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

Veldhuis, M.S.; Ariëns, Simone; Ypma, Rolf J.F.; Abeel, T.E.P.M.F.; Benschop, Corina C.G.

DOI
10.1016/j.fsigen.2021.102632

Publication date
2022

Document Version
Final published version

Published in
Forensic Science International

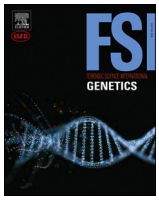
Citation (APA)
Veldhuis, M. S., Ariëns, S., Ypma, R. J. F., Abeel, T. E. P. M. F., & Benschop, C. C. G. (2022). Explainable artificial intelligence in forensics: Realistic explanations for number of contributor predictions of DNA profiles. *Forensic Science International*, *56*, Article 102632. https://doi.org/10.1016/j.fsigen.2021.102632

Forensic Science International: Genetics 56 (2022) 102632

Contents lists available at ScienceDirect

Forensic Science International: Genetics

journal homepage: www.elsevier.com/locate/fsigen



Research paper

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

Marthe S. Veldhuis [a,b], Simone Ariëns [b], Rolf J.F. Ypma [b], Thomas Abeel [a], Corina C. G. Benschop [c,*]

[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

ARTICLE INFO

Keywords:
Number of contributors
Explainable artificial intelligence
DNA mixtures
Machine learning
Counterfactual explanations

ABSTRACT

Machine learning obtains good accuracy in determining the number of contributors (NOC) in short tandem repeat (STR) mixture DNA profiles. However, the models used so far are not understandable 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, more understandable models that achieve lower accuracy, we use techniques that can be applied to any machine learning model. Our explanations incorporate SHAP values and counterfactual examples for each prediction into a single visualization. Existing methods for generating counterfactuals focus on uncorrelated features. This makes them inappropriate for the highly correlated features derived from STR data for NOC estimation, as these techniques simulate combinations of features that could not have resulted from an STR profile. For this reason, we have constructed a new counterfactual method, Realistic Counterfactuals (ReCo), which generates realistic counterfactual explanations for correlated data. We show that ReCo outperforms state-of-the-art methods on traditional metrics, as well as on a novel realism score. A user evaluation of the visualization shows 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

The interpretation of DNA profiles consists of several steps, including estimating the number of contributors (NOC). Determining the NOC from short tandem repeat (STR) profiles can be 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 [10–12]. However, most probabilistic genotyping software that is used for weight of evidence calculations does require the user to supply the NOC [13,14], which can have profound impact on the size of the likelihood ratio returned [2,14–19].

Valuable steps have been taken to develop methods for accurately predicting the NOC. A simplistic method is provided by the maximum allele count (MAC), which constitutes taking the locus with the most alleles, dividing by two and rounding up [20]. Improvement of this method comes from incorporating more information such as the total allele count (TAC), peak heights, drop out and stutter rates, the distribution of allele counts, and population allele frequencies [3,5,8,9]. More complex techniques like Bayesian networks are also used [4]. Machine learning models have shown to outperform other methods on both accuracy and speed, though the quality of the models is highly dependent on the training data [16,21–24]. However, many machine learning algorithms are considered to be *black boxes* [25–32], because the predictions they produce are made based on generalization from training data and the exact mechanism is not easily understood. Knowledge on a model's performance can be obtained through validation and training, though it may remain unclear to DNA experts *why* the algorithm or model made a certain prediction. To gain knowledge on whether to trust the outcome on a per-case basis, the expert may learn from the model if it considered some information that the expert did not consider.

* Corresponding author.
E-mail addresses: msveldhuis96@gmail.com (M.S. Veldhuis), s.ariens@nfi.nl (S. Ariëns), r.ypma@nfi.nl (R.J.F. Ypma), c.benschop@nfi.nl (T. Abeel), t.abeel@tudelft.nl (C.C.G. Benschop).

https://doi.org/10.1016/j.fsigen.2021.102632
Received 16 July 2021; Received in revised form 8 November 2021; Accepted 9 November 2021
Available online 21 November 2021

Conversely, experts may notice that the prediction cannot be trusted if the model made that decision based on information that should not be relevant to determine the NOC. By delivering such explanations, more informed decisions can be made.

Kruijver et al. presented a decision tree as a more transparent way to use machine learning to estimate the NOC [23]. 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 of Kruijver at al. also relies heavily on filtering of artefacts, for which another decision tree is used. Furthermore, the data used in this study are 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 are used, the performance of a simple model may decrease even further.

None of the previously mentioned techniques have explored the field of explainable artificial intelligence (XAI). XAI has emerged to provide explanations for machine learning models, since users want to know *why* a certain prediction is made [25–32]. The European Commission recently underlined the importance of explainability in a proposal for rules on AI systems in higher-risk settings such as law [33]. 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 from the perspective of NOC estimation. For a more in-depth overview of the field of XAI, we recommend surveys by Carvalho et al. [27] and Barredo Arrieta et al. [28].

1.2. Explainable artificial intelligence (XAI)

We introduce the main XAI terminology through the lens of applying machine learning to NOC estimation. The decision tree from Kruijver et al. is simple enough to walk through each node of the tree where decision conditions are assessed, until an output leaf node is reached [23]. This type of explanation is denoted as being *model-specific*, as it uses the structure of the decision tree model to explain the predictions made [27,28,31,32]. Some XAI techniques can instead work with only the training data and the predictions made on the training data, showing which parts of the input have influenced the output. This is called *model-agnostic*, which means that it works on all types of machine learning models, without relying on the (complex) mechanisms of how those models reach their decisions. Model-agnostic explanations can be used to describe how predictions are reached by the model on average, by using *global* methods [25,27–32,34]. However, DNA experts analyse one profile at a time and they would like the most accurate description of the prediction. For this case, *local* explanations could be more suitable. Local means that a new explanation is generated per sample, instead of averaging how the model makes predictions in general. Local, model-agnostic explanations are generally one of two types; *feature importance* or *counterfactuals*.

Feature importance methods highlight which feature values of the input, e.g. MAC or TAC values of the DNA profile, were most influential for a particular model prediction [25,27–29,31,34]. This effectively answers the question *"Why did the model predict A?"*. An established method for arriving at such explanations is SHAP (Shapley Additive Explanations) [35]. SHAP values indicate how the feature values of the input have caused the model to make a certain prediction, in comparison to the average prediction. For example, the average prediction of a regression model that predicts the NOC, where the output can take any value between 1 and 5 contributors, is about 3 contributors. This average is also referred to as the *base* prediction. From this base value, feature values are assigned positive SHAP values if they made the prediction higher, and negative SHAP values if they made the prediction lower. For example, a SHAP value of 0.2 for a feature indicates that the predicted NOC is 0.2 higher due to that feature value. Adding all positive and negative SHAP values together with the base value, yields that profile's prediction.

Counterfactual explanations are example data points which have a different prediction from the input data point [25,27,28,31,36–51]. By highlighting the differences between these instances, the user could derive how the original instance could have been predicted differently if certain input features had different values. As such, counterfactuals answer the question *"Why did the model not predict B?"*. In this paper, we present a counterfactual method that is suitable for practical problems that show strong correlations between the features, such as in NOC prediction.

The alternative outcome of the counterfactual is 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. This 'similarity' is most commonly measured by difference between the input and the counterfactual [37–43,45–48,52]. Most methods use L_1 or Manhattan distance, which corresponds to the sum of the absolute differences in the feature values [37,38,40–42,46,48]. Alternatively, the 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].

In summary, we regard a counterfactual useful if it is:

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

Counterfactuals can either be created by choosing instances from the training data [45,53], or by artificially sampling them [37–42,46,47,49–51]. The main advantage of presenting a training data point, is that it is a real-life example. Training data can for instance consist of real person's DNA profiles, which are realistic data points. One study demonstrates how to create counterfactuals from the training data, though they rely on the assumption that there are inherently sparse counterfactuals in the training set (with less than 2 feature differences) [45]. As the authors point out, this method will most likely fail on more real-life datasets which can be thinly populated. The most similar counterfactual might still be widely different from the input that you are comparing to. Sampling-based approaches usually do not suffer from this problem as they simulate an abundance of new data in close range to the input [37–42,46,47,49–51]. The issue with such sampling-based methods is realism. As the samples are often generated by randomly changing feature values, or by combining instances, they might be infeasible. For example, consider a machine learning model that decides whether a person is eligible for a loan. A generated instance in this context might be a 20-year-old person with 15 years of working experience as an ideal candidate for a loan. Such examples are of no value to the user as they do not represent real-life situations. A counterfactual example must be a plausible data point to be valuable to the user.

There have been some attempts to create more plausible counterfactuals. These often assume that features are independent. For example, the distance to the training data can be measured to see if the found counterfactual is close to the realistic training data [37]. Similarly, the sampled feature values can be limited to user-defined ranges, or ranges derived from the training data [39,41]. The issue with our previous 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. However, these methods require the user to model the relationships between all features [40,42]. 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, while presenting sparser counterfactuals than those taken from

the training data.

Finally, presenting counterfactuals in a visual way is regarded as helpful to 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], or in natural language [48,57,58]. This does not clearly communicate the magnitude of the feature value differences between these instances. With a visual approach, communication of these magnitudes can be improved [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].

1.3. 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 NOC to a DNA profile, which can be applied to any type of machine learning model. To that end we used a slightly modified version of an existing machine learning model that we trained on an expanded dataset. The original model was developed by van der Linden and coworkers [21]. The 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, yet finds 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

We used a dataset from a previous study by Benschop et al., consisting of 590 PowerPlex® Fusion 6C (PPF6C) profiles, either from a single donor, or from a mixture of up to 5 donors [21]. 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 regarded representative of real casework. The ground-truth NOC was therefore available. Each profile x was characterized by 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 simulated 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. DNAStatistX implements an algorithm to calculate the Maximum Likelihood Estimate which is largely based on the source code of the probabilistic genotyping system EuroForMix [15]. 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. Since the generated profiles were simulated before the features were extracted, a well-fitting probability model could be used to generate as plausible as possible profiles. As the used features are strongly correlated (see Supplementary Figure 1), and not as well studied as the DNA profiles, it would make sampling in a later step more difficult [68,69].

Simulating DNA profiles that are generated using capillary electrophoresis techniques is challenging and require a careful examination. To ensure that the simulated DNA profiles appropriately reflect the ground truth samples, we compared the features that were extracted from the simulated 5000 data points to those from the original dataset of 590 instances. In Supplementary Figure 2 and 3, we observed some discrepancies in variation between the two datasets. With further analysis, about half of the newly generated features appeared to have been drawn from a different distribution as compared to the original dataset (see Supplementary Figure 4 and 5, 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 or perhaps more drop-in. 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 performs better once trained on the combined dataset of the original 590, and simulated 5000 samples together (see Supplementary Table 4 and Supplementary Figure 8 and 9, in comparison to Supplementary Table 5 and Supplementary Figure 6 and 7).

2.2. Machine learning model

The machine learning model for NOC estimation used in this study is described in [21]. In that previous study, the prediction of the NOC was treated as a classification problem, $f(x) = y$, where x is an input profile consisting of the 19 features, y represents the possible NOC output $y \in \mathbb{N}1 \leq y \leq 5$, and f is a random forest classifier (denoted RFC19) [21].

Since the NOC outputs of the model are ordinal, meaning that there is an inherent order between the output categories, the problem could benefit from being tackled with a regression model. In this way, we can penalize a wrong prediction more if it is further away from the ground-truth; a prediction of an NOC of 4 is more wrong than an NOC of 3 when the profile has an NOC of 2. After a short benchmarking study with a default random forest regressor (see Supplementary Figure 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 ones.

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 $y \in \mathbb{R}1 \leq y \leq 5$. This value can give an impression of certainty of the prediction, as a value close to an integer (e.g., 3.1) indicates a more confident result than a value that lies on a threshold between two integers (e.g., 3.4).

In this study, we used the regression model trained on the expanded dataset (denoted RFR19_merged) as we find it most intuitive for this problem. Note that the XAI method will be applicable independent of the type of machine learning model, though we recommend regression for ease of interpretation.

2.3. Explanation goals

The explanation should answer two main questions [25,38,39]:

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?*

These questions fit with the scenarios that DNA experts are faced with when analyzing the output of the machine learning model, and deciding on the NOC. The first question can best be answered by an explanation of what features of the profile contribute most to the prediction [62], for which we deem SHAP suitable. For the second question, a counterfactual explanation demonstrating what changes in this profile would lead to a different prediction is best fit.

2.4. Requirements counterfactual explanations

To develop the most suitable counterfactual method, we derived a list of requirements that it must accommodate (see Table 1 and Sections 1.2 and 2.3).

Given possible future improvements to machine learning modeling, a *model-agnostic* method is preferred. In this way, the same explanations can be generated regardless of the underlying algorithm.

Most existing counterfactual methods assume a binary classification problem, meaning that there are only two possible outcomes. The counterfactual target then automatically becomes the outcome that was not reached in the current prediction. In our problem however, the range of possible outcomes is 1–5. In an example where the NOC is predicted to be 2, certain users might be interested in a counterfactual target NOC of 1, while others prefer 3 or 4. 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 most similar counterfactual example is very different from the input profile, that shows a limitation of using the dataset in this way. 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 and 5 to match directly with the NOC that DNA experts must report. Since the model in this study uses regression, we consider instances with a rounded-off prediction that match the target to be *valid* counterfactuals.

We would like the input- and counterfactual profiles to differ in a small number of their features. This *sparsity* is encouraged to prevent users from experiencing cognitive overload. We also know that humans pick explanations in a biased way, meaning that if many options are available, only a few will be selected and the rest ignored [26]. This would not paint an accurate picture of the differences between the profiles; therefore, a small number is regarded better.

For the distance between the input and counterfactual, we first analyzed the underlying data. The distance function used should be catered towards the problem [38]. As our dataset has outliers, and most features are not normally distributed (see Supplementary Figure 4 and 5), L_1 distance is most appropriate. We also scale with each feature's range to minimize the influence of features with different ranges, variations, and distributions [37,39]. The distance measure is shown in Equation 1.

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

Where $\widehat{R}_i$ represents the range of the i-th feature, x the profile to be explained, and $x^{'}$ the counterfactual profile. We divide by n, the number of features ($n = 19$ in this study), such that the score lies between 0 and 1, independent of how many features a profile consists of in the data. This score can also be used for categorical variables by replacing $\frac{1}{\widehat{R}_i}|x_i - x^{'}{}_i|$ with $\mathbb{I}[x_i \neq x^{'}{}_i]$.

Table 1
Requirements for the counterfactual explanations with their definitions.

Requirement	Definition
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	Has minimal number of features differing between input and counterfactual
Proximal	Has minimal distance between input and counterfactual
Realistic	Has plausible combinations of feature values in counterfactual

The requirements discussed so far have been well-covered 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 points that are plausible. None of the sampling methods discussed in Section 1.2, 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. Instead, we utilize the training data which inherently consist of realistic instances.

2.5. Realistic Counterfactuals (ReCo)

To fulfil all previously defined requirements, 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.

2.5.1. 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 $f(x^*) = y'$.

ReCo then finds the optimal set of instances with regards to sparsity and proximity using multi-objective optimization. By minimizing both objectives simultaneously, we find non-dominated instances. These are depicted by the filled green crosses in Fig. 1. What this entails is that for each of these instances, there exists no better alternative; there cannot be an improvement for one objective, without getting a worse score for another objective [37]. As we intend to present a single counterfactual, we select the median instance x' from this set which balances the two scores most equally as highlighted by the circled data point in Fig. 1.

The advantage of this approach is that 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

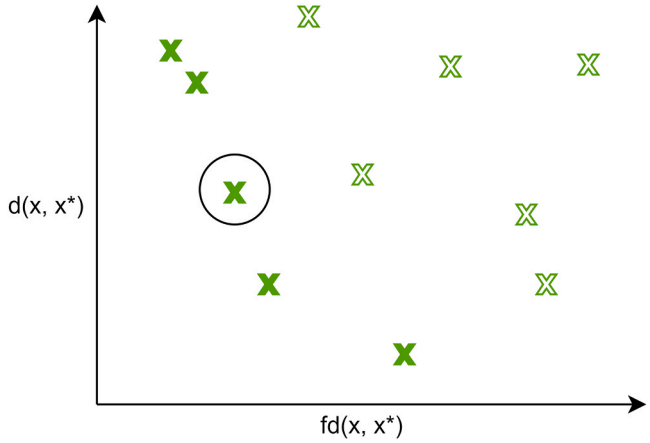


Fig. 1. The filled green crosses represent data points that are non-dominated; they have good scores for both $d(x, x^*)$ as defined in Eq. 1, and $fd(x, x^*)$ which represents the number of feature differences between profiles x and x^*. For both metrics, lower is better. The circled instance will be chosen as it is the median of the set of non-dominated instances.

can also be compared without any normalization.

The counterfactual instance x' is part of the training data, making it a realistic data point to present. However, such an instance likely has many different feature values as compared to the input profile; it is not sparse. Moreover, not all of these differences are relevant to arrive at the target prediction as it is simply an example. ReCo tackles both issues by applying a filter to the found counterfactual instance, selecting only the most relevant differences from the input.

2.5.2. Filtering the counterfactual training instance

Filtering is done using the following five steps. Table 2 shows an illustrative 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 2, there are three features in this set.

$$\text{differences} = \{\forall i \in \mathbb{N} \wedge 1 \leq i \leq n | x_i \neq x'_i\} \quad (3)$$

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 2, the SHAP change of Feature 1 is largest, while it is the smallest for Feature 3.

$$\text{SHAP\_change} = \{\text{SHAP}(x'_i) - \text{SHAP}(x_i) | \forall i \in \text{differences}\} \quad (4)$$

3. To make the counterfactual instance sparser in its differences 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 2; 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 for the counterfactual prediction, and could therefore possibly be filtered from the counterfactual instance.

$$\text{misaligned} = \begin{cases} \{\forall i \mid \text{SHAP\_change}_i > -\epsilon\}, & \text{if } f(x') - f(x) < 0 \\ \{\forall i \mid \text{SHAP\_change}_i < \epsilon\}, & \text{otherwise} \end{cases} \quad (5)$$

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. We will denote x^i as the counterfactual with feature i filtered. If the prediction of this filtered counterfactual $f(x^i)$ stays the same as the target y', it is labelled as irrelevant_diff.

$$\text{irrelevant\_diff} = \{\forall i \in \text{misaligned} | f(x^i) = y'\} \quad (6)$$

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

$$\text{counterfactual} = x^i | \forall i \in \text{irrelevant\_diff} \quad (7)$$

Table 2
Example of how a counterfactual is filtered. The input instance has a prediction of 4 contributors, and the counterfactual has a prediction of 3 contributors; 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.

	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

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; the number of times we ran through steps a and b.

Please refer to Fig. 2 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 whether 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 input) 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.

2.7. Set-up quantitative evaluation ReCo

To determine the quality of ReCo, we have assessed it on the six metrics defined by the requirements 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. Proximity to the training data is often used as a score of realism, measuring the distance between the counterfactual and the closest training instance. However, we argue that our realism metric defined in Section 2.6 reflects this purpose better. We will

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

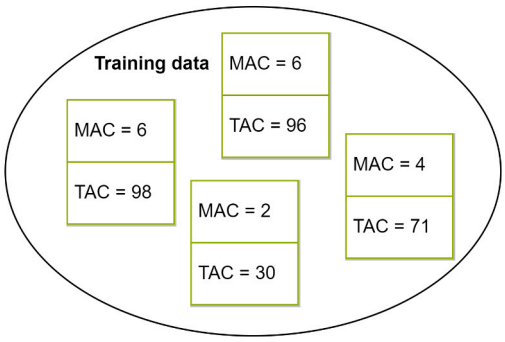


Fig. 2. Example of a counterfactual that receives a realism score of 0; the proposed counterfactual contains a feature combination that could not have resulted from a DNA profile.

present both for comparison.

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]. *DiCE random* is a sampling approach that generates counterfactuals from the input by randomly sampling different feature values until the target prediction is reached [40]. 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* for which we also used the default parameter values. Its implementation is inspired by GeCo [42], as it generates counterfactuals 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.

For all these methods, the target is set to the second most likely prediction.

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. The following requirements were considered from conferring with the consulted DNA experts in addition to some requirements already expressed in the literature [63,64].

First, the visualization is *consistent*. Each profile is presented in the same format to help users reach some level of familiarity with the visualization over time. The feature values are plotted on a normalized scale to get a visual representation of how large a value is compared to the range of possible values. 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. Thirdly, the used color palettes are specifically chosen to be *accessible* as they are distinguishable to the color-blind [65]. Lastly, some *interactivity* is introduced by enabling the user to choose the counterfactual target.

2.9. Set-up user study

It was important to evaluate the explanation 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 who use a NOC machine learning model in casework at NFI. In total, eight experts participated.

The evaluation was set up around two 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. Examining whether users can more accurately determine the NOC was out of scope for this study 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 are also not fully understandable to users as many of the features are too abstract or unclear in how they relate to NOC estimation.

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 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 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 asked these questions when users were only presented with the prediction. Then we asked them once again after a state-of-the-art explanation, and once after our visualization. If users had more trust in the prediction after seeing the explanation, we expected them to be able to pinpoint the NOC more (choose less options), and believe the prediction is (more) correct. For profile 1, we compared our visualization against a SHAP force plot [66]. 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 or irrelevant. Since the scale of the survey was small, we allowed users to motivate their answers.

Before these questions were asked, all types of explanations (SHAP force plot, counterfactual table, our visualization) were introduced with a video, figure and bullet points to ensure that the participants understood the presented information in each explanation. The most important features and their interpretation were presented as well. 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 quantitative evaluation of ReCo, after which we present the visualization and the corresponding user study results.

3.1. Quantitative evaluation ReCo

The obtained scores on the test data for the four methods can be found in Fig. 3.

The WhatIf method could be seen as a baseline, using only existing training examples as counterfactuals. The 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 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 is quite high, and the distance to the training data is practically zero. We can attribute these better scores to the fact that this approach combines existing profiles from the training data. However, this crossover step still mixes the feature values of two instances into one, which can create unlikely feature combinations. The mutation step 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 scores 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 counterfactuals are already sparser. Then because of ReCo's filtering step, which removes any irrelevant differences, these two scores can go down further. 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 several factors. First, 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 the current realism metric is strict; it

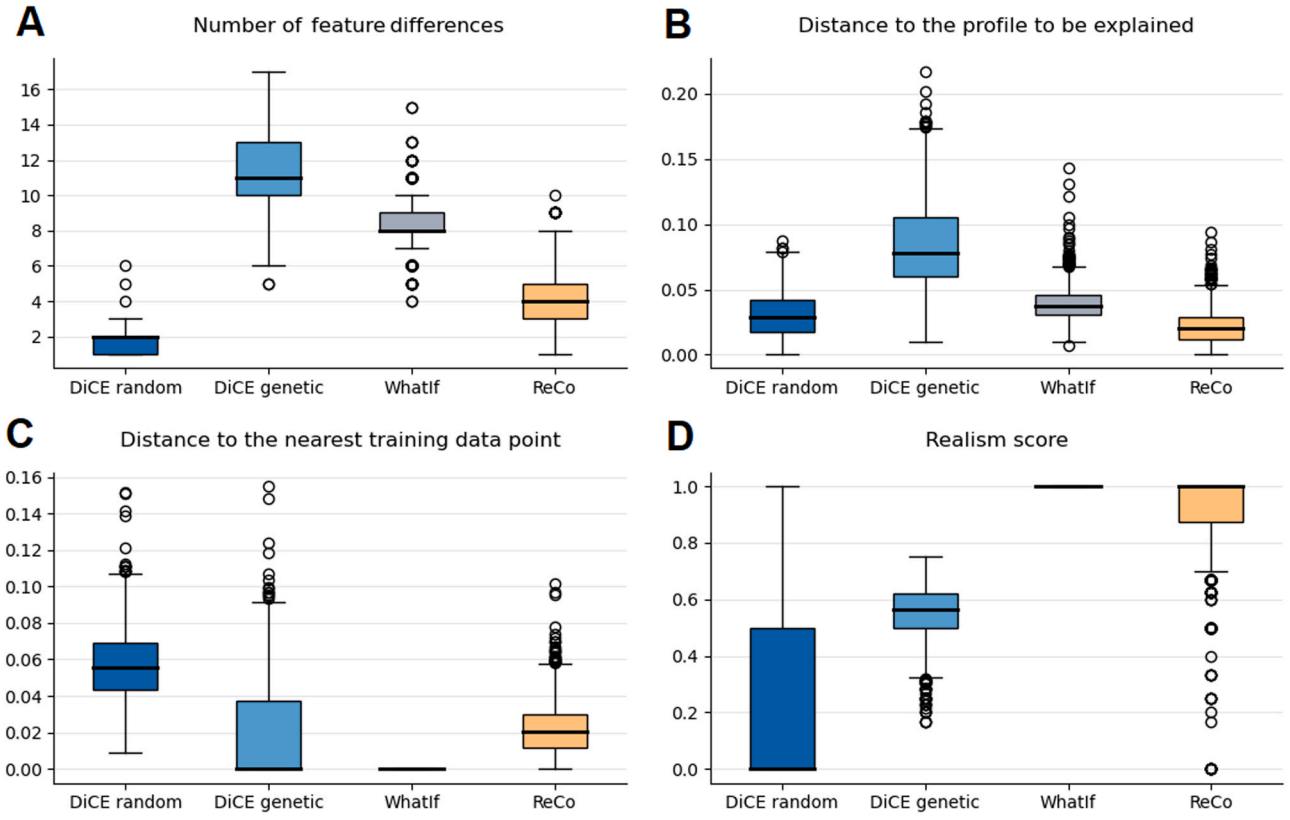


Fig. 3. Quantitative evaluation of ReCo in comparison to WhatIf, DiCE random and DiCE genetic on four different metrics; sparsity measured by the number of feature differences (A – lower is better), proximity measured by the distance to the input (B – lower is better), the distance to the training data (C – lower is better), and realism (D – higher is better).

does not check if a feature value is close to known combinations in the training data, the values must match 100%. In theory, it would be more appropriate to compute the total probability density of the counterfactual feature values. As this is infeasible, the current realism score approximates this idea.

Although the current implementation of ReCo is used for regression, it can be used for classification as well. In this case, we do not analyze if SHAP values are aligned with the *direction* of the change in prediction, but we determine if changes in feature values correspond to *more positive* SHAP values for the target class. However, we argue that regression is more fit for the domain of NOC estimation as the SHAP values are more informative. For example, a negative SHAP value in a regression context means that the corresponding feature value fits with a lower NOC, whereas within the context of classification it means that the feature value does not fit the predicted class. With the latter, it is unclear if that value fits better with a higher or lower number of contributors.

3.2. Visualization

The visualization for the explanation of a single DNA profile prediction is depicted in Fig. 4.

The top line informs the user about the current profile and what the model's prediction is. We show the raw and rounded prediction to give the user an impression about the certainty of the prediction. 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. Starting from the base 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. For this prediction, the model observed this profile's higher values of *MAC*, *loci with 5 or 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 locus indeed imply more donors, and lots of low peaks indicate that more donors may have contributed than expected based on the number of detected alleles. For example, alleles might have dropped out as a result of low quality or quantity of the DNA profile. In contrast, the TAC and peak height variation at locus vWA have low values that typically occur in lower-order mixtures.

To generate a counterfactual explanation for this profile, we have set the target at two contributors. As often a minimum NOC is reported it might be interesting to be able to rule out a NOC of two, and instead go with the current prediction of three contributors. Within the application, the user can first explore the factual explanation consisting of the features and SHAP values before choosing their desired 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, number of loci with 5 or 6 alleles, and number of peaks below the stochastic threshold of 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 two 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 two.

Note that the explanation gives information about the model, not about the underlying data. This means that even though the model might be certain about a prediction, it can still be incorrect. The user should be aware that the explanations only give insight into how the model makes decisions.

3.3. User study results

In total, eight answers to the survey described in Section 2.9 were collected from DNA experts. One of these responses had to be eliminated as they failed the qualification tests. Because of the limited size of the

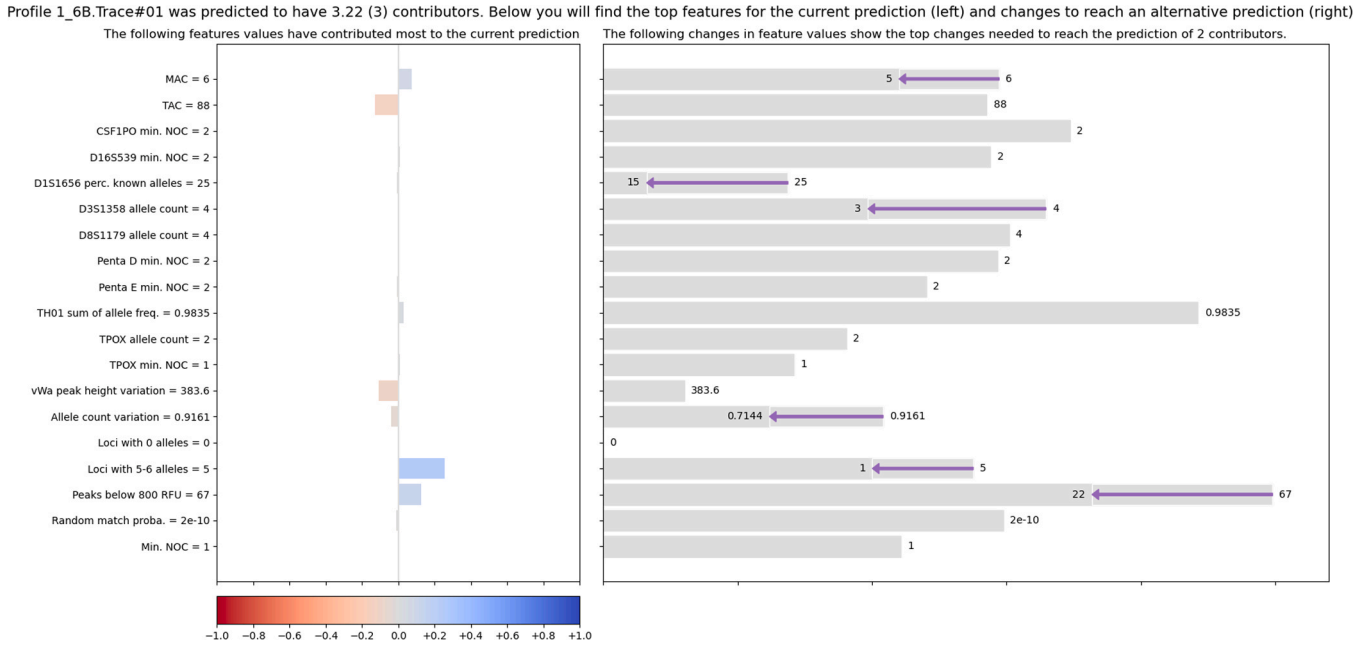


Fig. 4. Visualization for the explanation of a profile with 3 contributors, that was correctly predicted to have 3 contributors (profile 1 in the user study). Its feature values are listed on the left and plotted on the right. SHAP values are depicted on the left with red and blue bars, and a counterfactual example generated by ReCo for a prediction of 2 contributors is shown on the right with arrows.

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 Fig. 5. It shows whether users gained or lost trust in the prediction after seeing the two explanations for profile 1 and 2, in comparison to seeing no explanation. For profile 1, the intended goal was to increase users' trust in the prediction, while for profile 2, we hoped to decrease their trust. Though the results in Fig. 5 show only minor discrepancies between each comparison of our visualization against either the SHAP force plot and CF table, we could derive more nuance from the participants' motivations.

When the model is fairly certain about the prediction (profile 1, presented in Fig. 4), seeing any explanation makes some users (3/7) both gain more trust in the correctness of the prediction, and in pinpointing a certain NOC. 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 one 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 two contributors, and therefore dropped this outcome from consideration. One user thought that the visualization presented similar information to reach a prediction of two contributors as they would have thought, thereby increasing their trust in this explanation.

When the model is uncertain or incorrect (profile 2, presented in Fig. 6), the counterfactual 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. Note that this is in comparison to only seeing the prediction without any explanation, so the fact that this prediction is close to the boundary of three and four contributors is already considered. From the additional textual input, the majority of the 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 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.

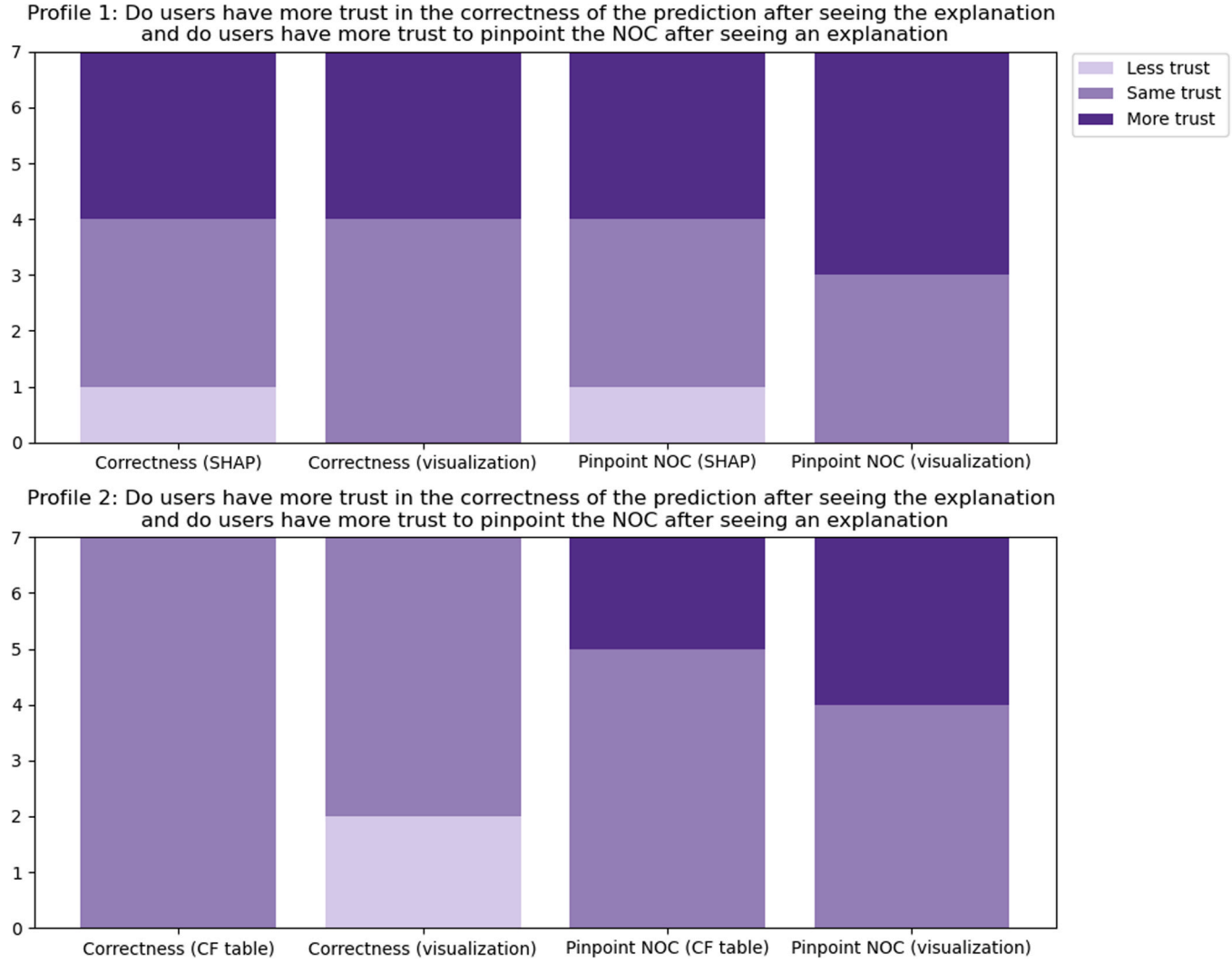


Fig. 5. Results from the user study trust exercise. For profile 1 (top), it shows the influence of seeing a SHAP force plot explanation in comparison to our visualization, on trust in the correctness of the model prediction, and on the users trust to pinpoint the NOC. For profile 2 (bottom), it shows the influence of seeing a counterfactual (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. Less, same and more trust are in relation to the answers given after only seeing the prediction without any explanation.

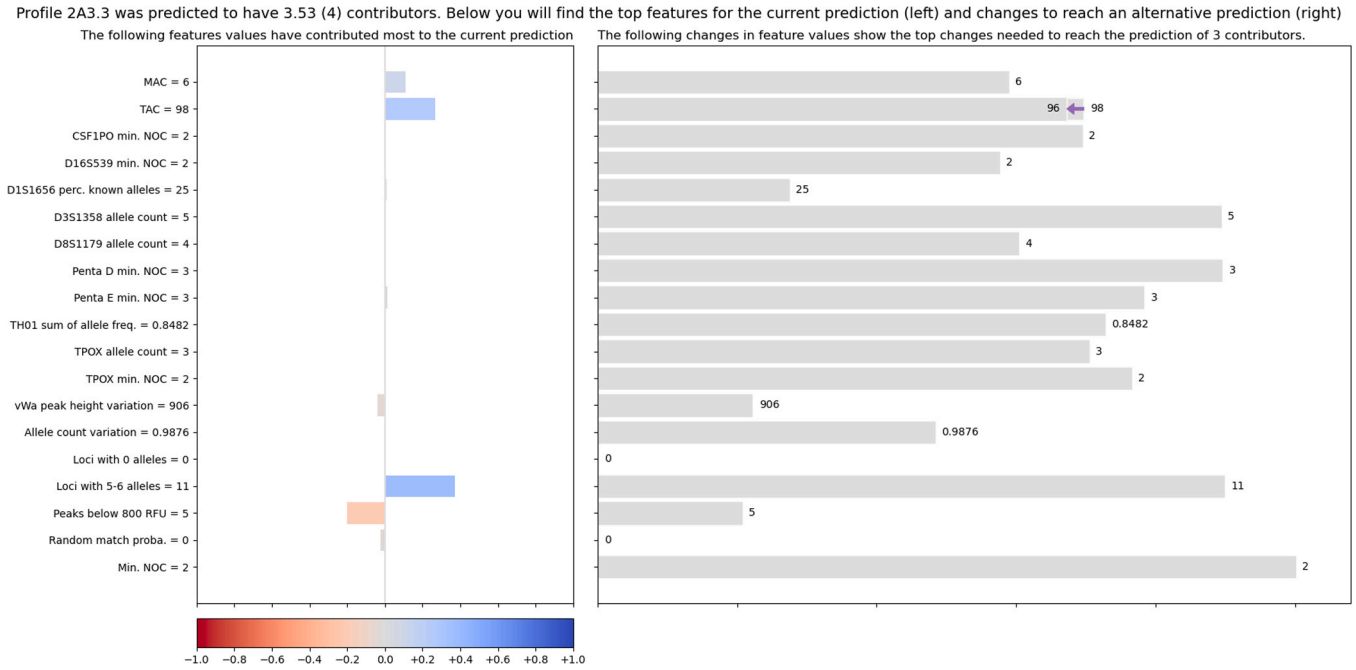


Fig. 6. Visualization for the explanation of a profile with 3 contributors, that was incorrectly predicted to have 4 contributors (profile 2 in the user study). Its feature values are listed on the left and plotted on the right. SHAP values are depicted on the left with red and blue bars, and a counterfactual example generated by ReCo for a prediction of 3 contributors is shown on the right with arrows.

The visualization in general made users more confident to pinpoint the NOC as they considered less options than with the SHAP or counterfactual table explanations (more trust to pinpoint NOC in Fig. 5).

In short, it seems that our visualisation (Figs. 4 and 6) 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, or revisited with a larger group of people. We expect that the users' impression might change once they gain more experience with these explanations.

As a final remark, we want to re-iterate that the current features with which the DNA profiles are presented, are not fully understandable to the users. We aided the experts during this user study by providing simplified explanations for groups of features, but ultimately the features should be intuitive on their own.

The results of the second task about user preferences can be found in Table 3. 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.

3.4. Future work

In this study, an existing machine learning model was used [21]. XAI was applied to this machine learning model which we slightly modified and trained on an expanded dataset. As the DNA experts we consulted have indicated, the features on which the explanation are based are still difficult to comprehend. It also seems that the features of the machine learning model that was used 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. With any machine learning model, it is advised that the features are made comprehendible to the users prior to implementing the explanations in forensic DNA casework, as this will further enhance understanding the model's predictions.

Though the explanations work for both classification and regression, we recommend using regression for NOC estimation as the interpretation of the SHAP values becomes more informative. 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 [67].

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.

Table 3
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.

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

4. Conclusion

This study describes the research and design of an application of XAI for predictions of the number of contributors of DNA profiles which can be used for any type of machine learning model. The explanation consists of SHAP values and a counterfactual example incorporated into a compound visualization, which we believe is the first visualization to unify these two components. With a user study, the explanation has been evaluated by a small group of forensic DNA experts. From their

observations, the visualization seems to provide 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 counterfactuals that have fewer feature differences than by using training examples, and are more realistic than counterfactuals generated by sampling-based approaches. To the best of our knowledge, ReCo is the first method that can handle correlated data automatically, but still creates 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.

The code is available at github.com/martheveldhuis/ReCo.

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, forensic DNA experts from the NFI for participating in the user studies, and Jason van Breukelen for providing feedback on the visualization.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.fsigen.2021.102632.

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 Sci. Int.: Genet. 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 Sci. Int.: Genet. 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? J. Forensic Sci. 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 Sci. Int.: Genet. 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 Trans. Comput. Biol. 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 Sci. Int.: Genet. 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 Sci. Int.: Genet. 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 Sci. Int.: Genet. 16 (2015) 172–180.
[9] C. Benschop, A. Backx, T. Sijen, Automated estimation of the number of contributors in autosomal STR profiles, Forensic Sci. Int.: Genet. Suppl. Ser. 7 (2019).
[10] J. Pérez, A. Mitchell, N. Ducasse, J. Tamariz, T. Caragine, Estimating the number of contributors to two-, three-, and four-person mixtures containing DNA in high template and low template amounts, Croat. Med. J. 52 (2011) 314–326.
[11] S. Norsworthy, D.S. Lun, C.M. Grgicak, Determining the number of contributors to DNA mixtures in the low-template regime: exploring the impacts of sampling and detection effects, Leg. Med. 32 (2018) 1–8.
[12] J. Buckleton, J. Curran, P. Gill, Towards understanding the effect of uncertainty in the number of contributors to DNA stains, Forensic Sci. Int Genet 1 (1) (2007) 20–28.
[13] M.D. Coble, J.-A. Bright, Probabilistic genotyping software: an overview, Forensic Sci. Int.: Genet. 38 (2019) 219–224.
[14] D. Taylor, J.-A. Bright, J. Buckleton, Interpreting forensic DNA profiling evidence without specifying the number of contributors, Forensic Sci. Int.: Genet. 13 (2014) 269–280.
[15] 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 Sci. Int.: Genet. 21 (2016) 35–44.
[16] 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 Sci. Int.: Genet. 49 (2020), 102390.
[17] 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 Sci. Int.: Genet. 42 (2019) 31–38.
[18] 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 Sci. Int.: Genet. 40 (2019) 150–159.
[19] 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 Sci. Int.: Genet. 38 (2019) 225–231.
[20] T.M. Clayton, J.P. Whitaker, R. Sparkes, P. Gill, Analysis and interpretation of mixed forensic stains using DNA STR profiling, Forensic Sci. Int. 91 (1) (1998) 55–70.
[21] 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 Sci. Int.: Genet. 43 (2019), 102150.
[22] M.A. Marciano, J.D. Adelman, Developmental validation of PACETM: automated artifact identification and contributor estimation for use with GlobalFilerTM and PowerPlex® fusion 6c generated data, Forensic Sci. Int.: Genet. 43 (2019).
[23] 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 Sci. Int.: Genet. 50 (2021), 102407.
[24] M.A. Marciano, J.D. Adelman, PACE: probabilistic assessment for contributor estimation— a machine learning-based assessment of the number of contributors in DNA mixtures, Forensic Sci. Int.: Genet. 27 (2017) 82–91.
[25] B. Mittelstadt, C. Russell, S. Wachter, Explaining Explanations in AI, 2018.
[26] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artif. Intell. 267 (2019) 1–38.
[27] D.V. Carvalho, E.M. Pereira, J.S. Cardoso, Machine learning interpretability: a survey on methods and metrics, Electronics (Switz.) 8 (8) (2019).
[28] 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, Inf. Fusion 58 (2020) 82–115.
[29] Z.C. Lipton, The mythos of model interpretability: in machine learning, the concept of interpretability is both important and slippery, Queue 16 (3) (2018).
[30] 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.
[31] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (XAI), IEEE Access 6 (2018) 52138–52160.
[32] M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, Commun. ACM 63 (1) (2020) 68–77.
[33] E. Commision, Fostering a European approach to Artificial Intelligence, 2021.
[34] W.J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, Proc. Natl. Acad. Sci. U. S. A. 116 (44) (2019) 22071–22080.
[35] S. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, 2017.
[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, Adv. Data Anal. Classif. 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, Harv. J. Law Technol. 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 Intell. Syst. 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 Trans. Vis. Comput. Graph. 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. 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 - Kunstl. Intell. 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] 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.
[64] K. Sokol, P. Flach, Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety, 2019.
[65] P. Kovesi, Good Colour Maps: How to Design Them, ArXiv abs/1509.03700 (2015).
[66] 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, Nat. Biomed. Eng. 2 (10) (2018) 749–760.
[67] 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.
[68] M.G. Kendall, A new measure of rank correlation, Biometrika 30 (1–2) (1938) 81–93.
[69] 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).