**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 in short tandem repeat profiles has been shown to obtain good accuracy. However, these predictions are not understandable to the biologist user who would normally determine the number of donors themselves. Therefore, we have created 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 the feature combinations. Since the features from STR data are highly corelated, we have implemented a new method that generates realistic counterfactuals on highly correlated data, and also cuts the number of feature changes in half as compared to presenting the closest counterfactual training data point.

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. Most software that is used for DNA interpretation requires the NOC to be entered by the user [1]. This could make the difference of including or excluding a person of interest, which is important in court. The expert in question must be able to explain why.

There have been a multitude of different methods to estimate the NOC. From methods that use only allele counts in a qualitative approach such as the MAC [2], Maximum likelihood approach [3],

Probabilistic mixture algorithm [4]

NOCit tool [5] which is a likelihood-based approach taking into account both qualitative and quantitative data.

1-3 person mixtures. Synthetically created from 58 known donors. Modelling the types of peaks, peak ehights, drop out and stutter rates. posterior probability via a Monte Carlo-based approach. maximum probability method with NOCIt resulted in accuracies of approximately 83%. Computationally slow! Up to 9h for a 5 person mixture.

True allele up to ten contributors? probability model

nC-tool [6] uses both the MAC and Total Allele Count (TAC), as well as categories of drop-out to obtaining better results.

quantitative continuous model peak heights in the DNA profile and considers the

effect of artifacts and allelic drop-out. By using this software, the likelihoods of 1–4 persons’

contributions are calculated, and the most optimal number of contributors is automatically

determined; Kongoh was validated using 27 two-person mixtures, 27 three-person mixtures, and 18 fourperson mixtures. These mixtures were experimentally prepared using non-degraded DNA

from pristine blood samples.

Machine learning approach

PACE random forest [7]: 969 non-simulated DNA samples of 1 to 5 contributors generated from a combination of 120 individuals. They achieved about 90% accuracy on identifying 1, 2, 3, 4+ profiles.

Bayesian probability framework TrueAllele [8]. Closed-source. Minutes to longer for more complex

The results that machine learning methods can obtain has been demonstrated to outperform standard methods [9]. This raised the question however of making these predictions understandable for experts. Without explanations, the experts cannot determine if they should trust the prediction or not. It is also difficult to defend why the expert picked one NOC over the other, if they only relied on the output of a machine learning model.

A decision tree was presented in a paper that made an attempt to make predictions more transparent [10]. However, this led to less accurate predictions (77%) and relied heavily on the filtering of artefacts, which are inherently part of STR data and the NOC estimation process. Moreover, from the explainable AI community, decision trees are not considered an explanation [source]. You are also forced to then use a decision tree, while more interesting predictors are being modelled that may perform even better in the future. The advantage of using an explanation per prediction, is that they are more accurate, and there is no need to filter through an entire decision tree.

* 1. *eXplainable Artificial Intelligence*

The field of eXplainable Artificial Intelligence (XAI) is important whenever any type of AI is used. Especially when the users of such an algorithm are not familiar with how it works.

[SHAP explanation]

* 1. *Counterfactual explanations*

A counterfactual is defined as follows; consider an input instance , a black box machine learning model .

Desiderata for explaining decision trees [11].

Most of the literature aims to find counterfactual explanations that have:

* The desired output (are valid) [12-27]
* The smallest distance to the profile we want to explain (are proximal) [11, 13-19, 21-24].

Some studies also include:

* The fewest number of feature differences in comparison to the profile we want to explain (are sparse) [12, 13, 16, 18, 20, 21, 23, 24, 26, 27]

Though diversity is often encouraged [13, 15-18, 24], we do not see its value for this problem currently. This is mainly because the explanations are new to the users and they do not want to be confused by seeing multiple, possibly contradicting examples.

Actionability [13, 15, 16, 19, 20, 27]. Unlike studies about loan applications and similar situations, actionability is not a goal of this study. The DNA profiles cannot and will not be altered in the future.

Random **sampling**-based approaches [14-16, 22, 23, 25, 27], sampling through a genetic algorithm [13, 17, 18, 26]. Linear optimization [24], learning a local decision tree [26], greedily moving from the original instance until the decision boundary is reached [27].

Genetic algorithms are based around the idea of selecting the ‘fittest individuals’; at first, a starting population is used. This could be the training set of instances with the target prediction, randomly generated instances [17], or the original instance [18]. These are then evolved by 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 methods with the best fitness score are kept. This fitness score is usually defined by the distance.

A method based on the training data, assumes that there are some inherent, sparse counterfactual training points, that then can be adapted to become counterfactuals for other data points [21].

Using the gradient of the loss with respect to the input [20], a method that is based on the data, not on the classifier. None of these methods are suitable for datasets with correlated features, as they would produce unlikely feature combinations.

A similar piece of work uses SHAP values for the current instance to be explained, for both the predicted class A, and target class B [25]. Specific counterfactual instances are then generated by sampling nearest neighbours, changing only the features from the original instance that have negative SHAP values for class B. This approach suffers from the fact that 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, a paper discusses using LIME and SHAP to generate counterfactuals from [12]. However, their method is based on highly-dimensional (1000+), behavioural or textual data. They produce counterfactuals by iteratively setting the top contributing features to 0, until the target class is reached. This is not viable in our dataset, since setting a value to 0 does not usually correspond to a realistic feature value.

* 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) 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 [10]. The NOC was based on ground-truth information. Each profile was then represented by 19 features such as allele counts, allele frequencies and peak heights. These features are almost all very highly correlated.

This high-dimensional dataset was too sparse for generating counterfactual explanations using the training data; the most similar profile was still very different.

In sampling-based explanation approaches, a dense neighbourhood is created around the current profile to be explained. However, none of these methods can handle correlated features. Therefore, strange

* 1. *Machine Learning model*

Originally, the estimation of the NOC was treated as a classification problem [1]. 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 better results. 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 NFI is looking to improve the model and used features in the future. Therefore, a model-agnostic 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.

To map out what the counterfactual explanations must accommodate, a list of desiderata was determined.

* Model-agnostic
* Interactive (target can be chosen by user)
* Valid (has desired output)
* Sparse (has not too many feature differences)
* Proximal (is close enough)
* *Robust (the same every time for the same profile)*
* Realistic (makes combinations of feature values that make sense)

The level of interaction we determined was most valuable, was letting the user input the target prediction. Most existing methods assume a binary case, and thus do not have to concern themselves with which target 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.

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 [13]. 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 data instances, we implemented an L1 norm function as shown in Equation 1.

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

Where represents the range of the -th feature, the number of features, the profile to be explained, the counterfactual profile. Though some methods use distance [19, 23],

distance is the measure of choice by the literature as it does not blow outlier distances out of proportion as L2 distance tends to do [13, 14, 16-18, 22, 24]. Though much of the literature scales each distance by the Median Absolute Deviation (MAD) [14, 16, 21, 22, 24], 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 [13, 15]. 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.

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

However, **realism** is often overlooked or handled quite poorly. One way that the realism of counterfactual is considered, is by its distance to the closest training data point [13]. 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 [15, 17]. In a post-hoc filtering approach, the user can supply causal knowledge to the model to remove counterfactuals with impossible feature combinations [16]. 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 [18].

FACE makes use of the training data to form a connected path from one instance to one of the target prediction [19]. 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 [21]. 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 [15, 16, 28], 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. *Realistic Counterfactuals (****ReCo****)*

From the original profile and its prediction , the user can define a target prediction that is different from the original prediction . ReCo then finds all instances from the training data with the target prediction . These must match with their ground truth NOC. 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 the most 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. In that case, 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 |

A similar piece of work uses **SHAP** values for the current instance to be explained, for both the predicted class A, and target class B [25]. Specific counterfactual instances are then generated by sampling nearest neighbours, changing only the features from the original instance that have negative SHAP values for class B. This approach suffers from the fact that 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, a paper discusses using LIME and SHAP to generate counterfactuals from [12]. However, their method is based on highly-dimensional (1000+), behavioural or textual data. They produce counterfactuals by iteratively setting the top contributing features to 0, until the target class is reached. This is not viable in our dataset, since setting a value to 0 does not usually correspond to a realistic feature value.

Desiderata for explaining decision trees [11].

* 1. *Realism score*

We present a novel realism score which is calculated as follows:

1. When the dataset is loaded, a list is generated for each feature, ranking all other features according to their correlation:

.

1. Once a counterfactual is found, the feature differences with the original instance are defined as:
2. Then for
3. each element in :
   1. Look up its top correlated feature,  and take that feature’s value
   2. Look up the combination of with of with the feature value in step a in the training data.
   3. If the combination exists, the realism score is incremented by 1, otherwise by 0.
   4. If the feature in step a was part of , we repeat step a with the next highly correlated feature.

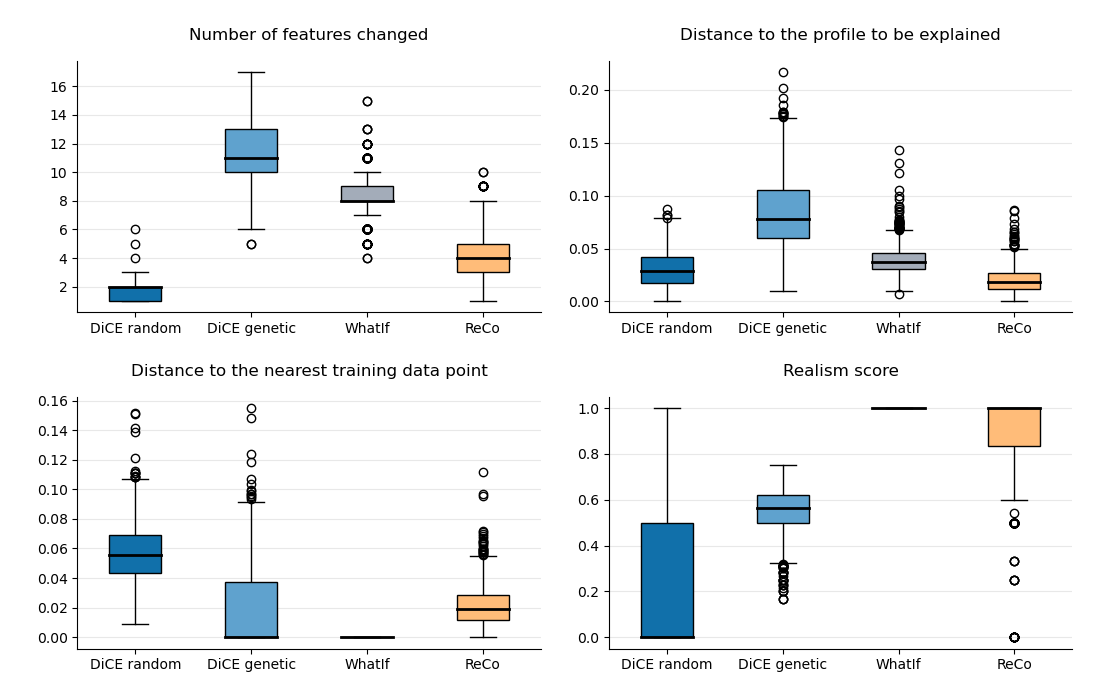
This score captures the issues with sampling methods quite well.

* 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 per [22], we were inspired to incorporate them into one picture. Visual graph with distributions [27]. For tabular data, there have been several approaches to present the information. For example, by a conversational statement [29]. 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 the higher realism score to the ‘mating’ of the DNA profiles.

* 1. Future work

The

1. **Conclusion**

This study describes

**Acknowledgements**

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

**References**

1. P.M. Schneider, R. Fimmers, W. Keil, G. Molsberger, D. Patzelt, W. Pflug, T. Rothamel, H. Schmitter, H. Schneider, B. Brinkmann, The German Stain Commission: recommendations for the Interpretation of mixed stains, Int. J. Legal Med. 123 (2009) 1–5.

1. 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.

2. 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.

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. 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.

5. 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.

6. 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**.

7. 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**.

8. 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.

9. 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.

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

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

12. 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.

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

14. 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.

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

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

17. 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.

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

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

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

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

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

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

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

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

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

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

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

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

**Supplementary Material**

Supplementary Table 4. Overview of the 11 locus features which were calculated for each of the 23 autosomal loci within the PowerPlex® Fusion 6C profiles.

|  |  |  |
| --- | --- | --- |
| **Number** | **Feature** | **Details** |
| 1 | Allele count | Number of alleles |
| 2 | Minimum NOC | Allele count / 2, rounded up to 0 decimals |
| 3 | Maximum PH | The largest, smallest, mean, median or standard deviation of the peak height (PH, in RFUs) of alleles at the particular locus |
| 4 | Minimum PH |
| 5 | Mean PH |
| 6 | Median PH |
| 7 | Standard deviation PH |
| 8 | Minimum AF | The lowest or highest allele frequency (AF) of an allele, or the sum of the allele frequencies of the alleles, or the percentage of alleles that are within the population database. *I.e.*: |
| 9 | Maximum AF |
| 10 | Sum AF |
| 11 | Percentage of AF |