizing two scores at the same time can be solved with multi-objective optimization. For solving multi-objective optimization problems, there exist several approaches with their own advantages and disadvantages. We considered two simple strategies; weighted-sum and non-dominated.

The simplest solution is considered to be the **weighted-sum** in which scores are collapsed into a single objective by adding them together with predefined weights. This method's main disadvantage is that it is difficult to balance objectives properly. In our case, we only have two objectives; the distance and the number of feature differences. Though both scores lie between zero and one, their medians and variance still differed which can be derived from Table 9.

	Distance score	Number of feature differences score
Median	0.125	0.684
Mean	0.128	0.704
Maximum	0.426	1.000
Minimum	0.004	0.158

Table 9: Median, mean, maximum and minimum of the two scores that we wanted to minimize, based on the training data.

We first thought that it would suffice to assign the weights based on the median of both scores that we obtained from the training data. However, we had no clear perception of how these weights influenced the obtained points. The weighted-sum method has no optimality guarantees. If a new objective was added, or the training data was expanded, the weights would require re-adjusting as well. Even though the method is fast and easy to implement, we did not find it adequate to this problem.

For this reason, we switched to the non-dominated strategy. Compared to the weighted-sum, this technique is guaranteed to find the Pareto-optimum set of solutions [28, 29]. For the sake of the NFI's future plans, more objectives can also be added. One disadvantage is the computational effort, since many comparisons need to be made in order to identify the non-dominated set. For this dataset and problem, this was hardly noticeable in practice.

5. Final user study

It was important to do a soft evaluation get the DNA experts' opinion of the explanation. We also wanted to test if our explanation could help users gain some insight into how the model makes predictions. Therefore, a final survey was created (implemented with Google Forms). There were 8 responses in total, of which 7 were useable.

5.1 Set-up

The main goal of the survey was to establish if our visualization can help users gain insight into how the model makes predictions. By extension, we tested if this information helps regulate the users' trust in the model. This means that when the model is quite sure about a prediction, that this influences the users to trust the prediction more. Conversely, if the model is unsure or incorrect, the explanation should influence the users to doubt the model. For this exercise, we compared against two other state-of-the-art explanations; a SHAP force plot and a counterfactual table.

Besides this regulation of trust, we also wanted to determine if the visualization is user-friendly. The NFI placed emphasis on this latter part, as the users need to want to use it, before they would invest more time into it.

The survey was structured into 5 sections:

1. Demographics.
2. Introduction to the different explanations.
3. Can our visualization increase trust in the prediction when the model seems certain?
4. Can our visualization decrease trust in the prediction when the model seems uncertain or wrong?
5. Which explanation is most user-friendly?

Demographics are important to help understand the background of each participant in case this has an influence on the outcome. Section 2 was essential for users to learn what the different visualizations encode and how to interpret them. In sections 3 and 4 we tested if the visualization can regulate trust, and in section 5 user preferences were evaluated.

5.2 Demographics

There were three relevant demographics questions for the users; age, English reading and listening level, and level of openness towards using machine learning in their decision-making process. Age can influence how open someone is towards technology, or how well they can learn new things [30]. Therefore, we found it useful to gather this information. The survey was created in English to help with the analysis of the results. We were informed that the users understand English well, as they read scientific literature. To ensure that this is indeed the case, we added a question about their English reading and listening level. Lastly, colleagues remarked that some people might be morally opposed to any sort of AI, and therefore might fill in the survey with a negative attitude. Therefore, we asked about their attitude towards using AI beforehand to map out any bias they might have.

5.2.1 Demographics results

A variety of different people participated, with ages spread over the youngest four groups as can be seen in Figure 12.

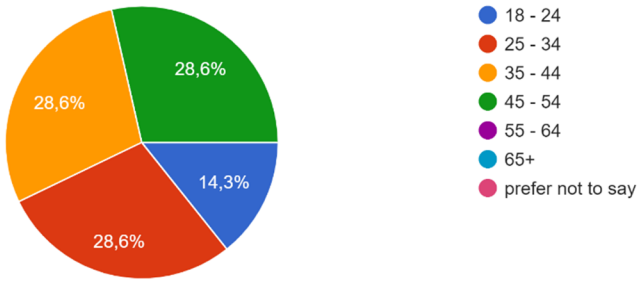


Figure 12: Age demographic answers.

When it comes to English reading and listening ability, every person deemed themselves at least above average which we think is suitable for the survey (see Figure 13).

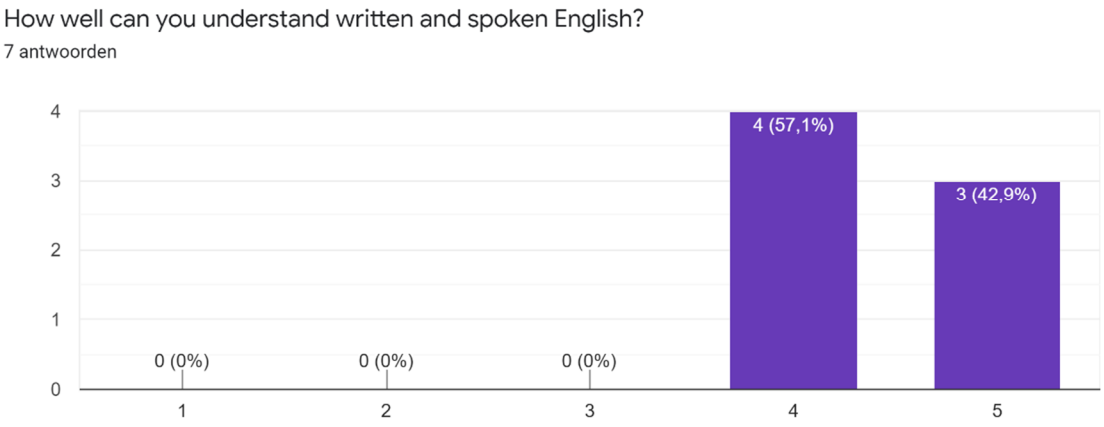


Figure 13: English level answers.

All participants have a positive perception of machine learning as a support tool for their decision-making process, so no upfront negativity was measured (see Figure 14).

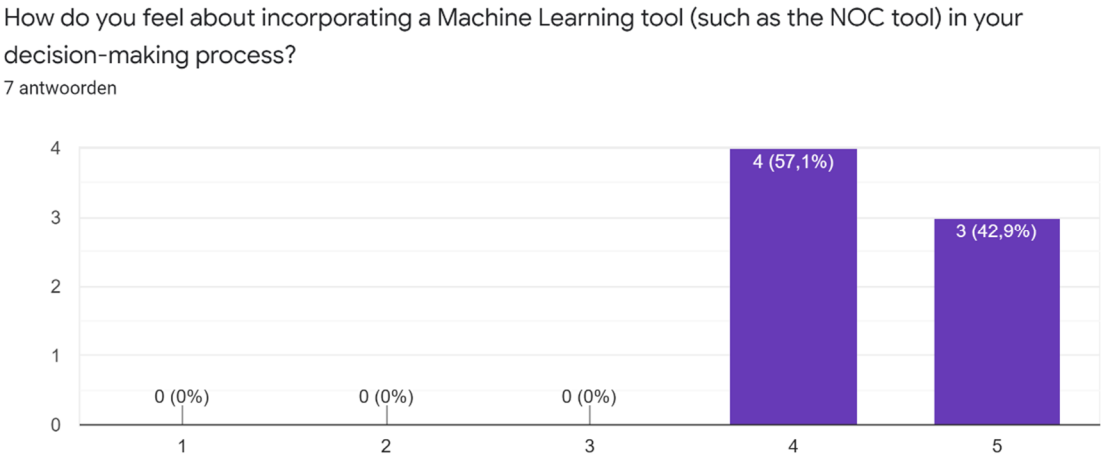


Figure 14: Attitude towards machine learning answers.

No surprising results or particular correlations between the demographics and given answers were found.

5.3 Introduction to explanations

This section was used to introduce the SHAP force plot, counterfactual table, and compound explanation to the users. It included two disclaimers.

The first disclaimer concerned the fact that these explanations were meant to be consulted in a standalone fashion. They do not test how well the user can determine the NOC; they test if users can understand the way the machine learning model makes decisions. However, we understood that some users have some aversion to solely rely on the machine learning model and explanations, and might want to look at the profile anyways. For these users, we added the profiles in their analysis software DNAxs and told them they were free to consult the profile if they do so desire.

Secondly, the DNA experts have repeatedly expressed that they do not understand many of the current 19 features, or how they contribute to making a prediction of the NOC. As such, we had to instruct the users to focus only on a few features. Some of the features are familiar to the users; the *TAC* and the *MAC*. Others are quite simple to understand, such as *Loci with 5-6 alleles*, which represents the number of alleles with 5 or 6 alleles. Some features give an impression of quality of the profile; more *Peaks below 800 RFU* and *Allele count variation* could indicate lower quality, and/or more drop-out. For all locus-specific features, we told the users to view them as indications of the amount of information at each of those loci.

After the two disclaimers, users were presented with an introduction of the three explanations. For each type, we presented an image, an explanatory video, and a short summary of bullet points.

5.3.1 SHAP

Figure 15 shows the SHAP force plot used for introduction.

Figure 15: SHAP force plot used for introduction.

The introductory video can be watched from the following YouTube link: https://www.youtube.com/watch?v=lysnLemJTfg .

The summary of the SHAP explanation consisted of the following points:

- Shows which feature values of this profile the NOC tool used to make this prediction.
- Some values push the prediction down towards 1 or 2 (blue bars), while other values push the prediction up towards 4 or 5 (or more) (red bars). This is relation to the base value of 3.
- Shows only information about the current profile and prediction.

The control question verified that users had read the summary and/or watched the video and understood that the values with red bars push the prediction up:

What is true about the SHAP explanation example we saw?

A. **The value of "Allele count variation = 0.9991" of this profile has caused the NOC tool to make a slightly higher prediction.**
B. The value of "Allele count variation = 0.9991" of this profile has caused the NOC tool to make a slightly lower prediction.

All participants answered this question correctly.

5.3.2 Counterfactual table

Figure 16 shows the counterfactual (CF) table used for introduction.

	2.29		Run 1_Trace 1613475856276
MAC	4.000000	MAC	4.000000
TAC	68.000000	TAC	77.000000
CSF1PO min. NOC	2.000000	CSF1PO min. NOC	2.000000
D16S539 min. NOC	2.000000	D16S539 min. NOC	2.000000
D1S1656 perc. known alleles	20.000000	D1S1656 perc. known alleles	27.272727
D3S1358 allele count	4.000000	D3S1358 allele count	4.000000
D8S1179 allele count	3.000000	D8S1179 allele count	3.000000
Penta D min. NOC	1.000000	Penta D min. NOC	1.000000
Penta E min. NOC	2.000000	Penta E min. NOC	2.000000
TH01 sum of allele freq.	0.346523	TH01 sum of allele freq.	0.636930
TPOX allele count	2.000000	TPOX allele count	2.000000
TPOX min. NOC	1.000000	TPOX min. NOC	1.000000
vWa peak height variation	0.000000	vWa peak height variation	3744.780192
Allele count variation	0.999054	Allele count variation	0.698364
Loci with 0 alleles	0.000000	Loci with 0 alleles	0.000000
Loci with 5-6 alleles	0.000000	Loci with 5-6 alleles	0.000000
Peaks below 800 RFU	2.000000	Peaks below 800 RFU	11.000000
Random match proba.	0.000000	Random match proba.	0.000000
Min. NOC	1.000000	Min. NOC	1.000000
		NOC	3.000000

Figure 16: Counterfactual table used for introduction.

The introductory video can be watched from the following YouTube link: https://www.youtube.com/watch?v=-VRIsHA8Sq4 .

The summary of the CF table explanation consisted of the following points:

- Shows a comparison of the current profile with a profile that had a different rounded-off prediction.
- Differences are highlighted in red boxes.
- We do not know which differences are relevant to arrive at another prediction.

The control question verified that users had read the summary and/or watched the video and understood that the values highlighted in red boxes are simply the differences between two example profiles and they are not all necessarily relevant to change the prediction:

What is true about the CF table explanation example we saw?

A. The feature values highlighted in red boxes have the most influence to change the prediction from 2 to 3 contributors.
B. The feature values highlighted in red boxes are differences between two example profiles with a prediction of 2 and 3 contributors.

All participants answered this question correctly.

5.3.3 Compound visualization

Figure 17 shows the compound visualization used for introduction.

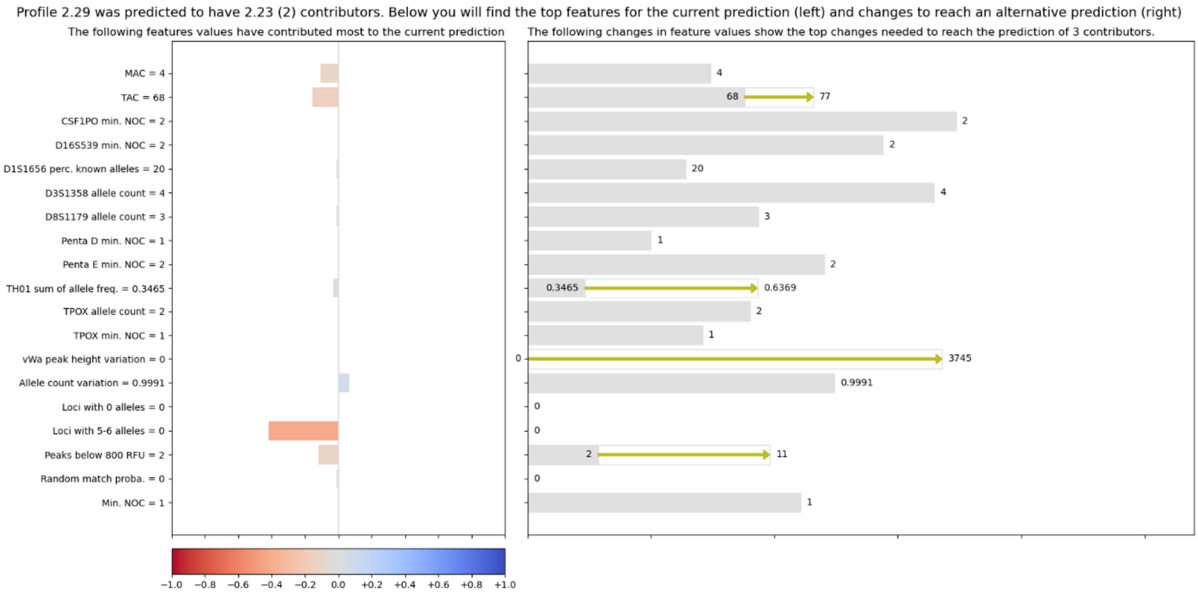


Figure 17: Compound visualization used for introduction.

The introductory video can be watched from the following YouTube link: https://www.youtube.com/watch?v=rz3zm5AQ94c .

The summary of the compound visualization explanation consisted of the following points:

- Consists of two parts; one showing how the feature values have influenced the prediction, one showing the relevant feature value changes needed to reach a different prediction. The only connection that the two parts share, are the feature values.
- Shows how high each feature value (change) is compared to the entire range of that feature.
- By looking at the right side; seeing how large and relevant the necessary feature value changes are, you could determine how clear the NOC tool is about the prediction.

The control questions verified that users had read the summary and/or watched the video and understood that the left side consists of the same values as the SHAP explanation, but the right side is different from the counterfactual table, as only relevant feature value changes are shown. The two sections are not directly connected; we do not know what happens when we change one value:

What is true about the compound explanation example we saw?

A. The left side of the explanation shows the same information as the SHAP explanation, but the right side shows only relevant feature value changes, which is different from the Counterfactual table.

B. The left side of the explanation shows the same information as the SHAP explanation, and the right side shows the same information as the Counterfactual table explanation.

What is true about the compound explanation example we saw?

A. If we change the TAC value from 68 to 77, we know for certain that the TAC value's red bar on the left would become blue.

B. If we change the TAC value from 68 to 77, we do not know for certain how that will affect the red and blue bars on the left, because they only give information about the current profile's feature values.

One participant answered both of these questions wrong, which could mean that they did not watch the video and/or read the summary well. From the open-ended questions, we could also derive that this participant probably did not read all the text in the survey; they filled in that they would need to see the profile before they could make any decisions, yet on every page there were instructions on how to look up the profile if they felt inclined to do so. We therefore had to remove their answer from the responses.

5.4 Regulate trust

At the end of the introduction, we presented the users with the overview in Figure 18 to help them understand this section of the survey. The schematic shows that two profiles will be presented. These two profiles correspond to the two use-cases we described before; one profile was fairly simple for the model to predict and thus the explanation could show this to the user. On the other hand, the second profile was difficult to the model and even led to an incorrect prediction. Per profile, the prediction from the model is presented. This prediction serves as the baseline; there is no explanation. Then, two explanations are presented. For both of these profiles, we will show our compound visualization and another state-of-the-art explanation that is fit for each use case. For profile 1, we compare against a SHAP force plot, while for profile 2 a counterfactual table was used for comparison. Initially, we had planned to compare our visualization against both the SHAP plot and the counterfactual table for each profile, but the survey became too long. As SHAP is originally designed to understand "why a model makes a certain prediction" [21], we deemed it fit for the goal of increasing trust. In contrast, counterfactuals show how a different prediction can be reached. If little change is required for that change to occur, this can decrease trust in the original prediction.

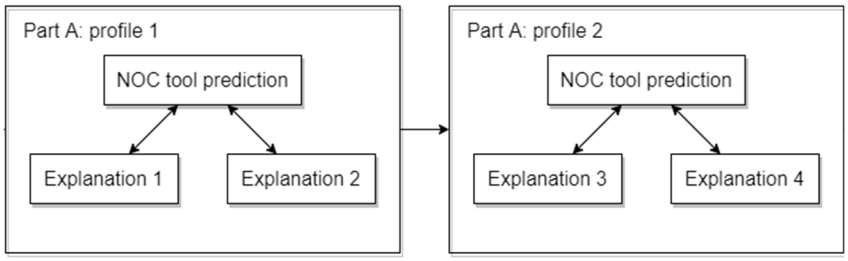


Figure 18: Overview of the survey.

5.4.1 Questions: increase trust

The first use-case that we put to the test was to see if our visualization can increase trust in the prediction when the model seems fairly certain. For this aim, we chose to show profile 1_6B.Trace#01. This profile was chosen because it was difficult in an old NOC interpretation training by the NFI; DNA experts would define this profile to have a NOC of 2 or 3, 3, 3 or 4, 4, 4 or 5. It has a lot of missing alleles, which is why it proved difficult. However, the model correctly identifies it as a 3-person mixture.

The following questions were asked (including Figure 19, Figure 20 and Figure 21):

1. *Please select all number(s) of contributors you would consider after seeing a prediction of 3 (3.22).*
2. *Do you think the prediction of 3 (3.22) is correct?*
3. *Please select all number(s) of contributors you would consider when you can consult this explanation (SHAP). Do you consider the same, or less options than in question 1?*

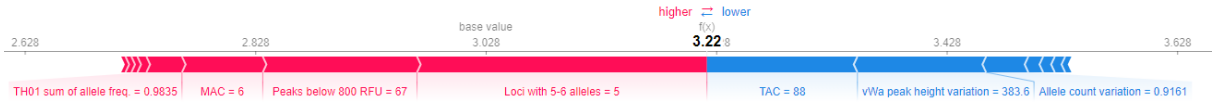


Figure 19: SHAP explanation for profile 1_6B.Trace#01 used in the final survey for question 3.

4. *Do you think the prediction of 3 (3.22) is correct?*
5. *Can you explain why you have answered questions 3 and 4 differently (or not) than questions 1 and 2 after looking at the SHAP explanation, in comparison to only seeing the prediction of 3 (3.22)?*
6. *Please select all number(s) of contributors you would consider when you can consult the explanation (Compound explanation). Do you consider the same, or less options than in question 1? Note: we are comparing with predictions of 2 and 4 donors. Normally, you would be able to choose which comparison you would like to make.*

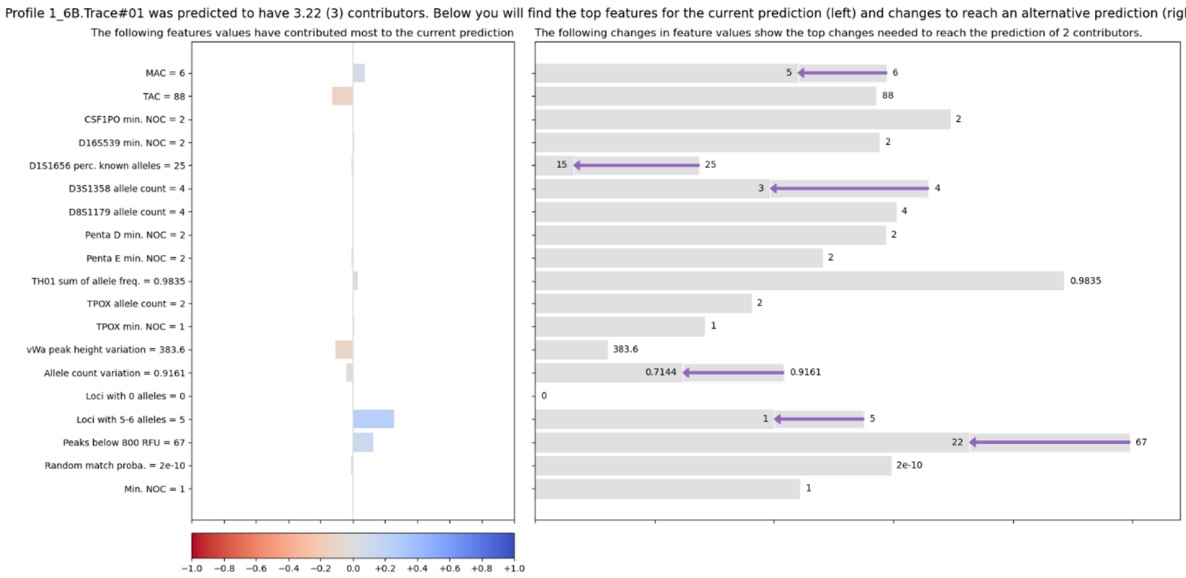


Figure 20: Our visualization for profile 1_6B.Trace#01, comparing to a NOC of 2, used in the final survey for question 6.

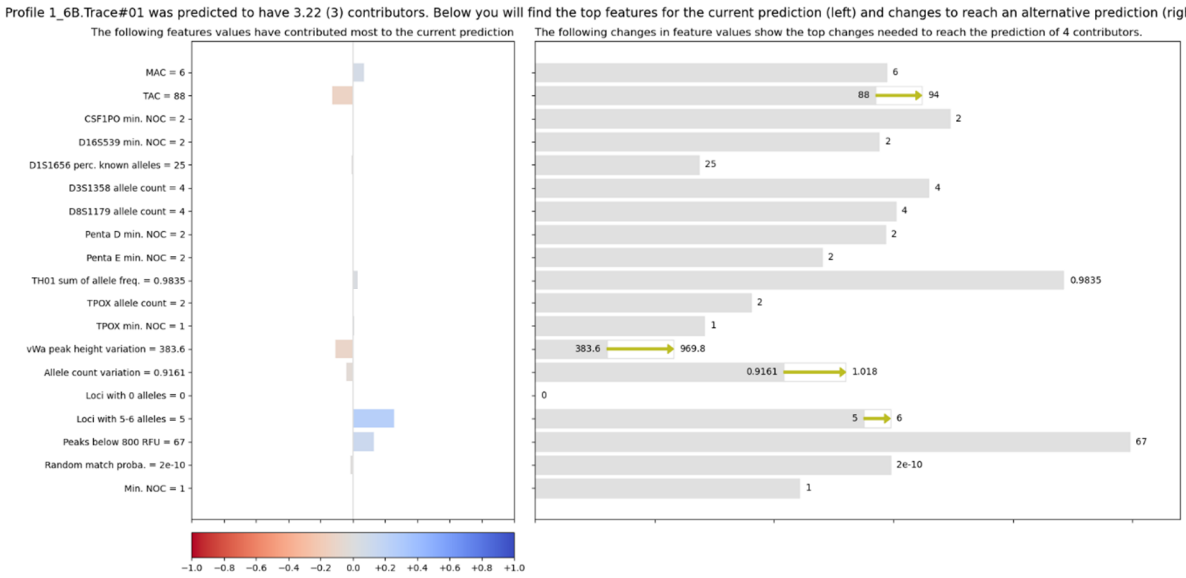


Figure 21: Our visualization for profile 1_6B.Trace#01, comparing to a NOC of 4, used in the final survey for question 6.

7. *Do you think the prediction of 3 (3.22) is correct?*
8. *Can you explain why you have answered questions 6 and 7 differently (or not) than questions 1 and 2 after looking at the Compound explanation, in comparison to only seeing the prediction of 3 (3.22)?*

5.4.2 Answers: increase trust

We wanted to see if our visualization could take away some doubt about the prediction. We compared against the SHAP explanation. Increased trust means that **less options for the NOC** are considered, and that **users think the prediction is correct**.

The answers to questions 1, 3, and 6 are summarized in Figure 22. The influence that the SHAP explanation has on which NOC is considered is shown in the top graph. We can see that most people switched from considering 2, 3 and 4 contributors, to only 3 and 4. The others stayed with the same as what they chose after only seeing the prediction, or even started to consider 2 after previously only considering 3 and 4. There is quite a range of different answers.

Similarly, the influence of our visualization on which NOC is considered is shown in the bottom graph. Most people switched from considering 2, 3 and 4 contributors, to only 3 and 4. The others stayed with their original consideration which lies close to the prediction.

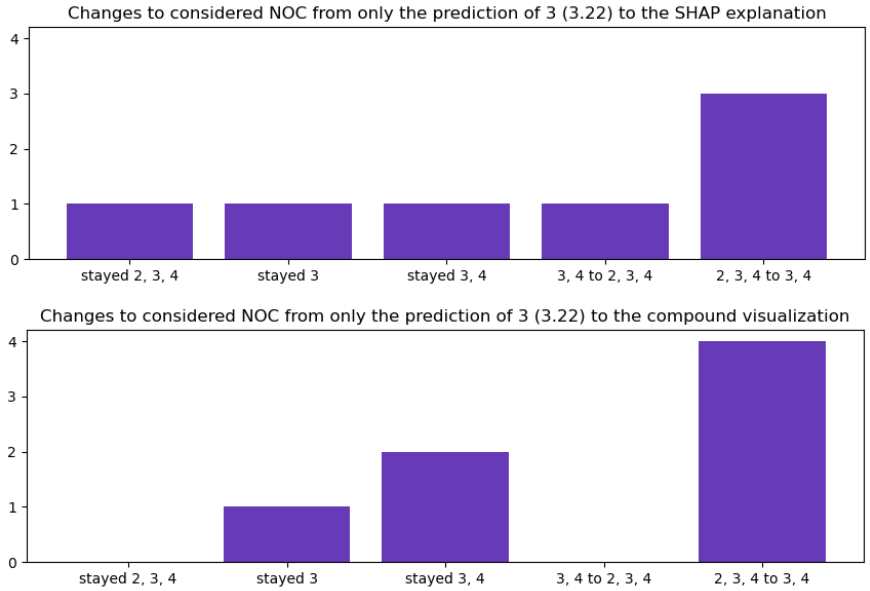


Figure 22: Changes in answers from questions 1 to 3 in the top graph (prediction to SHAP) and from questions 1 to 6 in the bottom graph (prediction to visualization).

A similar trend can be seen in Figure 23 for questions 2, 4 and 7. In general, more people seem to think the prediction is correct after seeing the SHAP explanation (going from the first to the second pie chart). Our visualization has a comparable effect (going from the first to the third pie chart).

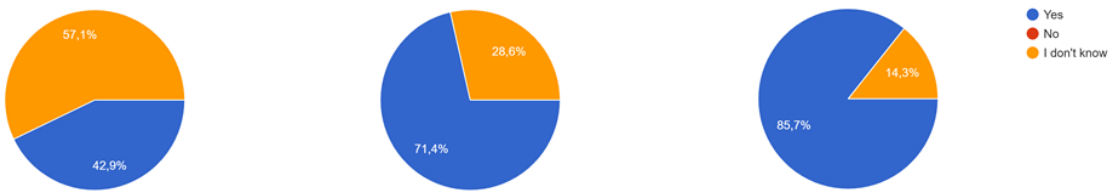


Figure 23: Do participants think that the prediction of 3 (3.22) is correct after question 2 (only the prediction) in the left chart, question 4 (SHAP) in the middle chart, and question 7 (compound visualization) in the right chart.

5.4.3 Motivations: increase trust

The motivations that the participants gave in question 5 concern why they considered different answers after seeing the **SHAP explanation in comparison to no explanation**.

There were a few users that expressed to no longer consider 2 contributors because of feature values. One noted the MAC and TAC in combination with the high number of peaks below 800 RFU would not suit 2 donors, another mentioned the high number loci with 5-6 alleles. Another participant came to the same conclusion because there are more values pushing the prediction up than down.

Interestingly, one user actually switched from only considering 3 and 4, to also including 2 because the red blocks are located all the way into the 2.6 area. Another user became similarly confused as the red and blue bars seem to pull the prediction to both sides. They both answered "I don't know" instead of "yes" to whether they think the prediction is correct after seeing the SHAP explanation.

The two participants that stayed with their original answers of 3 and 2, 3, 4 had similar reasoning where they expressed to trust the prediction as many changes would need to be made to reach a different prediction. One person seemed more inclined to just take the multiple options either way.

The motivations noted from question 8 show why participants gave different answers after seeing our **compound visualization in comparison to no explanation**.

Most users noticed that to reach a prediction of 2 contributors, a lot of feature values needed to change, and also by a large extent. Especially in comparison to what needs to be altered to reach a prediction of 4 contributors. That is why they no longer considered 2 contributors. One user came to this conclusion because they thought the compound visualization indicated some of the change that they considered themselves (lower MAC and peaks below 800 RFU).

The person who stayed with 3 and 4 mentioned that they would like to look at the EPG before making a decision. The person to remained at 3 noted that they report a minimum NOC, so 3 would be certain enough.

5.4.4 Questions: decrease trust

The second use-case that we put to the test was to see if our visualization can make users trust the prediction less when the model seems uncertain or is simply wrong. For this aim, we chose to show profile 2A3.3. This 3-person mixture profile was predicted by the model to have 4 contributors. As the output is 3.53, we can tell that the model is unsure about this profile.

The following questions were asked (including Figure 24, Figure 25 and Figure 26):

1. *Please select all number(s) of contributors you would consider after seeing a prediction of 4 (3.53)*
2. *Do you think the prediction of 4 (3.53) is correct?*
3. *Please select all number(s) of contributors you would consider when you can consult this explanation (Counterfactual table). Do you consider the same, or less options than in question 1? Note: we are comparing with example profiles with a prediction of 3 and 5 donors. Normally, you would be able to choose which comparison you would like to make.*

	2A3.3		2C3.3
MAC	6.000000	MAC	6.000000
TAC	98.000000	TAC	96.000000
CSF1PO min. NOC	2.000000	CSF1PO min. NOC	2.000000
D16S539 min. NOC	2.000000	D16S539 min. NOC	2.000000
D1S1656 perc. known alleles	25.000000	D1S1656 perc. known alleles	25.000000
D3S1358 allele count	5.000000	D3S1358 allele count	5.000000
D8S1179 allele count	4.000000	D8S1179 allele count	4.000000
Penta D min. NOC	3.000000	Penta D min. NOC	3.000000
Penta E min. NOC	3.000000	Penta E min. NOC	3.000000
TH01 sum of allele freq.	0.848201	TH01 sum of allele freq.	0.848201
TPOX allele count	3.000000	TPOX allele count	3.000000
TPOX min. NOC	2.000000	TPOX min. NOC	2.000000
vWa peak height variation	906.029889	vWa peak height variation	430.111334
Allele count variation	0.987636	Allele count variation	0.916144
Loci with 0 alleles	0.000000	Loci with 0 alleles	0.000000
Loci with 5-6 alleles	11.000000	Loci with 5-6 alleles	10.000000
Peaks below 800 RFU	5.000000	Peaks below 800 RFU	7.000000
Random match proba.	0.000000	Random match proba.	0.000000
Min. NOC	2.000000	Min. NOC	2.000000
		NOC	3.000000

	2A3.3		5.27
MAC	6.000000	MAC	6.000000
TAC	98.000000	TAC	101.00000
CSF1PO min. NOC	2.000000	CSF1PO min. NOC	2.000000
D16S539 min. NOC	2.000000	D16S539 min. NOC	2.000000
D1S1656 perc. known alleles	25.000000	D1S1656 perc. known alleles	25.000000
D3S1358 allele count	5.000000	D3S1358 allele count	5.000000
D8S1179 allele count	4.000000	D8S1179 allele count	4.000000
Penta D min. NOC	3.000000	Penta D min. NOC	2.000000
Penta E min. NOC	3.000000	Penta E min. NOC	3.000000
TH01 sum of allele freq.	0.848201	TH01 sum of allele freq.	0.683693
TPOX allele count	3.000000	TPOX allele count	3.000000
TPOX min. NOC	2.000000	TPOX min. NOC	2.000000
vWa peak height variation	906.029889	vWa peak height variation	302.477600
Allele count variation	0.987636	Allele count variation	0.920261
Loci with 0 alleles	0.000000	Loci with 0 alleles	0.000000
Loci with 5-6 alleles	11.000000	Loci with 5-6 alleles	10.000000
Peaks below 800 RFU	5.000000	Peaks below 800 RFU	14.000000
Random match proba.	0.000000	Random match proba.	8.60e-08
Min. NOC	2.000000	Min. NOC	2.000000
		NOC	5.000000

Figure 24: CF table explanation for profile 2A3.3, comparing to a NOC of 3 and 5, used in the final survey for question 3.

4. *Do you think the prediction of 4 (3.53) is correct?*
5. *Can you explain why you have answered questions 3 and 4 differently (or not) than questions 1 and 2 after looking at the Counterfactual table explanation, in comparison to only seeing the prediction of 4 (3.53)?*
6. *Please select all number(s) of contributors you would consider when you can consult the explanation below (Compound explanation). Do you consider the same, or less options than in question 1? Note: we are comparing with predictions of 3 and 5 donors. Normally, you would be able to choose which comparison you would like to make.*

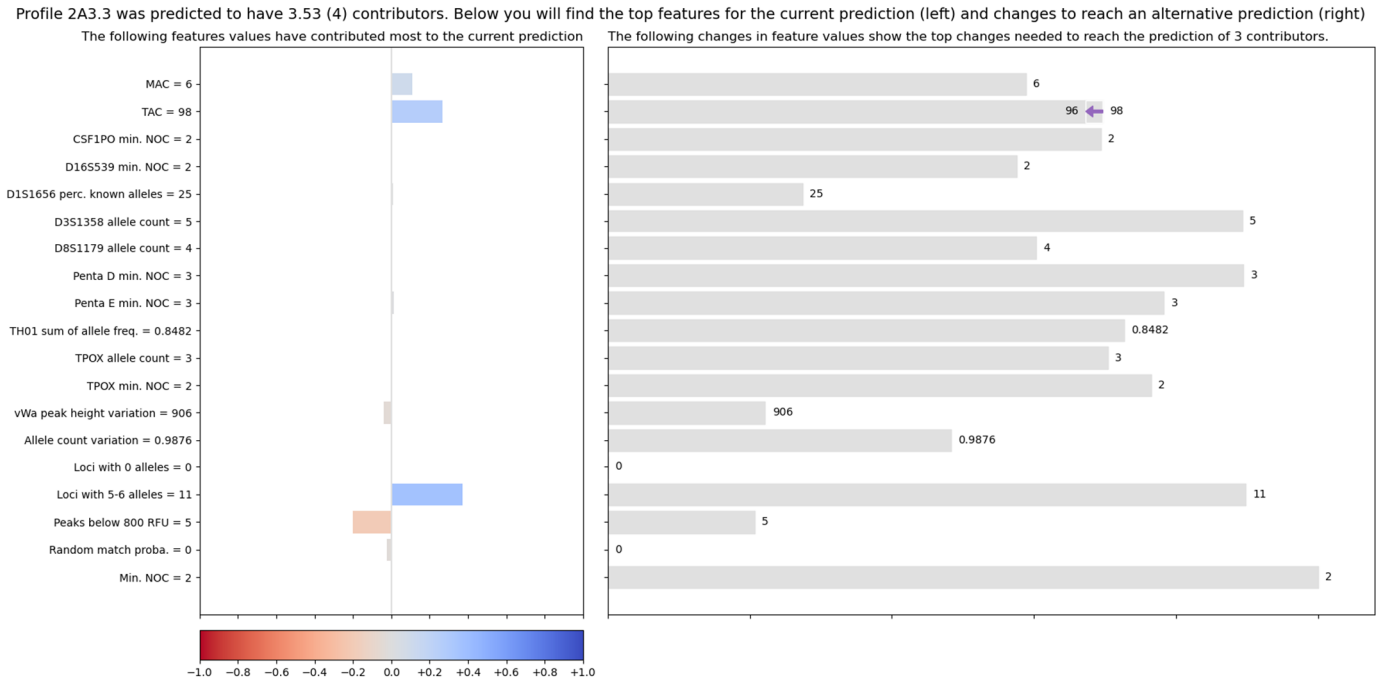


Figure 25: Our visualization for profile 2A3.3, comparing to a NOC of 3, used in the final survey for question 6.

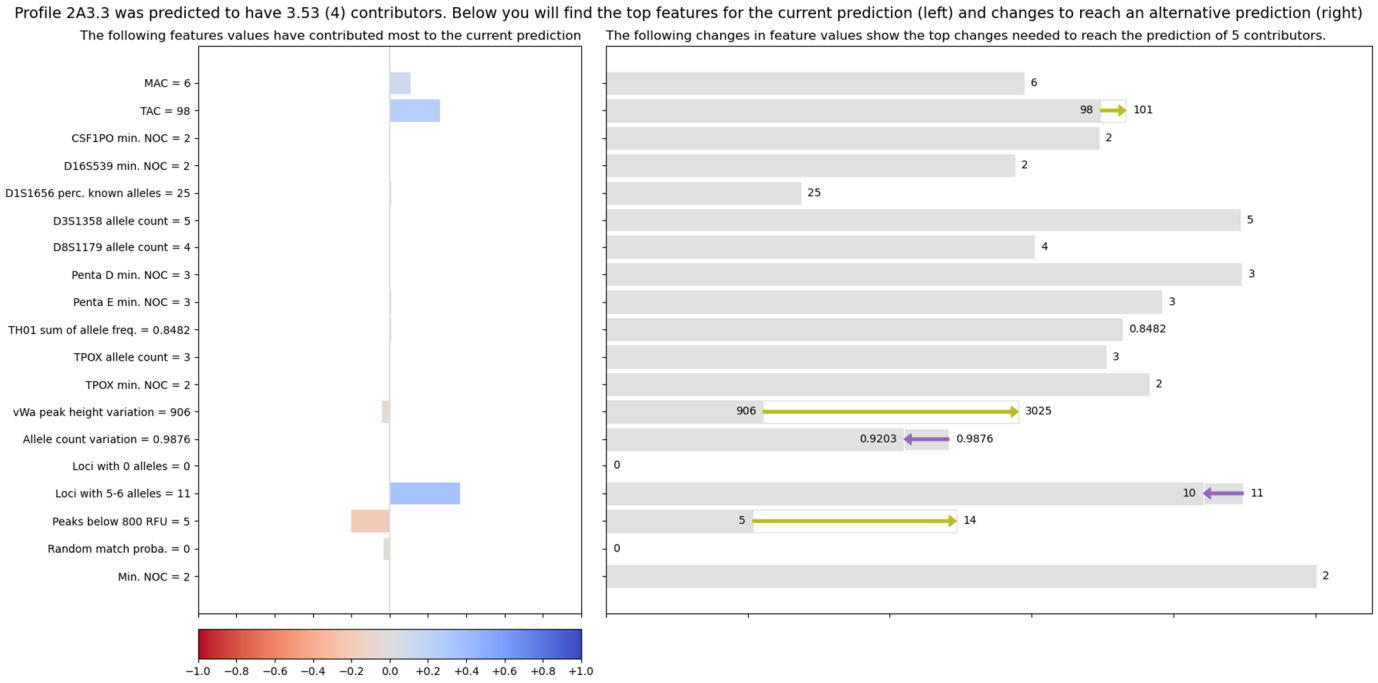


Figure 26: Our visualization for profile 2A3.3, comparing to a NOC of 5, used in the final survey for question 6.

7. *Do you think the prediction of 4 (3.53) is correct?*
8. *Can you explain why you have answered questions 6 and 7 differently (or not) than questions 1 and 2 after looking at the Compound explanation, in comparison to only seeing the prediction of 4 (3.53)?*

5.4.5 Answers: decrease trust

We wanted to see if our visualization can decrease trust in the prediction. We compared against a counterfactual (CF) table explanation. Decreased trust means **no less options for the NOC** are considered, and that **users doubt that the prediction is right**.

The answers to questions 1, 3, and 6 are summarized in Figure 27. The influence that the CF table explanation has on which NOC is considered is shown in the top graph. We can see that most people stayed with their initial estimation of 3 or 4 (or 5) contributors. One person dropped their consideration of 2 and 5 donors. Only one person was drastically different as they changed their answer to match the prediction of only 4.

The answers are largely the same for the compound visualization. The person that thought it was 4 contributors after the CF table, changed it back to 3 and 4. One of the people who first considered 3 and 4, thought it was 3 instead; they went against the prediction.

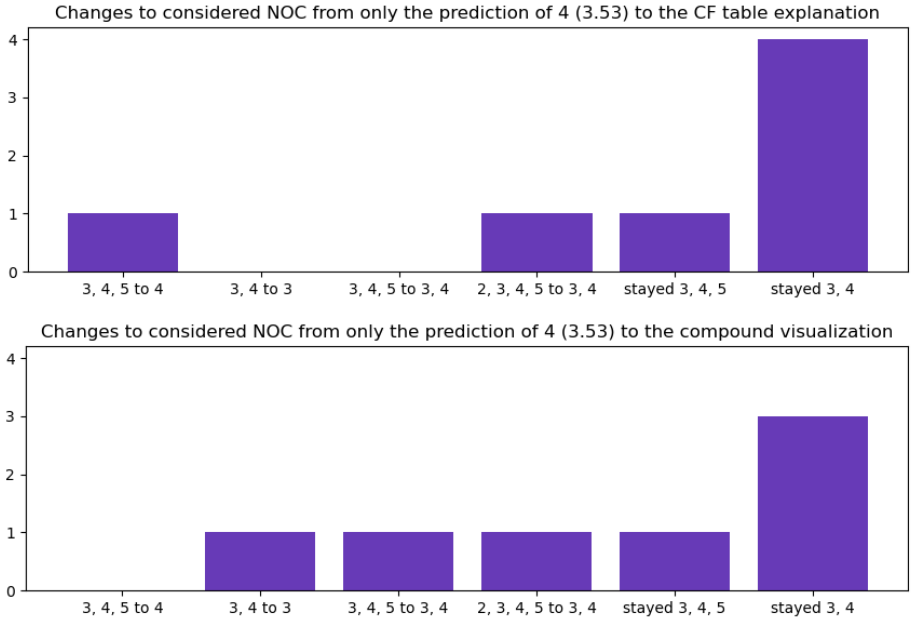


Figure 27: Changes in answers from questions 1 to 3 in the top graph (prediction to CF table) and from questions 1 to 6 in the bottom graph (prediction to visualization).

A similar trend can be seen in Figure 28 for questions 2, 4 and 7. Most people do not know if the prediction is correct or not. The CF table has no influence on this trust (going from the first to the second pie chart). With our visualization, one person changed their answer from "I don't know" to "No", and another changed from "Yes" to "I don't know" (going from the first to the third pie chart).

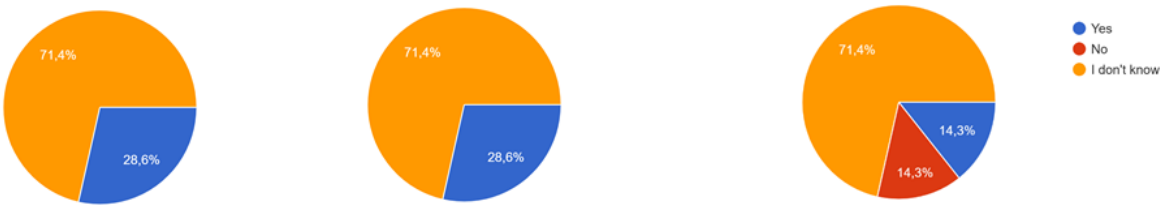


Figure 28: Do participants think that the prediction of 4 (3.53) is correct after question 2 (only the prediction) in the left chart, question 4 (CF table) in the middle chart, and question 7 (compound visualization) in the right chart.

5.4.6 Motivations: decrease trust

The motivations that the participants gave in question 5 concern why they considered different or similar answers after seeing the **CF table in comparison to no explanation**.

Two users expressed that the model seemed to be uncertain when looking at the prediction of 3.53 and therefore thought it should be 3 or 4. This means that the CF table did not influence their decision in any other direction as opposed to seeing only the prediction. The other two users that picked 3 or 4 contributors, noted that in the CF table there are less changes to be made to reach 3 contributors than to reach 5.

The person that ruled out 2 and 5 contributors, based this decision on the feature values, as 2 donors with these *MAC*, *TAC* and *peaks below 800 RFU* values seems improbable, and 5 donors are unlikely based on *vWa* and *peaks below 800 RFU* values.

The one person that chose 4 contributors noted that to reach 3 or 5, lots of change was required.

One user still considered their original three options as they noted that the counterfactual table does not show which feature differences are actually relevant.

The motivations noted from question 8 show why participants gave different answers after seeing our **compound visualization in comparison to no explanation**.

Five users noted that they were starting to doubt the model's prediction of 4, as only minor changes were required to reach the prediction of 3 (change TAC of 98 to 96). One participant noted that for a TAC of 98, there can be 2 artefact peaks. They therefore thought the prediction was incorrect.

One person noticed how it is easier to determine to what extend features need to change to reach the alternative predictions as compared to the CF table.

5.5 User friendliness

As subjective evaluation of user friendliness, the users were asked to pick which explanations they preferred in three categories:

- Ease of use (how easily they could find the relevant information)
- Appeal (how nice they thought it was to use)
- Completeness (how well they could form a total picture of the prediction)

For each category, only one explanation could be selected. The results can be found in Figure 29.

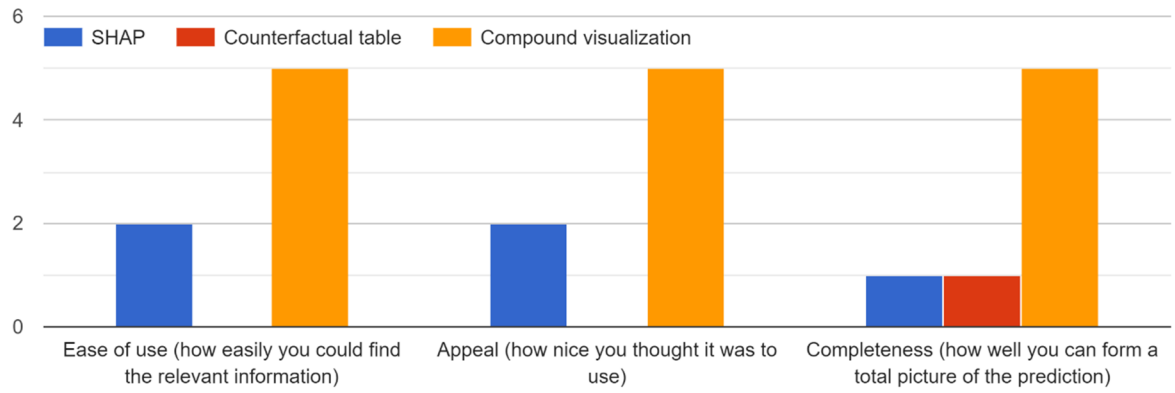


Figure 29: Results of subjective evaluation of the 3 presented explanations.

In general, the compound visualization scored well on all three categories. The main reasons for picking the visualization were that:

- There was more information (2x).
- There was less information, but the information was more relevant (1x).
- The information was easy to find (3x).
- It was presented in a visual way (5x).
- It was easier to understand (1x).
- It was more like how I would think about this problem (1x).

One user admitted to being a fan of the SHAP method and therefore chose it for all three categories. They expressed it was easy to use, the information was easy to find and the results were presented in a visual way.

5.6 Discussion and conclusion

Before we conclude, we note that there were some threats to validity that could have influenced the results.

First of all, the *ordering* of the explanations was SHAP first, CF table second, and our visualization last for all participants. This order was chosen as such because the compound visualization consists of SHAP values, and a modified counterfactual. It therefore makes sense to first explain the components, before explaining the whole. This could have introduced some bias in favor of SHAP and against our visualization, though we did not notice this in the results.

Secondly, we missed the fact that when the first explanation was shown for both profiles, this was also the first time that the participants came into contact with the *feature values* of the profiles. Some users focused on the feature values instead of on what the explanation was trying to communicate. With the features being quite difficult to understand, the translation into simpler terms was required for the users to have some idea of what they meant. However, this is an extra step that users needed to take before being able to understand the explanation. This could have caused fatigue or made participants feel unmotivated. The same can be said for the introductory section. Even though they enjoyed the way that it was explained, it still took a toll on their capacity for the rest of the survey.

Thirdly, a threat to construct validity was the fact that DNA experts at NFI report a *minimum* NOC. Because of this fact, a participant might have been inclined to choose a NOC of 3 instead of 4, not because the explanation made the prediction of 3 seem more likely. We noticed one participant that seemed to hint at this fact.

Lastly, we only received 7 replies from a group of about 35 experts. This selection of subjects means that the survey suffered from *representation bias*.

Because of these limitations, we approached the final evaluation as a subjective collection of participants' opinions, rather than try to perform statistical tests. The addition of textual responses was especially valuable to see how people thought about the explanations.

Within the section of increasing trust, SHAP seemed to induce some confusion with the bars pushing into each other. Most of the users that did express to drop one option, based this on the feature values, not the actual information that the SHAP plot was trying to communicate. In our visualization, users had the option to explore all feature values, as well as see how these values should be changed to reach a different outcome. The participants dropped one option because many feature values had to be changed and also by a large extent. One person noted that they dropped one option because the

counterfactual section of the visualization suggested the same changes in feature values that they considered themself.

When the model made a wrong prediction, the counterfactual table did not seem to influence many people's decision. Some participants noted the feature values as reasons to drop one NOC from consideration, and noticed that the prediction value was on a threshold. These facts are not specific to the CF table. Only a few users noticed that fewer, or less important features needed to change to reach 3 donors. With the help of the compound visualization, more users started to doubt the model's prediction, or even deemed it incorrect as no decision should be made on a difference of 2 alleles. One person noticed how it is easier to determine to what extend features need to change to reach the alternative predictions as compared to the CF table.

Despite the small group of people that participated, the uncontrolled environment, and various sources of bias, the results were insightful about the experts' various perceptions. They could be used to guide further direction, and generally show promise that experts could use our compound visualization as a tool to gain more insight into predictions of the model. We are mainly proud that it can help pinpoint the number of contributors better, and in some cases adjust trust in the prediction appropriately. Before such explanations can be used in practice however, training is required to help users become more familiar with them in practice.

6. Future feature engineering

Throughout the project, many of the DNA experts expressed that they did not like the current features, as they are too complicated, are incomplete, and do not seem to represent information that the experts think relates to the number of contributors. They have given us a lot of information on how to possibly improve the features in the future. From working with the features, we also uncovered some insights related to their use from a machine learning perspective. As such, we have accumulated the following recommended changes to the features:

1. Remove redundant features. For example, MinNOC*locus* encodes the same information as AlleleCount*locus*. MinNOC*locus* is equal to AlleleCount*locus* divided by two, rounded down. The one feature is not going to give more information than the other.
2. Remove locus-specific features. The loci that are included in the 19 features appear to be the loci that most often have the MAC of the profile. It would be interesting to replace the 23 locus-specific features with the following:
    - Locus (/loci) with the maximum allele count of the profile.
    - Locus (/loci) with the minimum allele count of the profile.

    The opinion of the users is that mostly profile-specific information is important. It is difficult to understand why certain loci are included in the current features and others are not. This proposal might remove that confusion. The way that the original 19 features were selected, was based on little data. This could be the reason why these locus-specific features were added.

3. Make profile features complete. For instance, the features now include the number of loci with 5 or 6 alleles, but not the number of loci with 7 or 8 alleles. Including the full range helps with consistency and understanding in explanations.
4. Add more information outside of the STR profile such as:
    - Number of peaks at stutter positions.
    - Number of unnamed peaks (visible in epg, can be sampled in EuroForMix).
    - Replicate runs.
    - Major / minor contributors.
    - The quality of certain channels.

    Some of these might be more viable and useful than others. They are ordered according to the expected feasibility. Most of these were brought up by the DNA experts.

5. Keep MAC and TAC features since they are very familiar to the users and give context to the complete explanation. Additionally, the TAC values can vary among different kits, which the users might not have experienced yet. Seeing the familiar variables with new values can help users get used to a new kit.

7. Reflection on methodology

Despite confirming with supervisors from the NFI and TU Delft that a linear strategy would suffice, we quickly realized that this would make it difficult to guard the quality of the project. It was a good decision to switch strategy to Agile such that new goals could be set and evaluated every three weeks. In this way, small increments are made to the product and less successful endeavors do not hinder progress as much. Perhaps this shows how little the TU Delft focusses on project methodology, as we relied on past experiences to organize this work.

With similar flexibility, we adapted the main research question to better fit the needs of users. Where we originally wanted to only focus on the counterfactuals because it is an interesting field of research, the users made us realize that a combination with feature importance is more valuable.

One aspect to improve upon is to have clearer agreements about the available time from the DNA experts and how many would be willing to participate. It seemed quite difficult to get a company to give some of their employees' time to an intern. Though we identified this risk from the start (see page 3), and we agreed with the NFI upon two surveys with users (one short and one long), no specific constraints were set about how many people would participate or for what period of time. Without Covid-19 restrictions, a fixed date and time could be planned for a sit-down evaluation at a specified location. This provides a lot of control as compared to sending a survey and hoping people will take the time to respond. Such a meeting could be communicated with the DNA experts' management as well so that this could be planned into their schedule. In this way, we could have presented each of the explanations with a suitable introduction where users could ask questions. This ensures that everyone understands the concepts before proceeding. In the final survey for example, we had to eliminate the response of one user because they answered the control questions incorrectly. If we were there to support that person, the outcome might have been different. Within Covid restrictions, a specific moment could still have been planned, but it would lose some of the controlled setting.

There was quite some pressure on the work; The thesis was completed in seven months instead of the usual nine, as the defense date was quickly set for mid-June. Perhaps it would be better to establish beforehand the expectations about this set date, before committing to it fully. This might have mitigated some stress. However, we are grateful to have finished the thesis before the summer.

References

1. Benschop, C.C.G., et al., *Automated estimation of the number of contributors in autosomal short tandem repeat profiles using a machine learning approach.* Forensic Science International: Genetics, 2019. **43**: p. 102150.
2. Benschop, C.C.G., et al., *The effect of varying the number of contributors on likelihood ratios for complex DNA mixtures.* Forensic Science International: Genetics, 2015. **19**: p. 92-99.
3. Adadi, A. and M. Berrada, *Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI).* IEEE Access, 2018. **6**: p. 52138-52160.
4. Barredo Arrieta, A., et al., *Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.* Information Fusion, 2020. **58**: p. 82-115.
5. Carvalho, D.V., E.M. Pereira, and J.S. Cardoso, *Machine learning interpretability: A survey on methods and metrics.* Electronics (Switzerland), 2019. **8**(8).
6. Gilpin, L.H., et al. *Explaining Explanations: An Overview of Interpretability of Machine Learning*. in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. 2018.
7. Lipton, Z.C., *The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.* Queue, 2018. **16**(3).
8. Lundberg, S.M., et al., *Explainable machine-learning predictions for the prevention of hypoxaemia during surgery.* Nat Biomed Eng, 2018. **2**(10): p. 749-760.
9. Yoo, T.K., et al., *Explainable Machine Learning Approach as a Tool to Understand Factors Used to Select the Refractive Surgery Technique on the Expert Level.* Transl Vis Sci Technol, 2020. **9**(2): p. 8.
10. Miller, T., *Explanation in artificial intelligence: Insights from the social sciences.* Artificial Intelligence, 2019. **267**: p. 1-38.
11. Verma, S., J.P. Dickerson, and K. Hines, *Counterfactual Explanations for Machine Learning: A Review.* ArXiv, 2020. **abs/2010.10596**.
12. Taylor, D., J.-A. Bright, and J. Buckleton, *Interpreting forensic DNA profiling evidence without specifying the number of contributors.* Forensic Science International: Genetics, 2014. **13**: p. 269-280.
13. Boavida, A., et al., *PowerPlex® fusion 6C system: internal validation study.* Forensic sciences research, 2018. **3**(2): p. 130-137.
14. Intituut, N.F., *HBS: Interpretatie van autosomale STR DNA-profielen*. 2020. p. 16.
15. Bleka, Ø., G. Storvik, and P. Gill, *EuroForMix: An open source software based on a continuous model to evaluate STR DNA profiles from a mixture of contributors with artefacts.* Forensic Science International: Genetics, 2016. **21**: p. 35-44.
16. Coble, M.D., et al., *Uncertainty in the number of contributors in the proposed new CODIS set.* Forensic Science International: Genetics, 2015. **19**: p. 207-211.
17. Clayton, T.M., et al., *Analysis and interpretation of mixed forensic stains using DNA STR profiling.* Forensic Science International, 1998. **91**(1): p. 55-70.
18. Benschop, C., *PowerPlex Fusion 6C Profile analysis & interpretation.* 2020.
19. Haned, H., et al., *Estimating the Number of Contributors to Forensic DNA Mixtures: Does Maximum Likelihood Perform Better Than Maximum Allele Count?* Journal of Forensic Sciences, 2011. **56**(1): p. 23-28.
20. Benschop, C., A. Backx, and T. Sijen, *Automated estimation of the number of contributors in autosomal STR profiles.* Forensic Science International: Genetics Supplement Series, 2019. **7**.
21. Lundberg, S. and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*. 2017.
22. Ribeiro, M.T., S. Singh, and C. Guestrin. *Anchors: High-Precision Model-Agnostic Explanations*. in *AAAI*. 2018.
23. Wachter, S., B. Mittelstadt, and C. Russell, *Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR.* Harvard journal of law & technology, 2018. **31**: p. 841-887.

24. Schleich, M., et al., *GeCo: Quality Counterfactual Explanations in Real Time*. 2021.
25. Russell, C. *Efficient search for diverse coherent explanations*. 2019.
26. Gomez, O., et al. *ViCE*. 2020.
27. Keane, M. and B. Smyth, *Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI)*. 2020.
28. Gunantara, N., *A review of multi-objective optimization: Methods and its applications.* Cogent Engineering, 2018. **5**(1): p. 1502242.
29. Chiandussi, G., et al., *Comparison of multi-objective optimization methodologies for engineering applications.* Computers & Mathematics with Applications, 2012. **63**(5): p. 912-942.
30. Rojas-Mendez, J., A. Parasuraman, and N. Papadopoulos, *Demographics, attitudes, and technology readiness: A cross-cultural analysis and model validation.* Marketing Intelligence & Planning, 2017. **35**: p. 18-39.