**Realistic visual explanations for predictions of the number of contributors made by any machine learning model**

**Bringing XAI to predictions of the number of contributors: visual explanations**

**Abstract**

Using machine learning to determine the number of contributors (NOC) in short tandem repeat (STR) profiles has been shown to obtain good accuracy. However, these predictions may not be fully understandable to the biologist user who is presented with these as a tool to help determine the number of donors. Therefore, we introduce the field of eXplainable artificial intelligence (XAI) to this problem, and to the audience in general. We apply XAI to NOC estimation by using a visual aid that incorporates explanations from SHAP values and counterfactual examples for each prediction. Existing methods for generating counterfactuals have not attempted to handle correlated features, causing those methods to find examples that are impossible given their feature combinations. Since the features from STR data are highly correlated, we have implemented a new method, ReCo, that generates realistic counterfactuals on highly correlated data. We show that ReCo outperforms state-of-the-art methods, with traditional metrics, as well as a novel realism metric. A final user evaluation demonstrates the opinions of end-users, a metric that we regard important in XAI studies.

1. **Introduction**
   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 software that is used for DNA interpretation does require the NOC to be entered by the user [10, 11], which influences the calculation of the evidence [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 [18, 19]. However, machine learning algorithms are often considered to be “black-boxes” [20-26], as the predictions they output 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.

A method using a decision tree was presented as a more transparent way to use machine learning [27]. However, using a simple model such as a decision tree leads to less accurate predictions; they reported a difference of over 10% as compared to a random forest. This method also relies heavily on filtering of artefacts, for which another decision tree is used. 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. But how can we explain predictions by more complicated models? The field of eXplainable Artificial Intelligence (XAI) has become increasingly important for any application of machine learning models. We aim to provide some basic insight into this discipline before diving into how it can be used in this domain.

* 1. *eXplainable Artificial Intelligence*

One of the main flaws of more complex (deep) machine learning algorithms is their lack of transparency [20-26]. They are often described as “black-boxes” as humans cannot easily discern exactly how the model functions and why it comes to its conclusions. From this demand for explanations, the field of eXplainable Artificial Intelligence (XAI) emerged. The European Commission recently underlined the importance of explainability in a proposal for rules on AI systems in higher-risk settings such as law [28]. Though NOC estimation does not directly make decisions without involvement of human experts, these experts should be well-informed about the AI system that they might let influence their decision.

Machine learning models roughly fall into two categories when it comes to how interpretable they are; transparent- and black-box models. With transparent models, one can derive the exact steps taken to arrive from input features to an output within reasonable time [23, 24]. 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 [29]. If all of these conditions are violated, a 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. 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) [20, 22-26, 30, 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.

There exist roughly two directions of generating local, model-agnostic explanations. The first concerns feature importance methods, 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?”*. SHAP values have a solid background in game theory **<TODO: source>**

Some research has implemented SHAP to real-life cases such as predicting hypoxia based on clinical data [32], and predicting the most fitting eye-surgery type [33]. They seem to have obtained valuable information for what are important factors to ML models. The second direction of explanations is more recent, 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 [21]. Since this field is new, numerous methods are being developed, yet none has been proven suitable for many real-life applications.

* 1. *Counterfactuals*

A counterfactual is an example instance that is similar to the instance we want to explain, but has a different prediction. The differences in feature values between the input- and counterfactual instances can give the user an impression about the local decision space of the model. More formally, a counterfactual can be described as follows [34]:

“*The model predicted outcome because input instance had values . If instead instance had values , and all other values had remained constant, the model would have predicted outcome ”*

This alternative outcome is often referred to as the target of the counterfactual [34-49]. To help the user relate this new prediction as a possibility for the original input, the counterfactual- and input instances must be similar. 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 [34, 36-41, 43-46, 50]. Though some methods use or Euclidean distance [41, 45], or Manhattan distance appears to be the measure of choice as it does not blow outlier distances out of proportion as distance tends to do [34, 36, 38-40, 44, 46]. 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 is measured by the number of differences in feature values in comparison to the input [35, 36, 38, 40, 42-46, 48, 49], sometimes referred to as distance.

In summary, 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 [36-40, 46], or providing counterfactuals that are actionable; meaning that the changes can be acted upon to reach that alternative outcome [36-38, 41, 42, 49]. This is useful when the input features can be changed in the future, for example by raising your income for a loan application.

Counterfactuals can either be chosen from the training data [43, 51], or can be artificially sampled [34, 36-40, 44, 45, 47-49]. 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 quite 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 [34, 37, 39, 44], or take the input and perturb its feature values until a different outcome is reached [36, 38, 40, 45, 47-49]. While most tackle this by randomly changing feature values [34, 37, 38, 44, 45, 47, 49], some take a more sophisticated route by using a genetic algorithm [36, 39, 40, 48]. 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.

Some approaches have leveraged the power of SHAP values to create their counterfactuals [35, 47]. By only changing the features from the input instance that have negative SHAP values for target class B, a counterfactual could be found [47]. 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. Similarly, by iteratively setting the features with the highest SHAP values for the predicted class A to zero, a the target class could also be reached [35].

An aspect of generating counterfactuals with sampling-based methods that is largely overlooked or handled quite poorly is realism. As these samples are often generated by randomly changing feature values, or by combining instances, they might be quite 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. That means they would have started their career at age 5. 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 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 [36]. 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 [41]. This ensures that the query instance can be transformed to take on another the target output, which is relevant for actionable settings. Similar to these approaches, the range of feature values can be limited [37, 39]. Either based on the training data, or inputted by the user. When considering our 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. Neither 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 [38]. 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. A Julia implementation 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 [40].

One method derives counterfactuals from training instances, which relies on the assumption that there are inherently sparse counterfactuals in the training set [43]. 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 [37, 38, 52, 53], no method has 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.

* 1. *Contribution*

The contribution of this paper is as follows:

* Introduce the concept of eXplainable Artificial Intelligence (XAI) to the field of forensic science by demonstrating its value on a practical issue.
* Generate explanations for individual predictions of the Number of Contributors (NOC) to a DNA profile by any machine learning model.
* Present the explanations consisting of SHAP values and counterfactual examples in a visualization.
* Implement a new method for finding realistic counterfactuals (ReCo) by deriving them from the training data. This produces examples that have fewer feature differences than using training examples, but are still plausible data points. To the best of our knowledge, this is the first method that handles correlated features automatically.
* Create a new realism metric that scores how plausible counterfactuals are in terms of their feature combinations, which is important with highly correlated features.

With this paper, we intend to introduce the concept of eXplainable Artificial Intelligence (XAI) to the field of forensic science by demonstrating its value on a practical issue. We generate explanations for individual predictions of the Number of Contributors (NOC) to a DNA profile by any machine learning model. These explanations consist of SHAP values and counterfactual examples in a compound visualization. We implement a new method for finding realistic counterfactuals (ReCo) by deriving them from the training data. This produces examples that have fewer feature differences than using training examples, but are still plausible data points. To the best of our knowledge, this is the first method that handles correlated features automatically. Lastly, we have created a new realism metric that scores how plausible counterfactuals are in terms of their feature combinations, which is important with highly correlated features. The explanations are assessed both with objective metrics, as by a small group of users of the application.

1. **Materials and methods**
   1. *Data analysis and sampling*

The dataset initially consisted of 590 PowerPlex® Fusion 6C (PPF6C) profiles, either from a single donor, or a mixture up to 5 donors [27]. The NOC was based on ground-truth information. Each profile is represented by 19 input featuresconsisting of allele counts, allele frequencies and peak heights such that . These are all continuous variables.

These features are almost all very highly correlated.

The original dataset was expanded with 5000 more samples to create a less sparse feature space. In a development version of the statistical library DNAStatistX [13], realistic profiles can be generated by using the same model that is used for calculating evidence (EuroForMix model [12]). This program generates parameters such as peak height, degradation, and mixture proportions within ranges as defined by the original dataset. In Appendix x, the exact parameters can be found. Based on population frequency, genotypes are generated randomly using the EuroFixMix model, as well as adding sporadic drop-in alleles. To ensure that all donors are visible in the profile, each donor must have an LR of at least 1000.

* 1. *Machine Learning model*

Originally, the estimation of the NOC was treated as a classification problem, such that where is an input profile consisting of 19 features such as allele count, allele frequency and variation in peak heights. The model is a random forest classifier (titled RFC19), which produces an output within five categories such that [18]. However, since the outputs of the model are ordinal, the problem could benefit from being tackled with a regression model. After a short benchmarking study (see Appendix 1), we concluded that a regression model can achieve more accurate predictions. The model can then be defined to map the input profile to an output where .

In this study, we used a Random Forest Regressor with default parameters.

Since determining the NOC with machine learning is still a novel approach, there is no consensus about which type of model is most fit. The Netherlands Forensics Institute (NFI) is looking to improve the model and used features in the future.

* 1. *Desiderata explanations*

A good explanation of a prediction answers two main questions [20, 34, 37]:

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?*

In a short user study, these questions were confirmed as well.

To answer question 1, we determined that the use of SHAP values would be sufficient to give an impression of feature importance. For question 2, we derived a list of desiderata in order to define what the counterfactual explanations must accommodate.

* Model-agnostic Works for any model
* Interactive Target output can be chosen by user
* Valid Target output is reached
* Sparse Minimal feature differences
* Proximal Minimal distance in feature differences
* Realistic Plausible combinations of feature values

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

Most existing methods assume a binary case, and thus do not have to concern themselves with which target output to pick other than the opposite. In this problem, 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.

The notion of Multi-Objective Counterfactuals was first proposed with four objectives, solved by a genetic algorithm [36]. Besides distance between x and x’, and the number of feature differences, they also consider the distance to the target outcome, and plausibility based on distance to the training data.

The choice of distance should be catered towards the problem [34].

To measure the distance between two instances, we implemented an norm function as shown in Equation 1.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Where represents the range of the -th feature, the number of features, the profile to be explained, and the counterfactual profile. This distance measure scales every difference between the input- and counterfactual instance by that feature’s range.

Though much of the literature scales each distance by the Median Absolute Deviation (MAD) [34, 38, 43, 44, 46], this is not appropriate for the current dataset because not all features are normally distributed. If a feature with a value much larger 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 [36, 37]. This is quite robust even for unscaled and unnormalized features with lots of outliers, which is the case in this dataset. Another nice property is that . It can also be used alongside categorical variables by replacing with .

The second objective is the number of different feature values between the two instances, measured with norm.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

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; to provide a user with multiple routes to reach the different outcome [34, 40, 46]. Moreover, presenting multiple, possible contradicting examples does not seem like user friendly introduction to counterfactual explanations.

The above constraints have mostly been covered quite well in the literature, with the exception of realism. None of the methods discussed in section 1.x, are automatically suitable for datasets with correlated features, as they would produce unlikely feature combinations.

* 1. *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 trainings 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 and its prediction , where can be any machine learning model, the user defines a target prediction . ReCo then finds all instances from the training data with the target prediction . This prediction must match with their ground truth NOC so that no incorrect predictions are presented as examples.

ReCo then scores the candidate instances by a weighted sum of two objectives, combining the distance score defined in Eq.1, and the number of different features defined in Eq. 2 (Section 2.3). By optimizing for both objectives, the obtained counterfactuals have a better balance between the multiple scores than by only optimizing one [36]. Though multi-objective optimization can be solved by finding the set of non-dominated instances with different trade-offs between the different objectives [36], we intend to present a single counterfactual with the lowest combined score. This can be achieved most efficiently and simply by computing the weighed sum [54, 55]. ReCo selects the instance with the minimum score to be the counterfactual instance:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Where the weights and are defined a priori based on the medians of both and in the training data, ensuring that both scores are considered equally.

The counterfactual instance is part of 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:** This 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 and the counterfactual .
2. Compute the SHAP values for both the input- and the counterfactual instance, per feature in . Then subtract these SHAP values of the input instance from the SHAP values of the counterfactual instance. This set is sorted by the magnitude of each value. This gives us an impression of which changes in feature values from the input- 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.
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 instance, 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 with the change in prediction. These feature differences are most likely not relevant to help reach the counterfactual prediction, and could therefore possibly be filtered from the counterfactual instance.
4. The next step is to check if the feature differences with SHAP change can be removed. ‘Removing’ in this context means that the feature value of the counterfactual is replaced with the feature value of the input instance . If the prediction of this filtered counterfactual stays the same as the target , it is labelled as .
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:

Table 1: Example of how a counterfactual instance is filtered. The input instance has a prediction of 4, and the counterfactual instance has a prediction of 3. Therefore, the direction of the change in prediction is negative. The SHAP values of all three features are calculated for both the input- and counterfactual instances. The change in SHAP value from the input- to the counterfactual instance gives an impression whether or not the feature difference is relevant. For Feature 1, the SHAP change is negative which matches the direction of the change in prediction. On the other hand, the SHAP change in Feature 2 is positive, and the SHAP change in Feature 3 is small. These features differences are therefore likely not relevant to the counterfactual, and thus can be filtered. The counterfactual will take the values of the input instance for Feature 2 and 3 as long as the target prediction is kept.

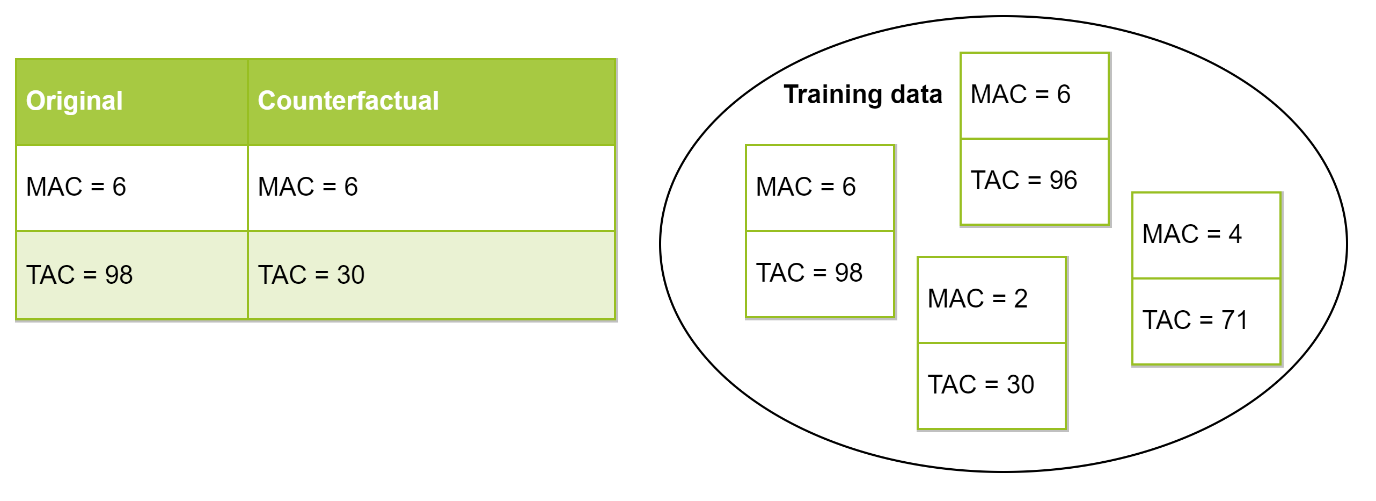
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Feature 1** | **Feature 2** | **Feature 3** |
| SHAP value in input | 0.300 | -0.200 | 0 |
| SHAP value in counterfactual | 0 | -0.150 | -0.001 |
| SHAP change | -0.300 | +0.050 | -0.001 |
| Candidate to be filtered from counterfactual? | No | Yes | Yes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

* 1. *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 .
   1. The feature’s top correlated variable is looked up from the list in step 1.
   2. Check that the value in combination with the value exists in the training data. If so, add 1 to the realism score. If not, add 0.
   3. If was also part of the set of features that differs between the original and the counterfactual instance, we return to step a and pick next most correlated feature with to be . In this way, the score is grounded in the values of a real instance.

Please refer to Figure x 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 TAC = 30 (from the counterfactual) in combination 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.



* 1. *Visualization*

To present the information to the user, a visual approach was used. We incorporated information that answered both questions into one visualization. Whereas previous implementations showed two separate graphs [44], we were inspired to incorporate them into one picture.

Visual graph with distributions [49].

For tabular data, there have been several approaches to present the information. For example, by a conversational statement or natural language [46, 56, 57]. This study also highlighted the importance of a grounded explanation; making sure the user understands under which conditions the explanation holds. The visualizations

Showing the tolerance of certain feature values for which the same prediction will hold [44].

It is important to show complete context of the explanation; what the input profile looks like, how it was predicted, and which features must not be changed [50, 52, 58].

Combine both into one explanation [43]

<How best visualization was derived>

* Only show changed features
* Scale features

- Each feature has their own scale

- Interpretation by experts

1. **Results and discussion**

It is important that explanations are evaluated for the application that they are used for [25, 31]. Therefore, we have only evaluated using our dataset.

Both objective and subjective evaluation are important. Comparing to past approaches [38].

Most importantly, user study [38, 46, 57, 59].

Performed a user study [57]

Using standard datasets such as LSAT (regression,3 features) [46], Diabetes (classification, 8 features) [34], HELOC loan application [44, 49], German Credit Risk [42, 57], multiple other ones [43].

Using domain knowledge to see if it makes sense [46].

Size (how many features changed) [35, 44]

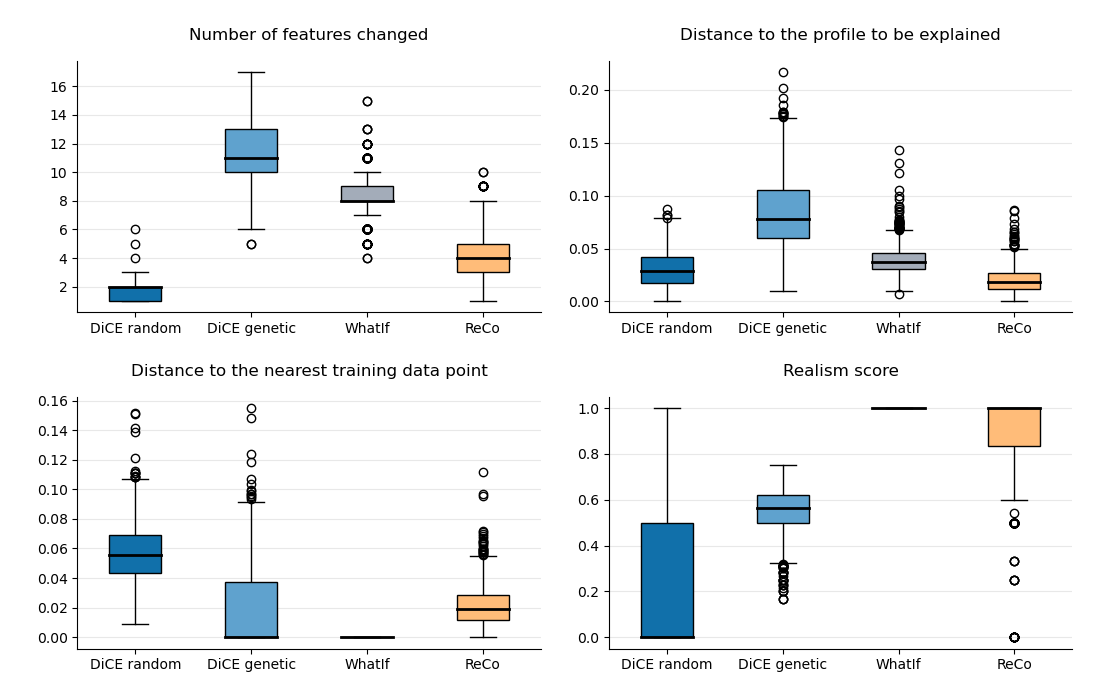
Although the current implementation of ReCo is used for regression, it only requires minor adaptations to fit classification as well.

* 1. *ReCo quantitative evaluation comparison to the state of the art*

To determine the quality of ReCo, we have compared it against the current counterfactual methods. WhatIf is our own implementation of Google’s What-If tool for searching the closest counterfactual from the training data [51].

While DiCE implements sampling approaches[38]. DiCE genetic is actually a Python adaptation of GeCo without its grouping constraints [40].

Even though SHAP can produce values that are lower for correlated features, they will not become negative while the true value is positive. ReCo mainly relies on the direction of the SHAP value, so whether it positively or negatively contributes to the prediction. Therefore, these inaccuracies are not important. 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.



While DiCE random performs best in terms of the number of 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.

An improvement can be seen when the genetic version is used; the median realism score almost hits a sufficient 0.6, and the distance to the training data is practically zero. We can attribute these better scores to the crossover of existing DNA profiles, though this crossover can still create unlikely feature combinations. It is interesting to see that this algorithm leads to significantly larger distances and feature differences. <explain why>

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.

ReCo seems to provide a happy medium.

* 1. *User study*
  2. *Future work*

The

1. **Conclusion**

This study describes

**Acknowledgements**

We are thankful to Corina Benschop for insightful discussions, Jerry for sampling 5000 DNA mixtures, BiS for participating in the user studies,

**References**

1. 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.

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. 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.

4. Biedermann, A., et al., *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, 2012. **6**(6): p. 689-696.

5. Paoletti, D.R., et al., *Inferring the Number of Contributors to Mixed DNA Profiles.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2012. **9**(1): p. 113-122.

6. Young, B.A., et al., *Estimating number of contributors in massively parallel sequencing data of STR loci.* Forensic Science International: Genetics, 2019. **38**: p. 15-22.

7. Grgicak, C.M., et al., *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, 2020. **47**.

8. Swaminathan, H., et al., *NOCIt: A computational method to infer the number of contributors to DNA samples analyzed by STR genotyping.* Forensic Science International: Genetics, 2015. **16**: p. 172-180.

9. 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**.

10. Coble, M.D. and J.-A. Bright, *Probabilistic genotyping software: An overview.* Forensic Science International: Genetics, 2019. **38**: p. 219-224.

11. 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.

12. 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.

13. Benschop, C.C.G., et al., *Multi-laboratory validation of DNAxs including the statistical library DNAStatistX.* Forensic Science International: Genetics, 2020. **49**: p. 102390.

14. Benschop, C.C.G., et al., *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, 2019. **42**: p. 31-38.

15. Bille, T., et al., *Interpreting a major component from a mixed DNA profile with an unknown number of minor contributors.* Forensic Science International: Genetics, 2019. **40**: p. 150-159.

16. Buckleton, J.S., et al., *The effect of varying the number of contributors in the prosecution and alternate propositions.* Forensic Science International: Genetics, 2019. **38**: p. 225-231.

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.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.

19. Marciano, M.A. and 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, 2019. **43**.

20. Mittelstadt, B., C. Russell, and S. Wachter, *Explaining Explanations in AI*. 2018.

21. Miller, T., *Explanation in artificial intelligence: Insights from the social sciences.* Artificial Intelligence, 2019. **267**: p. 1-38.

22. Carvalho, D.V., E.M. Pereira, and J.S. Cardoso, *Machine learning interpretability: A survey on methods and metrics.* Electronics (Switzerland), 2019. **8**(8).

23. Barredo Arrieta, A., et al., *Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.* Information Fusion, 2020. **58**: p. 82-115.

24. Lipton, Z.C., *The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.* Queue, 2018. **16**(3).

25. 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.

26. Adadi, A. and M. Berrada, *Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI).* IEEE Access, 2018. **6**: p. 52138-52160.

27. Kruijver, M., et al., *Estimating the number of contributors to a DNA profile using decision trees.* Forensic Science International: Genetics.

28. Commision, E., *Fostering a European approach to Artificial Intelligence*. 2021.

29. Fernández, R.R., et al., *Random forest explainability using counterfactual sets.* Information Fusion, 2020. **63**: p. 196-207.

30. Du, M., N. Liu, and X. Hu, *Techniques for interpretable machine learning.* Communications of the ACM, 2020. **63**(1): p. 68-77.

31. Murdoch, W.J., et al., *Definitions, methods, and applications in interpretable machine learning.* Proceedings of the National Academy of Sciences of the United States of America, 2019. **116**(44): p. 22071-22080.

32. 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.

33. 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.

34. 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.

35. Ramon, Y., et al., *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, 2020. **14**(4): p. 801-819.

36. Dandl, S., et al. *Multi-Objective Counterfactual Explanations*. in *Parallel Problem Solving from Nature – PPSN XVI*. 2020. Cham: Springer International Publishing.

37. Karimi, A.-H., et al. *Model-agnostic counterfactual explanations for consequential decisions*. in *International Conference on Artificial Intelligence and Statistics*. 2020. PMLR.

38. Mothilal, R.K., A. Sharma, and C. Tan. *Explaining machine learning classifiers through diverse counterfactual explanations*. 2020.

39. Sharma, S., J. Henderson, and J. Ghosh. *CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models*. 2020.

40. Schleich, M., et al., *GeCo: Quality Counterfactual Explanations in Real Time*. 2021.

41. Poyiadzi, R., et al., *FACE: Feasible and Actionable Counterfactual Explanations*. 2020. 344-350.

42. Moore, J., N. Hammerla, and C. Watkins, *Explaining deep learning models with constrained adversarial examples*. 2019. p. 43-56.

43. Keane, M. and B. Smyth, *Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI)*. 2020.

44. Grath, R.M., et al., *Interpretable Credit Application Predictions With Counterfactual Explanations.* ArXiv, 2018. **abs/1811.05245**.

45. White, A. and A. Garcez. *Measurable Counterfactual Local Explanations for Any Classifier*. in *ECAI*. 2020.

46. Russell, C. *Efficient search for diverse coherent explanations*. 2019.

47. Rathi, S., *Generating Counterfactual and Contrastive Explanations using SHAP*. 2019.

48. Guidotti, R., et al., *Factual and Counterfactual Explanations for Black Box Decision Making.* IEEE Intelligent Systems, 2019. **34**(6): p. 14-23.

49. Gomez, O., et al. *ViCE*. 2020.

50. Sokol, K. and P. Flach. *Desiderata for interpretability: Explaining decision tree predictions with counterfactuals*. 2019.

51. Wexler, J., et al., *The What-If Tool: Interactive Probing of Machine Learning Models.* IEEE Transactions on Visualization and Computer Graphics, 2020. **26**(1): p. 56-65.

52. Barocas, S., A.D. Selbst, and M. Raghavan. *The hidden assumptions behind counterfactual explanations and principal reasons*. 2020.

53. Bertossi, L., *Score-Based Explanations in Data Management and Machine Learning*. 2020. p. 17-31.

54. Chiandussi, G., et al., *Comparison of multi-objective optimization methodologies for engineering applications.* Computers & Mathematics with Applications, 2012. **63**(5): p. 912-942.

55. Gunantara, N., *A review of multi-objective optimization: Methods and its applications.* Cogent Engineering, 2018. **5**(1): p. 1502242.

56. Sokol, K. and P. Flach. *Conversational Explanations of Machine Learning Predictions Through Class-contrastive Counterfactual Statements*. 2018.

57. Sokol, K. and P. Flach, *One Explanation Does Not Fit All: The Promise of Interactive Explanations for Machine Learning Transparency.* KI - Kunstliche Intelligenz, 2020. **34**(2): p. 235-250.

58. Sokol, K. and P. Flach. *Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety*. 2019.

59. Adhikari, A., et al. *LEAFAGE: Example-based and Feature importance-based Explanations for Black-box ML models*. in *IEEE International Conference on Fuzzy Systems*. 2019.

**Supplementary Material**