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

**Bringing XAI to predictions of the number of contributors**

**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. This becomes increasingly difficult when the number of contributors rises [1, 2]. Most software that is used for DNA interpretation requires the NOC to be entered by the user, which influences the calculation of the evidence [3-5]. <more refs>

Valuable steps have been made to develop methods that can more accurately predicting 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 [6]. The improvement mainly corresponds with incorporating more information such as the Total Allele Count (TAC) [7], peak heights, drop out and stutter rates [8, 9], population allele frequency [10] <6 is ref naar True Allele? Dat is een PG system, geen NOC model. Ik mis hier nog een aantal refs. Aangeven dat je een paar voorbeelden geeft of een completere lijst met referenties geven. Als je het laatste wilt dan kan ik het wel aanvullen.> . From the multitude of models to estimate the NOC, machine learning models have shown to outperform standard methods on both accuracy and speed [11, 12]. However, machine learning algorithms are often considered to be black-boxes [13-19], as predictions are made based on generalization from training data, but the exact mechanism is not easily understood. This raises the question of how to make these predictions understandable. Without explanations, it is challenging for experts to determine if they should trust the prediction or not. With more information, it might also be easier to defend why a certain NOC was chosen over the other.

A method using a decision tree was presented as a more transparent way to use machine learning [20]. 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 applied to this domain.

* 1. *eXplainable Artificial Intelligence*

One of the main flaws of more complex (deep) machine learning algorithms is their lack of transparency [13-19]. 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 [21]. Though NOC estimation does not directly influence the DNA donors without involvement of human experts, these experts can be better 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 [16, 17]. A decision tree could be considered a transparent model, since it shows each decision made for any input to reach 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. Once 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 this explanation, we can choose to leverage some structures of the model, or create a model-agnostic explanation. Model-agnostic explanations do not make any assumptions of the type of model they are explaining. 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) [13, 15-19, 22, 23]. 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?”*. Some research has implemented SHAP to real-life cases such as predicting hypoxia based on clinical data [24], and predicting the most fitting eye-surgery type [25]. 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 [14]. 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 [26]:

“*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 [26-41]. 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 [26, 28-33, 35-38, 42]. Though some methods use or Euclidean distance [33, 37], 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 [26, 28, 30-32, 36, 38]. 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 [27, 28, 30, 32, 34, 35, 37, 38, 40, 41], 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 [28-32, 38], or providing counterfactuals that are actionable; meaning that the changes can be acted upon to reach that alternative outcome [28-30, 33, 34, 41]. 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 [35, 43], or can be artificially sampled [26, 28-32, 36, 37, 39-41]. 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 [26, 29, 31, 36], or take the input and perturb its feature values until a different outcome is reached [28, 30, 32, 37, 39-41]. While most tackle this by randomly changing feature values [26, 29, 30, 36, 37, 39, 41], some take a more sophisticated route by using a genetic algorithm [28, 31, 32, 40]. 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 [27, 39]. By only changing the features from the input instance that have negative SHAP values for target class B, a counterfactual could be found [39]. 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 [27].

None of these methods are suitable for datasets with correlated features, as they would produce unlikely feature combinations.

One issue that is largely overlooked or handled quite poorly is realism. One way that the realism of counterfactual is considered, is by its distance to the closest training data point [28]. Though this can give an impression of the general relation to the training data, it does not account for the correlation between features.

As an example, consider a profile with a Total Allele Count (TAC) of 98 and a Maximum Allele Count (MAC) of 6, that was predicted to have 4 contributors. To generate a counterfactual with a prediction of 3 contributors, the program might propose a profile with a TAC of 30. Though this would make the model predict a NOC of 3, the combination of the original MAC value of 6, with the new TAC value of 30 is impossible. There are a total of 23 loci in the profile; if there are 30 alleles in the entire profile (TAC), that would leave either 1 or 2 alleles per locus (30/23). It would thus be highly unlikely, or even impossible to have a locus with 6 alleles (MAC).

Another approach is to limit features to certain ranges defined by the user or based on the training data, which does not solve the problem of unlikely feature value combinations [29, 31]. In a post-hoc filtering approach, the user can supply causal knowledge to the model to remove counterfactuals with impossible feature combinations [30]. This again leaves the responsibility with the user, is time-consuming for many correlated features, and has the risk that all produced counterfactuals must be removed since the filtering happens after the examples are generated. There exists a recent Julia implementation that limits the feature combinations to the ones made in the training data, though the user needs to supply each of these relations manually [32].

FACE makes use of the training data to form a connected path from one instance to one of the target prediction [33]. This ensures that the query instance could be transformed into the counterfactual one. This relies heavily on the data having certain areas of higher density, and having connected areas from one prediction to the next. This might not be present in sparser real-life datasets. Their focus lies on if the transition is feasible, not if the end result is. There is not constraint for correlated features.

A method that derives counterfactuals from training instances has achieved more realistic instances [35]. They point out however, that their method will fail for datasets where there are no inherent good counterfactuals; with two or less feature differences.

Though several studies have brought up the issue there should be a way to handle correlated features [29, 30, 44], no method has inherently adapted this into their method that is viable for real-life data. To the best of our knowledge, we are the first to develop a method that is inherently 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.

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 [20]. 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 to create a less sparse feature space. To this aim, we used EuroForMix software to generate 5000 more profilesDit is niet met EuroForMix zelf gedaan, maar met code zoals geïmplementeerd in EuroForMix. Ik zou aan Jerry vragen om dit te omschrijven.>. Appendix x shows the parameters used. <more information about how EuroForMix generates realistic profiles>

* 1. *Machine Learning model*

Originally, the estimation of the NOC was treated as a classification problem, such that where is an input profile, , and is a multi-class classification model [11]. 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. Therefore, a model-agnostic explanation method is preferable.

* 1. *Desiderata explanations*

From meetings with the end users, we determined that there were two main questions of interest:

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

To answer question 1, we determined that the use of SHAP values would be sufficient to give an impression of feature importance.

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

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 above constraints have mostly been covered quite well in the literature.

The notion of Multi-Objective Counterfactuals was first proposed with four objectives, solved by a genetic algorithm [28]. 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.

To measure the distance between two instances, we implemented an L1 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, the counterfactual profile. Though much of the literature scales each distance by the Median Absolute Deviation (MAD) [26, 30, 35, 36, 38], 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 [28, 29]. 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 [26, 32, 38]. Moreover, presenting multiple, possible contradicting examples does not seem like user friendly introduction to counterfactual explanations.

* 1. *Realistic Counterfactuals (ReCo)*

From the original profile and its prediction , the user can define 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 wrong predictions are presented as examples. These candidates are scored by a weighted sum of two objectives; the distance score as defined in Equation 1, and the number of different features as defined in Equation 2. We select the instance with the minimum score to be the counterfactual instance:

The counterfactual instance is a data point from the training data. It is therefore a realistic data point to present. The biggest issues with such examples are:

* 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. Table 1 shows a practical example.

1. Start with the set of features that have different values between the original- and the counterfactual instance.
2. Compute the SHAP values for each of the features, for both the original- and the counterfactual instance. Then subtract the SHAP values of the original 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 original- 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 original instance, we need to remove the irrelevant feature differences. If the prediction goes down from the original- 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 misaligned 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 attempt to remove the feature differences that have caused the most misaligned SHAP change. This means that the feature value of the counterfactual is replaced with the original feature value. As long as the counterfactual prediction stays the same, more feature differences can be removed.
5. Once the target prediction can no longer be reached, all irrelevant features differences are filtered out. The final counterfactual is then defined as:

Table 1: Example of how a counterfactual instance is filtered. The original 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 original- and counterfactual instances. The change in SHAP value from the original- 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 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 original instance for features 2 and 3 as long as the target prediction is kept.

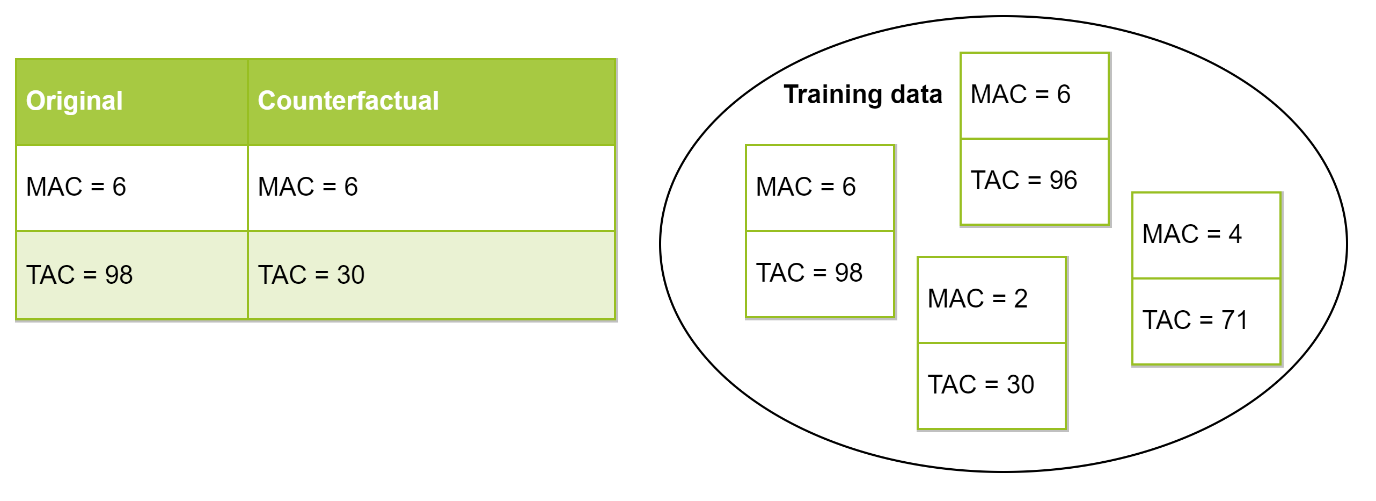
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Feature 1** | **Feature 2** | **Feature 3** |
| SHAP value in original | 0.300 | -0.200 | 0 |
| SHAP value in counterfactual | 0 | -0.150 | -0.001 |
| SHAP change | -0.300 | +0.050 | -0.001 |
| Candidate to be filtered from counterfactual? | No | Yes | Yes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

* 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.

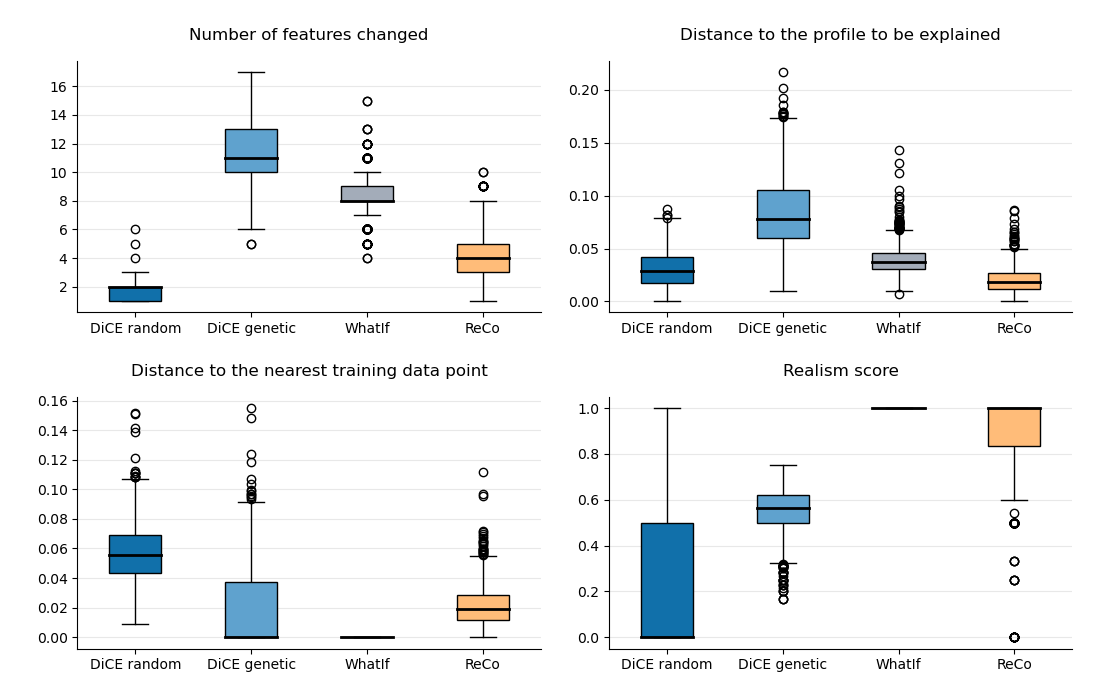


* 1. *Visualization*

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 [36], we were inspired to incorporate them into one picture. Visual graph with distributions [41]. For tabular data, there have been several approaches to present the information. For example, by a conversational statement [45]. This study also highlighted the importance of a grounded explanation; making sure the user understands under which conditions the explanation holds. The visualizations

1. **Results and discussion**
   1. *ReCo quantitative evaluation compared to the state of the art*

To determine the quality of ReCo, we have compared it against the current counterfactual methods. WhatIf is a method based on training data. While DiCE implements sampling approaches. DiCE genetic is actually a Python adaptation of GeCo.



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 with EuroForMix, 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. 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.

4. 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.

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

6. 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.

7. 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**.

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. Bauer, D.W., et al., *Validating TrueAllele® Interpretation of DNA Mixtures Containing up to Ten Unknown Contributors.* Journal of Forensic Sciences, 2020. **65**(2): p. 380-398.

10. 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.

11. 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.

12. 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**.

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

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

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

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

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

18. 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.

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

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

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

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

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

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

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

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

31. 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.

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

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

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

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

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

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

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

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

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

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

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

43. 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.

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

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

**Supplementary Material**