Predicting DNA repair-deficient genotypes based on Cas9-induced DNA repair products

by

P.J.M. Verkooijen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday January 29, 2021 at 13:00.

Student number:	4223322	
Project duration:	January 2020 – January, 2021	
Thesis committee:	Dr. J. S. Gonçalves,	TU Delft, supervisor
	Prof. dr. ir. M. J. T. Reinders,	TU Delft
	Prof. dr. M. Tijsterman,	LUMC

TU Delft

Predicting DNA repair-deficient genotypes based on Cas9-induced DNA repair products

P. J. M. Verkooijen
Faculty of Electrical Engineering, Mathematics & Computer Science
Delft University of Technology
Email: p.j.m.verkooijen@student.tudelft.nl

Double strand breaks are lesions to the DNA and can be fatal for cells. Therefore these breaks are repaired, primarily by one of the three major repair pathways. Two of these pathways are non-homologous end-joining (NHEJ) and theta-mediated end-joining (TMEJ). These pathways leave genetic alterations in their repair products, a form of DNA damage. DNA damage is linked to several diseases such as cancer. Understanding of these pathways is important and being able to recognize which pathways are active can be beneficial for research. In this work, repair products are used to predict repair-deficient genotypes using Cas9-induced repair products. Ku80 and PolQ deficient genotypes are used, impairing NHEJ and TMEJ respectively. The ability to recognize a repair-deficient genotype is tested using two predictive tasks. First statistical machine learning algorithms are used to predict the genotype where a repair product can be found. This is done by only using a single repair product as input. Secondly, a set of Cas9-induced repair products from a single cell culture is used to predict the genotype of that cell culture. Results show that when given a single repair products, models have difficulty predicting the correct genotype. However, results are modest and the best classifier achieved an AUC of 0.76*. For predicting the genotype of a cell culture using multiple repair products of that culture showed really promising results. When predicting on cell cultures with breaks induced on a target site which the model has seen in the training data, results are near perfect. Predicting on unseen target sites shows that there is room for improvement but the best performing models showed an average AUC of* 0.879 *across target sites. A Results show that Cas9-induced repair products can be used to predict repair-deficient genotypes.*

1 Introduction

DNA contains the genetic information of a cell and is vital for all living beings. However, the integrity of DNA is frequently compromised by lesions. One such lesion is a double strand break (DSB), where both strands of the DNA are severed. An illustration of a DSB can is shown in Figure 1b. Left untouched DSBs can lead to cell death. Cells have evolved to handle DSBs and are able to repair these by connecting the two severed strands of DNA. However DNA repair is not always flawless and can leave genetic alterations. This form of DNA damage is harmful and linked to aging [1,2] the formation of cancers [2,3] and neurodegeneration [4]. Three major pathways are known for the repair of DNA, of which two are known to leave genetic alterations. Understanding these pathways and DSB repair can be crucial for understanding the harmful consequences of DNA damage. The ability to differentiate the DNA repair pathways could lead to new insights and can be done by analyzing genetic alterations left by repair pathways.

DNA repair is largely done by three different pathways. The three major pathways are homologous recombination, non-homologous end-joining (NHEJ) and theta-mediated end-joining (TMEJ). The HR pathway uses a sister chromatid as a blueprint to repair a DSB error-free. This pathway is well known and reviewed here [5, 6]. NHEJ is known as a largely error-free pathway, but can leave alterations in certain situations [7]. NHEJ is active throughout the cell cycle [8] and is expected to repair up to 75% of all repair events in human cells [9]. NHEJ repair relies on, among others, the protein Ku80. NHEJ is a pathway known to make use of microhomology, with up to 60% of the repair in human cells showing microhomology of one or two base pairs [10]. Microhomology is the process of using complementary sequences of a few base pairs on opposing strands of the DSB in order to connect the severed strands. The final pathway TMEJ is known as an error-prone pathway. The pathway thanks its name due to it relying on polymerase theta (PolQ), which is shown true for several different organisms [11, 12, 13]. Polymerase theta is connected to low fidelity DNA repair [14]. TMEJ is not as active as NHEJ, it is a crucial backup in cells when other pathways cannot be used [15]. Evidence exists that TMEJ may be active during the S phase of a cell, primarily due to genetic alterations being left from TMEJ at replication structures [16, 17]. Similar to NHEJ, TMEJ makes use of microhomology. TMEJ is further known to leave templated insertions [18, 19]. A templated insertion uses a DNA sequence nearby the DSB as a blueprint to reconnect the severed strands of the DSB. Essentially, this duplicates a part of the DNA to help repair the DSB. From these three pathways, NHEJ and TMEJ show the most distinct DSB repair due to leaving genetic alterations.

The DNA sequence after the repair of a DSB is known as a repair product. Some repair product will show genetic alterations, depending on the pathway used to repair a DSB. Repair products show sufficiently distinct patterns, that three machine learning exist able to predict the expected repair products found at Cas9-induced DSBs. One of these works makes use of a deep neural net [20], while the other two use logistic regression classifiers [21, 22]. Patterns can be found in repair products originating from the different pathways. This is shown by the NHEJ and TMEJ pathway-specific features presented in [23]. Although this is shown in mouse embryonic cells and might differ per organism, one such feature is that repair products from TMEJ show different deletion sizes from NHEJ. Thus the pathway used to repair a DSB plays a large role in which repair products found. Which pathway is used for DSB repair is dependent on several variables, best depicted by the decision tree shown in [24]. Variables for pathway selection include the chromatin context and the cell cycle phase.

About a decade ago, CRISPR-Cas9 was first used to consistently create DSBs at precise locations [25]. This technology greatly helps analyzing DNA repair mechanisms and repair products. DSBs can be induced in a cell culture at a specific location in the DNA, the target site. This target site can then be sequenced, making it possible to compare and analyze large amounts of repair products all formed at the same target site. This can be extended by using cell cultures deficient of a DSB repair pathway. Both NHEJ and TMEJ are dependent on a distinct protein, Ku80 and PolQ respectively. Knocking out these proteins would result in NHEJ and TMEJ being inactive. Thus the Ku80-/- and PolQ-/- genotypes contain repair products created by different DNA repair pathways. One question which arises, is whether these repair products are unique enough to use them to recognize and predict the genotype in which they were found. The goal of this thesis is find the degree in which it is possible to use repair products for genotype prediction. This will be tested using two predictive tasks:

1. Can the genotype in which a single repair product is found be predicted.
2. Can the genotype of a cell culture be predicted using all repair products from the culture.

To answer these two predictive tasks, statistical machine learning algorithms will be used. To train models, Cas9-induced repair products originating from mouse embryonic stem cells are used. These repair products originate from three different genotypes, namely Ku80-/- which is NHEJ deficient, PolQ-/- which is TMEJ deficient and a wild type which is not deficient in repair pathways. Repair products in the dataset originate from ten different target sites.

The first predictive task will show if for repair products it can be predicted in which genotype they are found. If models are able to predict genotypes correctly, it would indicate that genotypes leave very distinctive repair products. The second predictive tasks will show how well a genotype can be predicted using multiple repair products originating from the same target site. Rather than looking at the distinctiveness of a single repair product, a distribution of the repair products in the cell culture is used to make a prediction. Rather than showing distinctive features, this can indicate whether repair products from a genotypes follow a distribution which can be used to generalize models.

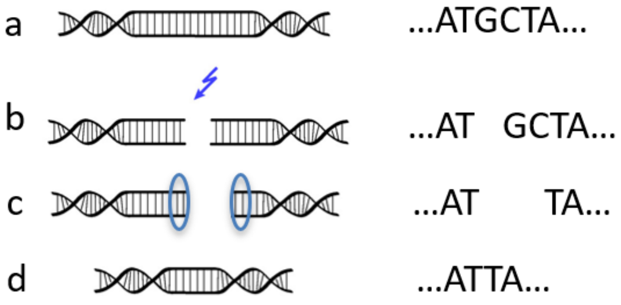


Fig. 1. Toy example of how repair products are obtained. **a:** A target site is chosen. **b:** A double strand break is induced at the target site. **c:** The cell repairs the double strand break. **d:** The DNA sequence after repair is the repair product. Note that the repair product left a mutation. Two nucleotides were deleted during the repair.

For both predictive tasks several models are created. First, models are trained using repair products from all target sites. These models are evaluated on a test set with independent repair products from all target sites. This will show how well models can generalize genotypes on learned target sites. Secondly, new models are trained using repair products from nine of the ten target sites. Evaluation will be done on the left out target site. This gives insight in how well models can generalize genotypes across new target sites.

2 Methods

Given a dataset containing Cas9-induced repair products, the goal is to predict the repair-deficient genotype in which the DSB was repaired.

2.1 Dataset

A dataset was kindly provided by the Tijsterman lab at the Leiden University Medical Center (LUMC). The data set contains repair products from CRISPR-Cas9 induced DSBs. The cells used are mouse embryonic stem cells. How repair products are obtained is shown in Figure 1.

Repair products are sequenced from Cas9-induced DSBs from 36 cell cultures. In each cell culture, DSBs are induced at a single target site. In total, ten different target sites have Cas9-induced DSBs. For each target site, cell cultures are grown with three different genotypes.

	Genotype		
Pathway	Wild Type	Ku80-/-	PolQ-/-
HR	✓	✓	✓
NHEJ	✓	-	✓
TMEJ	✓	✓	-
Other	✓	✓	✓

Table 1. Shows double strand break repair pathways active per genotypes. HR: Homologous recombination. NHEJ: Non-homologous end-joining. TMEJ: Theta-mediated end-joining. Other: Alternative pathways for DSB repair.

The three genotypes used are Ku80-/-, PolQ-/-, wild-type (WT). These first two genotypes result in repair pathway deficiency, as can be seen in Table 1. Thus, repair products originating from 10 different target sites are present. For each target site, repair products are retrieved from cultures with a Ku80-/-, PolQ-/- and WT genotype. Two target sites were sequenced twice with different primers. For each primer set a unique cell culture is present in the data. Thus for two target sites, cultures of two of each genotype are grown and sequenced with different primers. Thus the dataset contains sequence data of Cas9-induced repair products. A total of ten different target sites are used, of which 2 were sequenced twice with different primers. For each target site one culture is present with one of the three genotypes. This results in $8(targetsites) * 3(genotypes) + 2(targetsites) * 3(genotypes) * 2(primersets) = 36$ cell cultures.

The sequence data of the 36 cell cultures are kindly preprocessed by Robin van Schendel, a bioinformatician at the Tijsterman Lab. The sequence data is processed using a custom code, which applies quality control to sequence data. Furthermore, additional information is added to each repair product describing the product. This information includes whether deletions took place, the length of the deletion or if microhomology took place. This information is retrieved by comparing the DNA sequence of the target site with the DNA sequence of the repair product. Part of the information added during the preprocessing can be found in supplementary materials A. Keep in mind that supplementary materials A only shows information used by machine learning algorithms presented in this work. More information was present in the dataset, but has been removed due to not being used throughout this work. The final step of the preprocessing groups repair products from the same cell culture with equal DNA sequences and adds a percentage showing how frequent the repair product is seen. This value is called the frequency and has a value of $0 < frequency =< 1$. The higher the frequency, the more probable it is to see that repair product after repairing a Cas9-induced DSB. A toy example of how the preprocessed dataset looks can be found in Table 2. A common pattern seen is that most repair products occur infrequently, with around 75% of the repair products in a culture having frequency < 0.001. This means that even though there is a high variety in repair products per target site and genotype, only a select few are commonly found.

To summarize, the dataset contains repair products cell cultures with Cas9-induced DSBs. Repair products originate from one of ten target sites and each cell culture has one of the Ku80-/-, PolQ-/- and WT genotypes. For each repair product it is known in which genotype they are created, at which target site they were found, how frequently the repair product was found and the information shown in supplementary materials A. This information is retrieved from comparing the target sites original DNA sequence with the DNA sequence of the repair products.

2.2 Prediction tasks

The goal is to see how well repair-deficient genotypes can be predicted using repair products. This is answered using three predictive tasks. The first task tries to make prediction using only a single repair product. The second task will see how well genotype can be predicted when using a set of repair products, all coming from the same cell culture. The last tasks is set up in order to show how well models are able to generalize learned patterns across repair products from new target sites. This is done using by evaluating the first two prediction tasks in twofold.

For the first prediction task machine learning techniques will be used to predict the repair-deficient genotype in which a repair product is created. Thus, when a machine learning model is supplied with a single repair product it should be able to predict whether it was formed in a Ku80-/- or a PolQ-/- cell culture. Logically following, due to the repair-pathway deficiency in these genotypes it also means models predict by which repair pathway a DSB is **not** repaired. For example, predicting a repair product was formed in a Ku80-/- genotype also means that the model predicts it is not repaired by NHEJ. Such a statement is not possible for a WT genotype prediction. Therefore, the WT genotype is not used for this predictive task. Several predictive models are created to test the predictive task. How and which models are created will be discussed in subsubsection 2.3.1 and shown in Figure 2, orange column.

The second prediction task will also use machine learning technique in order to predict the repair-deficient genotype of a cell culture using repair products originating from that cell culture. Keep in mind that in a cell culture all repair products originate from the same target site. In order to make predictions, repair products from a culture will be grouped and described using summary features. These features will be used by machine learning algorithms. The features used can be found in supplementary materials A. Due to using multiple repair products, the general behaviour of repair products in a genotype can be learned. For example, summary features can show different average deletion lengths in different genotypes. This makes it possible to differentiate the WT genotype from the Ku80-/- and PolQ-/-

CultureID	Frequency	Genotype	Target site	Products DNA Sequence	Deletion	DelLen	Insertion	InsLen
1	0.8	Ku80-/-	TS_a	..agatgatc..	-	0	-	0
1	0.2	Ku80-/-	TS_a	..agaatc..	tg	2	-	0
2	1	PolQ-/-	TS_a	..agcatc..	atg	3	c	1
3	1	WT	TS_a	..agaatc..	tg	2	-	0

Table 2. Toy dataset for given target site TS_a with DNA sequence ..agatgatc.. Each row depicts a repair product after preprocessing. The columns from left to right indicate: Cell culture ID, from which culture does the repair product originate. Frequency, a percentage showing how frequently the repair product was seen. Genotype of the cell culture. Target site where the DSB is induced. Product DNA sequence shows raw DNA sequenced information. This data is normally around 200 base pairs long. Deletion shows the deleted nucleotides when comparing the repair product sequence with the original target sites sequence. DelLen shows the number of nucleotides deleted. Insertion shows the inserted nucleotides when comparing the repair products sequence with the target sites original DNA sequence. Note that it is possible to find equal repair products in cell cultures with different genotypes (row 1 and 4). More features are present in the data, of which most can be found in supplementary materials A.

genotypes. Therefore, models trained for cell culture genotype prediction will use cell cultures of all three genotypes. If successful, models could be trained to predict more genotypes. The ability to predict a cell cultures genotype is tested with several predictive models, which is discussed in subsubsection 2.3.2 and shown in Figure 2, blue column.

2.2.1 Learning algorithms and training

The predictive tasks will be examined using several predictive models. For both predictive tasks, three statistical learning algorithms will be compared. The algorithms used are the logistic regression classifier, a k-nearest neighbour (k-nn) classifier and a random forest classifier.

The logistic regression classifier is a linear parametric classifier which uses regression to fit a logistic function on the features of the training data. This function is can be used to determine the probability of a new data sample belonging to a certain class. The k-nearest neighbour is a distance based classifier, which memorizes training data. When given a new data sample, it calculates the distance to each sample in the training data. The k closest training samples are considered most similar to the new data and will be used to make a prediction. Each label of the k closest neighbours is counted and the label with the highest count will be the prediction of the new sample. K-NN classifiers will be trained with two k values. First a lower k is used, with value $k < 10$. Secondly a higher k is used of value $100 <= k <= 500$. The exact value of k is determined using a grid search, explained below.

A random forest classifier uses an ensemble of decision trees and makes predictions based on the predictions of the set of trees in the forest. Random forest models use 200 trees. Research has shown that using more trees in a random forest is beneficial [26]. However, using more trees in the forest became computationally expensive and therefore the largest feasible number of 200 trees is chosen. The logistic regression and random forest classifiers are chosen in order to get both a linear and non-linear approximations of the prediction tasks. The k-nearest neighbour classifier was added in order to use a distance based model.

Models are implemented using Python 3.7.0 (Python Software Foundation, https://www.python.org/) and the SciKit learn package [27]. For each model, hyperparameters are selected using a grid search with 10-fold cross validation. A set of possible hyperparameters is supplied and for each combination of hyperparameters a model is trained. Each model is trained using the training data and cross validation is applied. A custom cross validation function is used to make sure that each fold contains repair products originating from all target sites and each genotype. A full list of hyperparameters tested during the grid search can be found in supplementary materials C. After the grid search, the best performing model with the best performing hyperparameters is used as the final model. This process is applied for each created model. It should be mentioned that using cross-validation for hyperparameter selection results in biased performance estimates [28, 29]. To eliminate this bias a computationally more expensive nested cross-validation loop is recommended. However, strong evidence exists that models with few hyperparameters normal cross-validation shows equal performance with its nested variant [30]. With the used learning algorithms having few hyperparameters, no nested cross-validation is used.

The evaluation metric used to select the best hyperparameters for each model during cross-validation is the f1-score for the first predictive task. For the second predictive task, the macro-averaged f1-score is used. Different scoring metrics are used since the normal f1-score cannot be used for non-binary classification tasks. The first predictive tasks, prediction of genotype in which a repair product is formed, is binary. These models only predicts one of the Ku80-/- and PolQ-/- genotypes. The second predictive task, which predicts the genotype of a cell culture, also predicts the WT genotype and is a non-binary classification task. Therefore the normal f1-score cannot be used. Macro-averaged f1-score is arbitrarily chosen over its micro-averaged counter, since Each class has an equal contribution in the data used for the second predictive task.

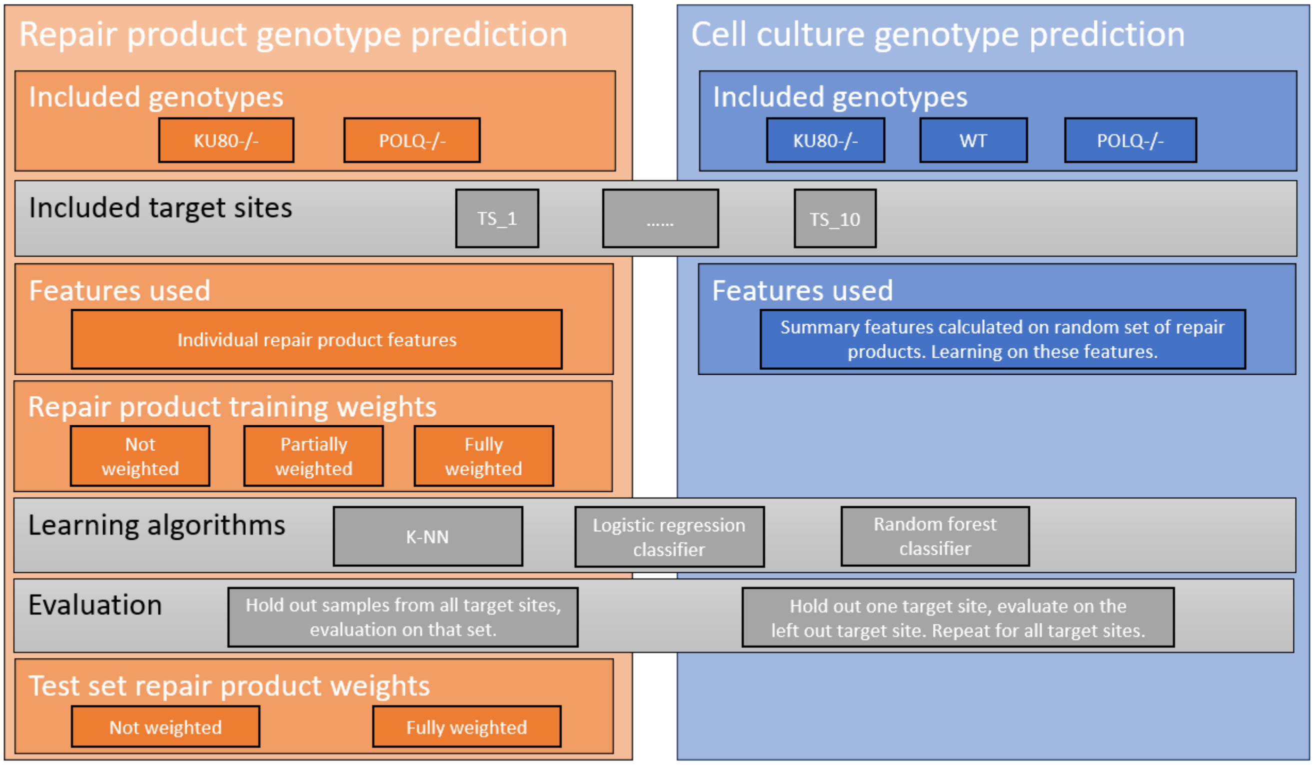

Repair product genotype prediction
Cell culture genotype prediction
Included genotypes
KU80-/-
POLQ-/-
Included genotypes
KU80-/-
WT
POLQ-/-
Included target sites
TS_1
......
TS_10
Features used
Individual repair product features
Features used
Summary features calculated on random set of repair products. Learning on these features.
Repair product training weights
Not weighted
Partially weighted
Fully weighted
Learning algorithms
K-NN
Logistic regression classifier
Random forest classifier
Evaluation
Hold out samples from all target sites, evaluation on that set.
Hold out one target site, evaluate on the left out target site. Repeat for all target sites.
Test set repair product weights
Not weighted
Fully weighted

Fig. 2. Setup of the models build for the predictive tasks. The orange column shows all variables for predicting the genotype in which a repair product can be found. The blue column shows the variables of the cell culture genotype prediction task. Gray rows indicate that variables are shared between predictive tasks. Additional target site information can be found in the supplementary materials A.

2.3 Evaluation and metrics

Two predictive tasks are tested in this work. The first predictive task is a binary prediction where the genotype in which a repair product is found is predicted. Both predictive tasks will be evaluated using hold-out test sets. With the first task being a binary problem, ROC curves and the area under the curve (AUC) will be reported. The second predictive tasks for cell culture genotype prediction will be evaluated using a generalization of the AUC for multinomial classification [31]. Scikit-learn has this generalization build in the roc_auc_score, by setting the 'multi_class' option to 'ovo'.

For both predictive tasks, a total of 11 models are trained for each learning algorithm. The first model will be trained using repair products originating from all target sites. The hold-out test set will contain independent repair products, also originating from all target sites. This should show how well models can learn the problem on seen target sites. Secondly models will be trained with all repair products from 9 target sites. The repair products originating from the tenth target sites will be used to evaluate the model. A total of ten models will be trained, so that each target site is evaluated once without being present in the training data. The evaluation of these models should show how well models can generalize knowledge on new target sites, which were not present in the training data. For each of the 11 models per learning algorithm, new hyperparameters are selected.

2.3.1 Setup: Repair product genotype prediction

The goal of this prediction task is to predict the genotype in which a repair product can be found. Whether this is possible is tested with the several predictive models. All variables for the predictive task are shown in Figure 2, orange column.

Included genotypes - repair products originating from the Ku80-/- and PolQ-/- genotypes will be used. The wild type genotype will be excluded, reasoning described in subsection 2.2.

Training sample weights - as mentioned in subsection 2.1, the dataset consists of unique repair products accompanied by a frequency showing how often they occurred in a cell culture. Repair products with a higher frequency are more likely to be formed during the repair of a DSB at the target site. Repair products with a higher frequency can therefore be seen as more important. This information can be used to train the model. This is done using sample weights on the training data. For this predictive task, models are trained using one of three different sample weights. Each different sample weight introduces a bias towards certain repair products during training. For each different sample weight method applied, a models hyperparameters are recalculated.

First, an equally weighted training method is tested. This method aims to have repair products originating from the same cell culture have an equal weight. This means that the frequency of repair products is disregarded and a bias towards infrequent repair products is introduced. For a cell culture with DSB induced a given target site, the sample weight of each repair product is calculated as following:

$$w = \frac{1}{primer\,sets * count(products)}$$

In this function w is the sample weight, $primer\,sets$ are the number of primer sets used to sequence the target site and $count(products)$ is the number of repair products in the cell culture. First of all, due to $count(products)$ the sample weight of each repair product originating from the same cell culture will be equal. Secondly, by using $primer\,sets$ all target sites have an equal contribution. Depending on the number of primer sets used, each target site is represented by one or two cell cultures with the Ku80-/- genotype and one or two cultures with the PolQ-/- genotype (supplementary materials B). Thus although target sites TS_9 and TS_10 are represented by twice the cell cultures, the sample weights for repair products from those cultures will be halved. Removing the $primer\,sets$ term from the equation would make models biased towards TS_9 and TS_10, since more data is available for these target sites.

The second method used for training sample weights is the fully weighted method. This method aims to give frequently occurring repair products a sample weight representing its frequency. This introduces a bias towards the most frequent repair products. For each repair product, the sample weight is calculated as:

$$w = \frac{frequency}{primer\,sets}$$

Here, the frequency is the frequency reported in the dataset which has a value between 0 and 1. Again, $primer\,sets$ is used to eliminate bias towards target sites represented by more cell cultures.

Finally, a partial weighting method is used. This method tries to reduce the bias towards frequent or infrequent repair products as seen by the other two weighting methods. To do this the following function is used:

$$w = \frac{\log_2((frequency * 100) + 1)^2}{primer\,sets}$$

Again $primer\,sets$ eliminates bias towards a target site. The $frequency$ is multiplied by 100 and 1 is added so that a positive number larger than 1 will be used when calculating the $\log_2$. This logarithmic function is used so that the sample weight does not scale linearly with the frequency. Finally this value is squared to slightly increase the difference in sample weights. The final weight for a repair product with a frequency between $0 < frequency <= 1$, will be $0 < frequency <\approx 44.33$ divided by $primer\,sets$.

These three different methods of setting training weights all favor different repair products. Equal weights is biased to infrequent repair products, while fully weighted is biased to frequent repair products. Partial weights reduces the bias towards both. It should be noted that the k-nearest neighbour classifier does not support sample weights in Sci-Kit learn. As an alternative, repair products in the train data are duplicated based on the assigned sample weight. Although has the same effect as using sample weights, it increases the computational costs.

Preprocessing - in order for models to properly handle the data, several preprocessing steps are performed;

1. Removal of the WT genotype cell cultures
2. Removal of error-free repair products
3. Recalculation of the frequency
4. Set repair products labels
5. Adding and encoding features and data imputation
6. Create a test set
7. Feature scaling

1: any cell culture with the WT genotype is removed from the data. The reason for WT removal is explained in subsection 2.2.

2: error-free repair products are removed. Error-free repair products are those without genetic alterations. Error-free repair products do not show any useful information of repair pathway deficiency. Therefore it was decided to remove these from the train data for this predictive task.

3: after removal of error-free repair products the sum of repair products frequency for each cell culture no longer equals 1. Since the frequencies are used to set training weights, the summed frequency of all repair products in for each cell culture should be equal. Therefore, frequencies are recalculated by normalizing the values per cell culture, so that the total frequency per cell culture is again 1.

4: the label of a repair product tells machine learning algorithms what should be predicted when seeing the data. For this predictive task, the used labels are either Ku80-/- or PolQ-/-. For each repair product, the label is set to the genotype of the cell culture a repair product originates from. In some cases, the same repair product can be found in both the Ku80-/- and PolQ-/- genotypes. Often in these cases, the frequency is much higher in one genotype than the other. Thus a repair product which can be seen in both genotypes, is more likely to be seen in one of them. Models using the partial or fully weighted sample weights be able to learn to predict the more likely label in these cases. Disregarding the training weight method used, repair products seen in multiple genotypes should act as noise. This should help make models more robust.

5: information from the dataset are used as features for machine learning algorithms. The dataset contains several non-numeric features, categorical features and features containing null-values. Non-numeric features have been removed, categorical features are one-hot encoded. The logistic regression classifier cannot handle null-values, thus these are imputed with 0. Null-values in the dataset show something did not happen. E.g. a homologyLength with a null-value means no microhomology took place. Therefore, values are imputed with 0 rather than other frequently used methods. A full list of used features can be found in supplementary materials A.1.

6: a hold-out test set is created. As mentioned in subsection 2.3, different models are created to evaluate on known and new target sites. For models which are trained using nine out of ten target sites, all repair products from the excluded target site are used as the hold-out test set. For evaluation on known target sites, a hold-out test set is created. This test set must meet the following criteria. First, the test set must contain repair products from each cell culture. This ensures that the test set contains repair products from each target site and all genotypes. Secondly, the test set must contain both infrequent and frequently occurring repair products from each culture. With only a small percentage of repair products having a high frequency, it should be ensured that these are found in both the training and test set. Finally, repair products in the test set should be unique from those in the train set. As mentioned above, the same repair products can be found in cell cultures of both the Ku80-/- and PolQ-/- genotypes. Repair products from different cell cultures are considered equal when they are found at the same target site and all feature values are equal (see supplementary materials A.1. for used features). If a repair product is found in both the Ku80-/- and PolQ-/- genotype, both their instances should be placed in either the training or the test set. To reach these three criteria, a custom function is used. First, repair products are split into two groups. The first group contains unique repair products, found only in a single genotype. The second group contains repair products found in both the Ku80-/- and PolQ-/- genotypes. Both groups are sorted on cell culture and secondary sorted on the frequency of repair products. For the first group, every 5th repair product in the list is selected for the test set. Due to the list sorting, the test set will receive a fair amount of high frequently found repair products of each culture. For the second group every 5th repair product is selected as well. This time however, we know there is a duplicate repair product found in another genotype. The two equal repair products found in different genotypes genotypes are both added to the test set. Furthermore, they are both removed from the list. This ensures that the test set is unique from the training set. Due to the sorting of both lists, all cell cultures are found in both the train and test set and both frequent and infrequent repair products should are present in both sets. By selecting every fifth repair product the test set contains approximately 20% of all repair products. Keep in mind that this test set is only used for models where all target sites are used during training. For models trained on all target sites, a single hold-out test set is created. This way different models can be evaluated using the same test set, ensuring a fairer comparison between classifiers.

7: the final preprocessing step applies a feature scaling. Feature scaling is used to reduce the range of feature values. This procedure can help improve a models ability to generalize and therefore improving performance. Numeric features are standardized, which transforms the values so that the mean of the feature becomes 0 and has unit-variance. Other methods such as MinMax scaling, normalization and log-transformation have been tried but performed worse.

Test sample weights - all models are evaluated using a hold-out test set. With similar reasoning as using training sample weights, repair products in the test set can be weighted as well. Therefore, each model will be evaluated twice with different test set weights applied. The first method is equal to the equally weighted method used for training. Within each cell culture, repair products will be weighted equally. Secondly, evaluation will be done using full weights, the same as its fully weighted training weights counterpart. This method will use the frequency of a repair product as sample weight. Keep in mind that for both the equally and fully weighted evaluation, the hold-out test set does not change. Thus any increase or decrease in performance is directly linked to the used sample weights.

2.3.2 Setup: Cell culture genotype prediction

The goal of this prediction task is to predict the genotype of a cell culture using a set of repair products from that cell culture. To make predictions, the set of repair products are described using summary features. These summary features are used to train models. The extend to which models are able to predict a cell cultures genotype is tested with models described below. The variables describing this predictive task are shown in Figure 2, the blue column.

Included genotypes - models created for this predictive task will use repair products originating from each of the Ku80-/-, PolQ-/- and wild type genotypes. This is shortly discussed in subsection 2.2

Preprocessing - for this predictive task the following preprocessing steps are performed;

1. Create copies of repair products based on their frequency
2. Distribute copies over groups, representing cell cultures
3. Label groups
4. For each group, calculate summary features
5. Create a test set
6. Set training and test sample weights
7. Feature scaling

1: the dataset contains repair products originating from 36 different cell cultures. To properly train classifiers, having more cell cultures in the training and test data would be preferable. In this step, groups are created, each representing a cell culture. This way, more cell cultures can be used for training and evaluating models. Groups are created using the repair products frequency. The first step is to create copies of repair products based on the frequency of the repair product. For each repair product, the frequency is multiplied by a fixed value. This value is named the duplicationFactor. Three different duplicationFactors are used, 1.000, 10.000 or 100.000. If the multiplication results in a decimal number, the number is rounded up. Thus, for a repair product with an frequency of 0.00015 a total of 1, 2 or 15 copies will be created, respective to the three duplicationFactors. Rounding up is done to make sure that repair products with a low frequency are copied at least once. Secondly, the duplicationFactor introduced a bias towards repair products with a lower frequency. As seen in the above example, a repair product with a frequency of 0.00015 will have a single copy made when using a duplicationFactor of 1.000. In contrast, a repair product with a frequency of 0.015 will have 15 copies with the same duplicationFactor. While the frequency is 100 times larger, only 15 times as many copies will be created. This introduces a bias to infrequent repair products for lower duplicationFactors.

2: now the copies of repair products are distributed over groups. Per cell culture, all copies are randomly distributed over 200 groups. Each group therefore contains copies of repair products originating from a single cell culture. The first two steps of copying and distributing repair products over groups is fairly similar to sampling without replacement, where the chance of sampling equals a repair products frequency. However, there is one key difference. In the first step a bias is introduced towards infrequent repair products using the duplicationFactor. This can act beneficial. With all 200 groups being sampled with repair products from a single cell culture, groups are likely to have a similar repair products in a group. With the introduced bias, groups are more likely to contain infrequent repair products than without an introduced bias. These infrequent repair products being included more result in more different repair products being present in a group, thus increasing the variance in repair products. This increase could contribute in making models more robust. Robustness is especially preferred when models are evaluated on a left-out target sites. Furthermore it should be noted that all data is distributed over the groups. With higher duplicationFactors creating more copies, this also means that groups created with higher duplicationFactors will contain more repair products than groups created with a lower factor.

3: groups need to be labeled with Ku80-/-, PolQ-/- or WT in order for models known what they generalize on. Since all repair products in a group originate from the same cell culture, a group will be labeled with the genotype of that cell culture.

4: with each group having a number of repair products in it, they must be described in a way so that the machine learning algorithms are able to handle them. For this work, summary features are used to describe all repair products of a group. These features can be found in supplementary materials A.2, and mainly consist of the mean values and standard deviations of features used in the first predictive task. For example, repair products have a feature deletionLength which describes the number of deleted nucleotides in a repair product. Using a feature as AVGdeletionLength will be used to describe each group. It is the average deltionLength calculated over all repair products in the group. Furthermore, the deviation in the deletionLength of all repair products will be reported as STDdeletionLength. All summary features are calculated using the NumPy library in Python.

5: the data is now ready to be split into a training and test set. Again, models will be created for evaluation on seen and unseen target sites. For the model trained on seen target sites, it is important to include groups containing repair products from each target site. As described above repair products from a single cell culture are used to create 200 groups, each described by summary features. Of each cell culture, a random 20% of the created groups will be selected for the test set. This makes sure that all target sites and all genotypes are equally represented in the test set. Secondly, ten models are trained using all groups from nine of the ten target sites. The all groups with repair products originating from the left-out target site are used as the hold-out test set.

6: sample weights in the training and test set will again be used to eliminate bias towards target sites. With target site TS_9 and TS_10 being sequenced with two primers (supplementary materials B), more cell cultures exists for them. Twice as many groups will be created for these target sites compared to other target sites. In order for models to not be biased to these target sites, the weights of groups summarizing repair products of these two target sites will be set to 0.5. Groups summarizing repair products from other target sites will receive a sample weight of 1. This makes sure that all target sites are equally represented. Furthermore, when looking at all groups describing a single target site each group has an equal weight.

7: similarly as in the previous previous experiment, features will be scaled in order for machine learning algorithms to better use them. This time, a MinMaxScaler($[0, 1]$) is used to scale all features to have a minimum value of 0 and a maximum value of 1. The transformation is learned on the training data and applied on the test data.

3 Results and discussion

In this section, the the results of the models built for each of the two proposed prediction tasks are reported. First of all, prediction of genotype in which a repair product is formed is reported and discussed. Secondly, results of cell culture genotype prediction will be shown. As mentioned in subsection 2.3, models are evaluated either using repair products originating from all target sites or repair products from a target site left out from the train data.

3.1 Repair product genotype prediction

For this prediction task, models are trained to predict the repair-deficient genotype which produces a repair product. A prediction of the PolQ-/- and Ku80-/- genotypes is made, which shows in which genotype the repair product is created. most likely found in that genotype. First results of classifiers trained using all target sites are discussed. Secondly, results of models trained on all but one target site are discussed.

3.1.1 Evaluation: all target sites

Figure 3 shows ROC curves for each of the classifiers. In the upper row, the test set is equally weighted. In the lower row, the test set is weighted using repair products frequency. Results seem to be modest. The AUCs range between 0.574 and 0.699 for equal weighted test set and between 0.584 and

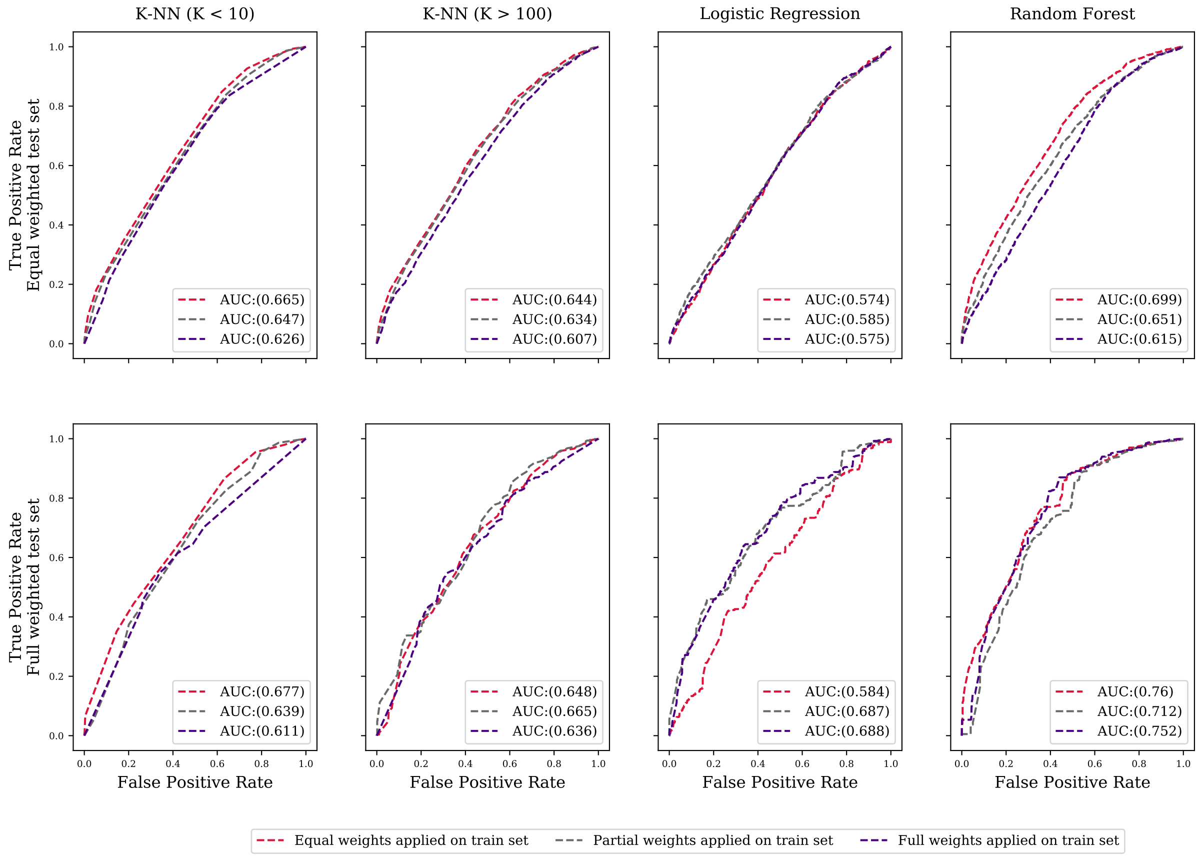
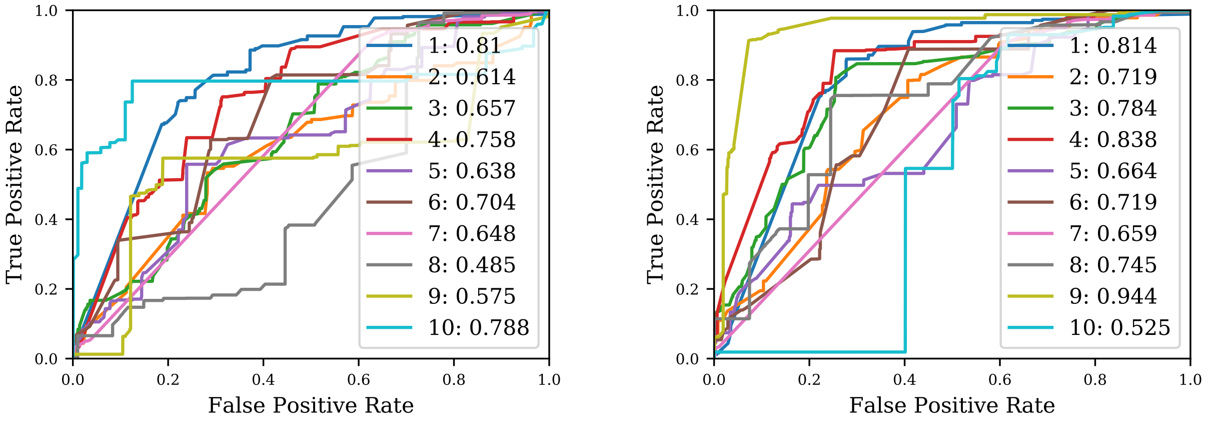


Fig. 3. Receiver operator characteristic curves of each models trained. During training, products from all target sites were present. Equal, partial and full train weights are indicated by the different curves. The area under the curve can be seen in the legend. The upper row shows results when test data is equally weighted, the lower shows results on a full weighted test data. Keep in mind that the upper and lower row are evaluated on the same data, only the sample weights of the data in the test set is different.

0.76 for the full weighted test set. Although these performances are better than a random guessing method, models are far from optimal.

Comparing the two the equally weighted and fully weighted test set evaluation (Figure 3, upper vs lower row), a minor increase in the AUCs is seen. This increase is seen across all classifiers and for all training weight methods. For the logistic regression and random forest classifiers, the largest increase in AUC is seen when using the fully weighted training samples method. Each model is being evaluated twice on the same hold-out set, with the first using equal test sample weights (upper row) and then using a full weighted test sample weights (lower row). Models and the test set do not change for these two evaluations, just the test set sample weights. This means that the increase in AUC for each model is directly linked to the different test set sample weights used. An increase of AUC would only be possible if higher weighted test data is predicted correctly more often than those with low weights. Since the fully weighted test set evaluation uses repair products frequency as sample weight, this means models are more likely to predict repair products with a higher frequency correctly. Thus models are able to generalize better to more frequent repair products.

Comparing the different training sample weights used, shown by the different colored graphs in Figure 3, only minor differences in AUCs can be seen. With the K-NN classifiers, almost no difference in the AUC of the different training weights can be seen. Only when evaluation is done using a weighted test set for the logistic regression model, a slightly clearer difference in AUCs can be seen. With the different training sample weights used, biases are introduced to either frequent or infrequent occurring repair products. One possibility for the minor difference is that infrequent repair products contain irrelevant information, also known as noise. Models can learn on this noise and when use it to generalize models become over fitted. Over fitting can be seen by comparing the training error and performance on the test set. Over fitting is a possible explanation for the minor gain in



(a) Logistic regression model

(b) Random forest model

Fig. 4. ROC per target site of logistic regression (a) and random forest (b) models trained on **all** target sites using partial train weights. Evaluation is done using weighted test data. The number 1 till 10 in the legend indicates the target site used for evaluation, the value behind it is the AUC achieved on the target site.

AUC from using different training sample weights. A comment is made on over fitting of these models in supplementary materials D.

When looking at the three learning algorithms, the random forest algorithms performs best. The other classifiers perform fairly similar. Interestingly, for the K-NN models the number of neighbours chosen has a very little effect on the models performance. Computationally speaking, the K-NN classifiers are a bad choice. The K-NN models are time consuming to train. Remember that K-NN models do not support sample weights and repair products in the dataset are duplicated instead. This means that K-NN models need to handle much more data increasing the computation time significantly. Furthermore their results are comparable to those of the logistic regression classifier. The random forest classifier seems to be the best option.

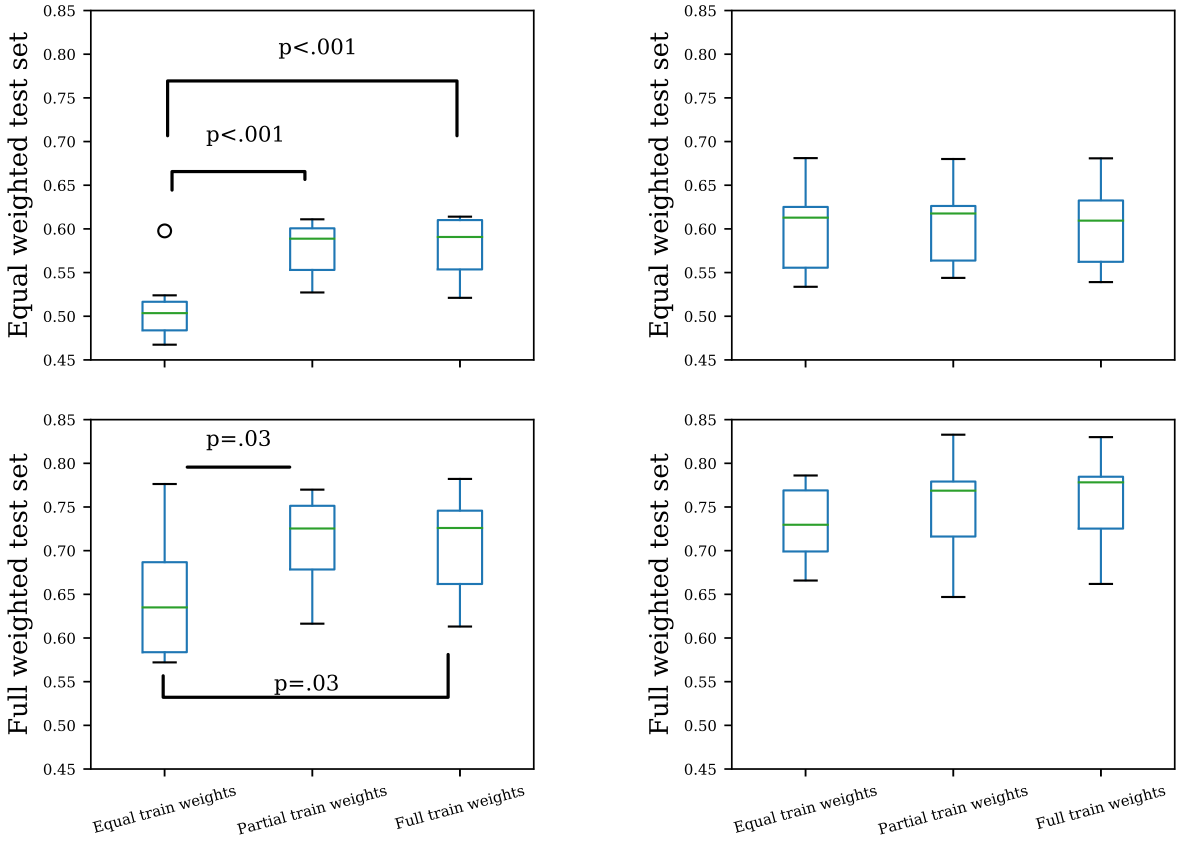
The results in Figure 3 show the performance of models trained using repair products originating from all ten target sites. The evaluation is also done using repair products originating from all ten target sites. For both logistic regression and random forest classifiers, the model trained using partial weights is arbitrarily chosen. These two models are then evaluated with the hold-out test set, however all repair products from each unique target site are evaluated separately. This results in ten ROC curves achieved by the model, each one showing the curve obtained at for a specific target site. For the logistic regression model, these curves can be seen in Figure 4a. The ROC curves of the random forest classifier per target site are shown in Figure 4b. Interestingly, the variance in the AUC achieved on different target sites is quite large. The logistic classifier (Figure 4a) shows an AUC of 0.485 on TS_8, while TS_1 has an AUC of 0.81. There is also a high variance in performance per target site for the random forest classifier (Figure 4b), with TS_10 achieving an AUC of 0.525 and TS_9 achieving an AUC of 0.944. This variance is likely due to models being able to generalize better to certain target sites. This would indicate that repair products from some target sites show similar patterns. Likely, repair products from for example TS_4 show similar characteristics as those from a few other target sites. This makes the performance of that particular target site really consistent across different classifiers. The variance over the performance of the different target sites is then directly correlated with how many similar behaving target sites there are in the dataset.

Interestingly, on target site TS_8, TS_9 and TS_10 show very different performances are reported when comparing the logistic regression model (Figure 4a) and the random forest model (Figure 4b). A possible explanation of this difference can be the difference in the models. Logistic regression is a linear learning algorithm, while random forest is non-linear. A possible explanation is that repair products originating from certain target sites can be generalized better by either linear or non-linear models. However, the difference is most likely due to the over fitting seen in models (supplementary materials D).

3.1.2 Evaluation: leave target site out

This section evaluates repair product prediction where models are evaluated on a target site left out from the train data. This gives insights in the generalization across new target sites. Again, models are trained using different training sample weights. This time however, only the logistic regression and random forest models are reported. Due to the computational cost of K-NN and it not showing any significant difference in the ROC curves compared to logistic regression (Figure 3), no K-NN models were trained.

A new model is trained to evaluate cell cultures from each unique target site. This results in 10 logistic regression and 10 random forest classifiers, each trained with repair products originating from all but one target site. The different AUCs obtained by the logistic regression models are



(a) Logistic regression model

(b) Random forest model

Fig. 5. Bloxplot showing the AUC of 10 logistic regression (a) and 10 random forest (b) models. Each model is trained on 9 target sites and evaluated on a single. The model is then evaluated using all repair products originating from the left out target site. In the upper row, repair products originating from the evaluated target site are equally weighted. In the lower row, they are weighted. Significance between training weight methods is only shown when $P<=.05$

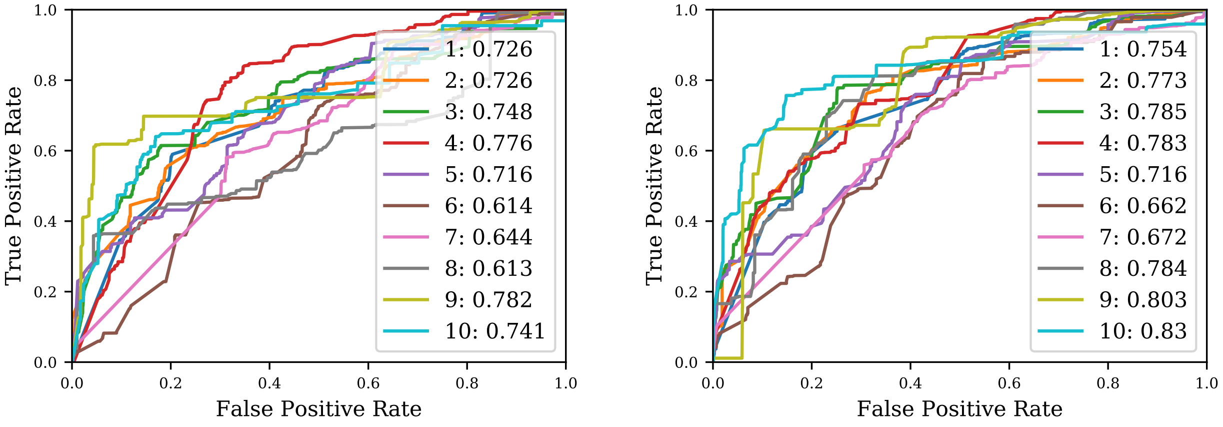
shown in Figure 5a, the AUCs from random forest models can be found in Figure 5b. A two-tailed U-test is performed between the different training sample weight methods, using the AUCs as values.

When comparing the different weight samples of the test set (upper and lower row of Figure 5) a clear increase is seen in the average AUCs achieved. This is the case for both logistic regression and random forest classifiers, across all train sample weight methods. As the full weighted test evaluation uses the frequency of repair products, it is shown again that models perform best on the more frequent repair products. This is in compliance with results conclusions made in subsubsection 3.1.1.

Comparing the different train sample weights used, logistic regression shows a clear and significant differences. Significance is obtained between equal and partial train weights, and obtained between equal and full train weights. Thus when models are to make predictions for new target sites, models can generalize better when the emphasis is put on more frequent repair products in the train data. By giving these higher weights, a performance increase is seen for logistic regression. This performance gain is not seen with significance for the random forest models. However, for logistic regression applying full or equal sample weights is definitively preferred.

For models created with the partial training sample weights, each ROC curves per classifier are shown in Figure 6a for logistic regression and in Figure 6b for random forest classifiers. These figures can be used to compare the ROC curves achieved on different classifiers shown in Figure 4. Keep in mind that Figure 4 shows models trained on all target sites. The variance in AUCs per target site is Figure 6 is smaller than that of Figure 4. More interestingly, the target sites having achieving performances for logistic regression and random forest models in Figure 4 (TS_8, TS_9 and TS_10), perform much more consistent in Figure 5. Most likely, varying results per target site in Figure 4 are due to the overfitting.

From comparing Figure 4 and Figure 6 it can be argued that models achieve better results when making predictions



(a) Logistic regression model

(b) Random forest model

Fig. 6. ROC per target site for logistic regression (a) and random forest (b) models. These models are trained using repair products from nine out of ten target sites using partial training weights. Evaluation is done using weighted test data. The number 1 till 10 in the legend indicates the target site used for evaluation, the value behind it is the AUC achieved on the target site.

on repair products from a left out target sites. When predicting on unseen target sites (Figure 4), models behave more consistent across different target sites. An explanation for the more consistent predictions can be due to the repair products used during training. Models trained on all target sites exclude 20% of repair products originating from each target site for the test set. Thus each target site is missing 20% of its repair products in the training data. On the other hand, models trained using nine out of ten target sites can use all repair products originating from target sites in the training data. Models seem to become more robust when all data of a target site can be used, rather than removing some for evaluation. This can explain why results in Figure 6 are much more consistent than those in Figure 4.

3.2 Cell culture genotype prediction

This section reports the results of models trained to predict the genotype of a cell culture. To make predictions, repair products are copied by a number of times depending on the duplicationFactor. A lower duplicationFactor increases a bias towards infrequent repair products. These are distributed over multiple groups, simulating a cell culture. A group is described with summary features, which are used by models to make predictions.

3.2.1 Evaluation: all target sites

Models trained to predict the genotypes of a cell culture use summary features of repair products. 20% of the groups are used to evaluate the models, and both the training and test data contain groups containing repair products from all target sites. Figure 7 shows the AUC of the ROC curve per classifier trained with duplicationFactors. As can be seen, models performs increases as the duplicationFactor increases, up to near perfection. It is clearly indicated that the repair-deficiency of a cell cultures can be predicted. Keep in mind that this stands for making predictions on cell cultures with DSBs induced at a target site which was present in the training data.

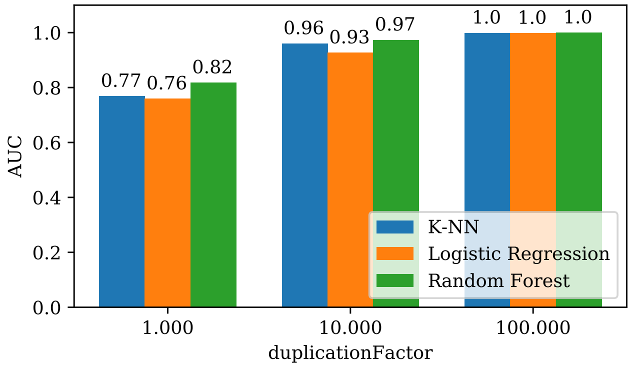


Fig. 7. The area under the ROC curve (AUC) achieved per learning algorithm and duplicationFactor. Numbers above each bar show the AUC value rounded to 2 decimals. A generalization of the AUC for multinomial predictions is shown [31].

When comparing the different classifiers used for cell culture genotype prediction, it can be seen in Figure 7 that models show extremely similar AUCs. While the random forest classifier seems to work best with a duplicationFactor of 1.000, the difference is not significant.

For the random forest classifier the test set is evaluated per target site. The model is trained using cell cultures with breaks induced at all target sites, but evaluation is done by calculating the AUC per unique target site in the test set. The AUC per target site can be seen in Figure 8. For the duplicationFactor 1.000, the AUC ranges from approximately 0.68 to 0.95. While the performance differs per target site,

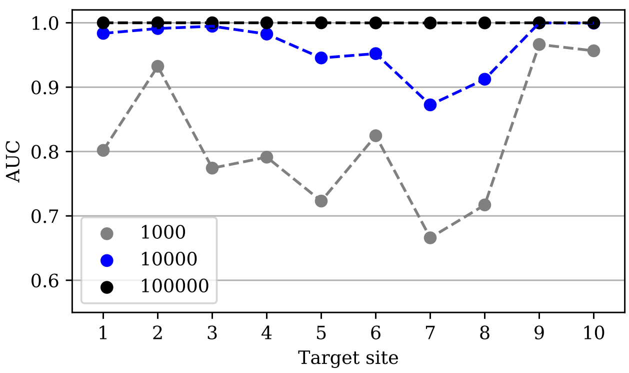


Fig. 8. Accuracy per target site achieved on the test set for the random forest classifier. The model is trained with cell cultures with breaks on all target sites. Note that the points depict the AUC achieved on a target site. Lines connecting two points are added for readability of overlapping points and do not show information about performance.

increasing the duplicationFactor reduces this variance. The larger variance in the AUC for the 1.000 duplication factor can be due to the bias. Whether this improves the ability to generalize cell cultures with DSBs at different target sites, can be better seen in the leaving one target site out evaluation.

3.2.2 Evaluation: leave target site out

Models were trained using cell cultures with breaks induced target sites. Here models were trained with cell cultures from nine out of the ten target sites and evaluated on the left out target site. AUCs achieved for these models are shown in Table 3.

At first sight, it is clear that models are able to predict across unseen target sites well. For example, when looking at the duplicationFactor 100.000 for the random forest classifier, the average AUC per target site achieved is 0.879. The K-NN classifier seems to be outperformed by the logistic regression and random forest classifiers.

When comparing the different duplicationFactors, it becomes clear that introducing a bias towards infrequent repair products has both advantages and disadvantages. The advantage is seen well for cell cultures with DSBs induced at TS_1. Increasing the duplicationFactor removes a bias towards infrequent repair products, resulting in groups having a distribution fitting more to real data. The summary features describing TS_1 seem to follow the patterns learned by models on other target sites. Thus having no bias towards infrequent products for a duplicationFactor of 1.000 results in models recognizing the target site well, even though models have never seen any cell cultures of that target site. On the other hand, results obtained at TS_3 suffer from a higher duplicationFactor. When comparing results obtained with duplicationFactor 1.000 with 100.000 at target site TS_3, it can be seen that the AUCs for K-NN and logistic regression dropped significantly. The random forest classifier is an exception here. Likely, the duplicationFactor introducing a bias towards infrequent repair products has a positive effect for the K-NN and logistic regression classifiers. With a bias introduced, groups show a slightly more variance in repair products. This helps the models make more robust, at least when evaluating TS_3. Interestingly, at TS_3 with a duplicationFactor of 10.000 logistic regression performs better than with a factor of 1.000 or 100.000. This effect is also seen for logistic regression at target sites TS_9 and TS_10.

Finally, it should be noted that the logistic regression and random forest can show widely varying AUCs on target sites. For example at TS_3 and TS_10 with a duplicationFactor of 100.000, the random forest clearly outperforms the logistic regression classifier. On the other hand, logistic regression achieves substantially better results at TS_1 and TS_7. Both seem to predict some target sites much worse than other target sites. For logistic regression predicting the genotype of cell cultures with DSBs induced at TS_3 and TS_10 is difficult, with AUCs of 0.513 and 0.665 respectively. For random forests TS_7 is more problematic, with an AUC of 0.656. Why these models have more trouble with different target sites, might be due to both models learning slightly different information. For example, if one model weights the average deletionLength higher, while the other finds the average microhomologyLength more important results can differ per target site. While repair products from cell cultures can be generalized and a genotype can be predicted, the manner in which a model does this can differ. This would mean that repair products from a target site show similar patterns with products from some other target sites. However, repair products from some target sites behave much less similar meaning repair pathways show slightly differing patterns in their repair products per target site.

4 Conclusion

This section describes the most important findings of both experiments. After the conclusions, some limitations are discussed. Finally recommendations for future work are made.

4.1 Predicting from single repair products

Two methods are used to evaluate models. First models are trained with with repair products originating from all target sites. Secondly, models are trained where a single target site is excluded from the training data on which the model is evaluated. Overall, results seem promising and a clear indication is made that statistical models are able to predict repair-deficient genotypes from Cas9-induced repair products.

For the models trained on data of all target sites, results are modest. Classifiers seem to have trouble generalizing the information of single repair products.The best model, a random forest classifier, achieves an AUC of 0.76. Other models

		Target site (TS) excluded from train data and used to evaluate model										
Duplication Factor	Classifier	TS_1	TS_2	TS_3	TS_4	TS_5	TS_6	TS_7	TS_8	TS_9	TS_10	AVG
1.000	K-NN	0.664	0.770	0.592	0.656	0.600	0.625	0.593	0.623	0.660	0.586	0.637
	Logistic	0.685	0.836	0.605	0.667	0.624	0.614	0.609	0.620	0.850	0.663	0.677
	Forest	0.673	0.810	0.621	0.712	0.651	0.671	0.639	0.648	0.820	0.743	0.699
10.000	K-NN	0.852	0.923	0.540	0.919	0.876	0.761	0.659	0.793	0.702	0.698	0.772
	Logistic	0.881	0.936	0.683	0.943	0.879	0.731	0.733	0.812	0.881	0.756	0.824
	Forest	0.927	0.926	0.660	0.943	0.900	0.754	0.746	0.861	0.902	0.814	0.843
100.000	K-NN	0.899	0.877	0.500	0.982	0.959	0.796	0.658	0.719	0.710	0.719	0.782
	Logistic	0.973	0.954	0.513	0.974	0.990	0.830	0.816	0.936	0.813	0.665	0.846
	Forest	0.922	0.999	0.735	1.00*	0.994	0.793	0.656	0.935	0.795	0.958	0.879

Table 3. AUC scores for models trained on all but one target site. Site excluded is shown in the column, the AUC score is from evaluation on the excluded target site. Scores are rounded to three decimals. *No AUC of 1, but shows so due to rounding.

trained achieve lower AUCs when trying to predict the genotype in which a repair product is created. However, results indicate that frequently occurring repair products are predicted correctly more frequently. When evaluating a model on an unseen target site, the AUCs achieved range between 0.662 and 0.803, depending on the target site evaluated. Although the results have a lot of room for improvement, there is a clear indication that statistical learning algorithms can be used to generalize repair products found in repair-deficient genotypes. When looking at a single repair product, models are able to predict the genotype in which it is found with reasonable accuracy. More data created in other repair-deficient genotypes might show if this can be extended.

4.2 Predicting cell cultures

When looking at a group of repair products from a single cell culture, the repair-deficient genotype can be predicted well. A generalized AUC for multinomial classifiers shows that models are able to predict a cell cultures genotype on seen target sites. When evaluation is done on cell cultures with induced DSBs on target sites which were also present in the training data, models are able to predict genotype with near perfection. This does depend on the duplicationFactor used. This does however indicate that cell cultures can be described well with summary features. When evaluation is done on target sites left out of the training data, results differ per target site. The best performing model, a random forest classifier obtained an average AUC of 0.879 across the different target sites. Summary features describing repair products can be generalized across target sites, but performance differs a lot per target site. Furthermore, introducing a slight bias towards infrequent repair products is beneficial for the performance on some target sites. These results indicate that it is possible to predict one of the Ku80-/-, PolQ-/- and WT genotypes for a cell cultures, using Cas9-induced repair products.

4.3 Limitations

Model thresholds - for both predictive tasks, models are trained with repair products originating from nine of the ten target sites. Evaluation is done using an AUC of the ROC curve. The AUC shows the true positive rate against the false positive rate of models at various thresholds. For evaluation of different target sites, different thresholds might be optimal. No model has been trained with repair products from less target sites, making evaluation on multiple unseen target sites possible. If optimal thresholds would differ significantly per target site, AUC scores for evaluation on multiple unseen target sites are likely lower than those presented in this work. AUCs presented in this thesis therefore might be slightly higher than when models are evaluated on multiple target sites which were not present in the training data.

Test set creation - as mentioned in subsubsection 2.3.1, repair product genotype prediction uses a custom function to select a test set with repair products from all target sites with a wide variety in the frequency. This function selects every fifth repair product in sorted lists to create a test set of 20%. This method removes randomness from the selection of the test set. Including randomness when selecting test data is good practice, and therefore this can be seen as a limitation. A solution would be to select one in every five repair products randomly from the sorted list, rather than the fifth. This increases the randomness in selection of the test set.

5 Future work

This section describes three points options for future work. The first point goes into how the presented models can be used to predict repair pathways, rather then a repair-deficient genotypes. Secondly, some recommendations are made for predicting different target sites. Finally, an alternative method was tried for prediction which might be improved.

5.1 Repair pathway research

This thesis was performed to assist the LUMC with researching theta-mediated end-joining. The goal was to get new insights in the repair mechanism using the repair products and statistical models. Ideally, models would be able to predict by which repair pathway DSBs are repaired, by looking at repair products. This would work likely best for cell cultures, where looking at a group of repair products it can be said which pathway was active or multiple in a WT genotype. Such a model like could be used to find genes related to repair pathways. This would be done by supplying a genotype with unknown relation to a repair pathway, and see which pathway activity is predicted by the model. However with different gene deficiencies, where genes relate to repair-pathways, might result in other distribution of repair products. To train a classifier able to properly predict active pathways using repair products, more data is needed. Specifically repair product data, where product are created in other genotypes would be needed.

5.2 Clustering target sites

In both predictive tasks, evaluation on a left out target site showed variance in the AUC per target site. For the second predictive task, this variance was higher especially for target site TS_3 (Table 3. One question that can be asked is if different target sites show slightly different patterns in their repair products. As seen in the table, for all duplicationFactors and classifiers target site TS_3 is one with the lowest AUC. This indicates that repair products from TS_3 cannot be generalized as well with the given training data. Models might be able to generalize better on repair products from more similar target sites. Models can be altered to account for this. For example, target sites can be clustered using unsupervised learning methods to group more similar behaving target sites. With the given dataset, TS_3 would preferably be in another cluster than the other target sites. Then models can then be trained on each cluster. This would result in models generalizing on more similar behaving target sites. However, determining when target site are similar might be a large predictive task.

5.3 Cell culture prediction alternative

Cell culture genotype prediction can be done using different methods. Here, cultures are described using summary features showing good results. However, alternatives can be tried. One example would be to use models from the first predictive task to predict the genotype in which a repair product is found. Using these models, prediction can be made for all repair products in a cell culture. This would result in a prediction for each repair product of either Ku80-/- or PolQ-/-. A second model can be trained on the distribution of prediction to determine the cell cultures genotype. A pipeline like this might be more robust for cell culture genotype prediction on new target sites. Results showed that predicting cell cultures genotype using summary features was effective, but future work can show if alternatives can work or might work better.

Acknowledgements

This thesis was commissioned by Delft University of Technology (TU Delft) and has been performed in cooperation with the Leiden University Medical Center (LUMC). The outbreak of COVID-19 and the lockdown increased the difficulty of certain aspects while working one this thesis. Therefore, I would like to express my sincerest gratitude to my two supervisors, Joana Gonçalves (TUD) and Robin van Schendel (LUMC). I'd like to thank both of them for their advice and guidance, continuing throughout the COVID-19 pandemic. Finally I'd like to thank Marcel Tijsterman and everyone at the Tijsterman lab (LUMC) for their support and help during the project.

References

[1] Lagunas-Rangel, F. A., and Bermúdez-Cruz, R. M., 2019. "The role of dna repair in cellular aging process". In *DNA Repair*, M. Mognato, ed. IntechOpen, Rijeka, ch. 8.

[2] Bernstein, C., Prasad, A. R., Nfonsam, V., and Bernstein, H., 2013. "Dna damage, dna repair and cancer". In *New Research Directions in DNA Repair*, C. Chen, ed. IntechOpen, Rijeka, ch. 16.

[3] Alt, F. W., and Schwer, B., 2018. "Dna double-strand breaks as drivers of neural genomic change, function, and disease". *DNA Repair,* **71**, pp. 158 – 163. Cutting-edge Perspectives in Genomic Maintenance V.

[4] Madabhushi, R., Pan, L., and Tsai, L.-H., 2014. "Dna damage and its links to neurodegeneration". *Neuron,* **83**(2), Jul, pp. 266–282. 25033177[pmid].

[5] Li, X., and Heyer, W.-D., 2008. "Homologous recombination in dna repair and dna damage tolerance". *Cell Research,* **18**(1), Jan, pp. 99–113.

[6] Krejci, L., Altmannova, V., Spirek, M., and Zhao, X., 2012. "Homologous recombination and its regulation". *Nucleic acids research,* **40**(13), Jul, pp. 5795–5818. 22467216[pmid].

[7] Bétermier, M., Bertrand, P., and Lopez, B. S., 2014. "Is non-homologous end-joining really an inherently error-prone process?". *PLoS genetics,* **10**(1), Jan, pp. e1004086–e1004086. 24453986[pmid].

[8] Mao, Z., Bozzella, M., Seluanov, A., and Gorbunova, V., 2008. "Dna repair by nonhomologous end joining and homologous recombination during cell cycle in human cells". *Cell cycle (Georgetown, Tex.),* **7**(18), Sep, pp. 2902–2906. 18769152[pmid].

[9] Mao, Z., Bozzella, M., Seluanov, A., and Gorbunova, V., 2008. "Comparison of nonhomologous end joining and homologous recombination in human cells". *DNA Repair,* **7**(10), 10, pp. 1765–1771.

[10] Pannunzio, N. R., Li, S., Watanabe, G., and Lieber, M. R., 2014. "Non-homologous end joining often uses microhomology: implications for alternative end joining". *DNA repair,* **17**, May, pp. 74–80. 24613510[pmid].

[11] Yousefzadeh, M. J., Wyatt, D. W., Takata, K.-i., Mu, Y., Hensley, S. C., Tomida, J., Bylund, G. O., Doublié, S., Johansson, E., Ramsden, D. A., McBride, K. M., and Wood, R. D., 2014. "Mechanism of suppression of chromosomal instability by dna polymerase polq". *PLOS Genetics,* **10**(10), Oct, p. e1004654.

[12] Mateos-Gomez, P. A., Gong, F., Nair, N., Miller, K. M., Lazzerini-Denchi, E., and Sfeir, A., 2015. "Mammalian polymerase θ promotes alternative nhej and suppresses recombination". *Nature,* **518**(7538), Feb, pp. 254–257. 25642960[pmid].

[13] Ceccaldi, R., Liu, J. C., Amunugama, R., Hajdu, I., Primack, B., Petalcorin, M. I. R., O'Connor, K. W., Konstantinopoulos, P. A., Elledge, S. J., Boulton, S. J., Yusufzai, T., and D'Andrea, A. D., 2015. "Homologous-recombination-deficient tumours are dependent on polθ-mediated repair". *Nature,* **518**(7538), Feb, pp. 258–262. 25642963[pmid].

[14] Arana, M. E., Seki, M., Wood, R. D., Rogozin, I. B., and Kunkel, T. A., 2008. "Low-fidelity dna synthesis by human dna polymerase theta". *Nucleic acids research,* **36**(11), Jun, pp. 3847–3856. 18503084[pmid].

[15] Wyatt, D. W., Feng, W., Conlin, M. P., Yousefzadeh, M. J., Roberts, S. A., Mieczkowski, P., Wood, R. D., Gupta, G. P., and Ramsden, D. A., 2016. "Essential roles for polymerase θ-mediated end joining in the repair of chromosome breaks". *Molecular cell,* **63**(4), Aug, pp. 662–673. 27453047[pmid].

[16] Lemmens, B., van Schendel, R., and Tijsterman, M., 2015. "Mutagenic consequences of a single g-quadruplex demonstrate mitotic inheritance of dna replication fork barriers". *Nature Communications,* **6**(1), Nov, p. 8909.

[17] Roerink, S. F., van Schendel, R., and Tijsterman, M., 2014. "Polymerase theta-mediated end joining of replication-associated dna breaks in c. elegans". *Genome research,* **24**(6), Jun, pp. 954–962. 24614976[pmid].

[18] Schimmel, J., van Schendel, R., den Dunnen, J. T., and Tijsterman, M., 2019. "Templated insertions: A smoking gun for polymerase theta-mediated end joining". *Trends in Genetics,* **35**(9), Sep, pp. 632–644.

[19] van Schendel, R., van Heteren, J., Welten, R., and Tijsterman, M., 2016. "Genomic scars generated by polymerase theta reveal the versatile mechanism of alternative end-joining". *PLoS genetics,* **12**(10), Oct, pp. e1006368–e1006368. 27755535[pmid].

[20] Shen, M. W., Arbab, M., Hsu, J. Y., Worstell, D., Culbertson, S. J., Krabbe, O., Cassa, C. A., Liu, D. R., Gifford, D. K., and Sherwood, R. I., 2018. "Predictable and precise template-free crispr editing of pathogenic variants". *Nature,* **563**(7733), Nov, pp. 646–651.

[21] Chen, W., McKenna, A., Schreiber, J., Haeussler, M., Yin, Y., Agarwal, V., Noble, W. S., and Shendure, J., 2019. "Massively parallel profiling and predictive modeling of the outcomes of crispr/cas9-mediated double-strand break repair". *Nucleic Acids Research,* **47**(15), Sep, pp. 7989–8003.

[22] Allen, F., Crepaldi, L., Alsinet, C., Strong, A. J., Kleshchevnikov, V., De Angeli, P., Páleníková, P., Khodak, A., Kiselev, V., Kosicki, M., Bassett, A. R., Harding, H., Galanty, Y., Muñoz-Martínez, F., Metzakopian, E., Jackson, S. P., and Parts, L., 2019. "Predicting the mutations generated by repair of cas9-induced double-strand breaks". *Nature Biotechnology,* **37**(1), Jan, pp. 64–72.

[23] Schimmel, J., Kool, H., van Schendel, R., and Tijsterman, M., 2017. "Mutational signatures of non-homologous and polymerase theta-mediated end-joining in embryonic stem cells". *The EMBO journal,* **36**(24), Dec, pp. 3634–3649. 29079701[pmid].

[24] Scully, R., Panday, A., Elango, R., and Willis, N. A., 2019. "Dna double-strand break repair-pathway choice in somatic mammalian cells". *Nature Reviews Molecular Cell Biology,* **20**(11), Nov, pp. 698–714.

[25] Garneau, J. E., Dupuis, M.-È., Villion, M., Romero, D. A., Barrangou, R., Boyaval, P., Fremaux, C., Horvath, P., Magadán, A. H., and Moineau, S., 2010. "The crispr/cas bacterial immune system cleaves bacteriophage and plasmid dna". *Nature,* **468**(7320), Nov, pp. 67–71.

[26] Probst, P., and Boulesteix, A., 2017. "To tune or not to tune the number of trees in random forest?". *J. Mach. Learn. Res.,* **18**, pp. 181:1–181:18.

[27] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay, 2011. "Scikit-learn: Machine learning in python". *Journal of Machine Learning Research,* **12**(85), pp. 2825–2830.

[28] Cawley, G. C., and Talbot, N. L. C., 2010. "On over-fitting in model selection and subsequent selection bias in performance evaluation". *Journal of Machine Learning Research,* **11**(70), pp. 2079–2107.

[29] Varma, S., and Simon, R., 2006. "Bias in error estimation when using cross-validation for model selection". *BMC bioinformatics,* **7**, Feb, pp. 91–91. 16504092[pmid].

[30] Wainer, J., and Cawley, G. C., 2018. "Nested cross-validation when selecting classifiers is overzealous for most practical applications". *CoRR,* **abs/1809.09446**.

[31] Hand, D. J., and Till, R. J., 2001. "A simple generalisation of the area under the roc curve for multiple class classification problems". *Machine Learning,* **45**(2), Nov, pp. 171–186.

A: Dataset features

Here, the features used by models are given. A minor description of what it does is given, the format of the feature and when necessary. Note that the sci-kit learn implementations of Logistic Regression and K-NN classifiers do not support empty or Null values. Missing values are filled with 0 rather than the mean of a feature. This is done since missing values more often indicate that there was no value to measure. For example, a null value in SNV type means no SNV was seen.

A.1. Repair product genotype prediction

Features used for genotype deficiency prediction based on a single repair product are shown in Table 4. The features shown here are calculated using the dataset provided by the Tijsterman lab at LUMC. Non cursive features are found directly in that dataset, while cursive features have been added or modified in preprocessing. The dataset contained more features, but were removed due to being repetitive or descriptive metadata. Besides the primer distance, no descriptive meta data is used. It is added in order to help models explain the difference in feature values for repair products with the same genotype and target site, but sequenced with different primers.

Feature Name	**Type**	**Describes**
Type	OHE	The type of the repair product, one hot encoded. Possible options are: deletion, insertion, delin, SNV, tandem duplication, WT (error-free product)
SNVType	OHE	A one-hot encoded representation of the type of SNV in a repair product (e.g. AT->TA).
delSize	int	The number of deleted nucleotides in the repair product. Zero when no deletion took place.
delATRatio	float	The AT ratio seen in the deleted nucleotides, between 0 and 1.
delCGRatio	float	The CG ratio seen in the deleted nucleotides, between 0 and 1.
insSize	int	The number of inserted nucleotides in the repair product. Zero when no insertion took place.
insATRatio	float	The AT ratio seen in the inserted nucleotides, between 0 and 1.
insCGRatio	float	The CG ratio seen in the inserted nucleotides, between 0 and 1.
homologyLength	int	The number of nucleotides which are possible for micro-homology. Zero when no homology is seen, -1 when no homology is possible (e.g. for insertions)
homATRatio	float	The AT ratio seen in the nucleotides used for microhomology, between 0 and 1.
homCGRatio	float	The CG ratio seen in the nucleotides used for microhomology, between 0 and 1.
mod3	int	Modulo3(insertion - deletion). Shows the shift in codon of due to the repair product. Thus values are either a 0, 1 or 2 nucleotides shift.
flankInsert	OHE	Whether a templated insertion is present. Templated insertions occur when the inserted nucleotides are a copy from other nucleotides nearby on the genome. Values are True, False and not possible (for example in deletions)
delRelativeStart	int	The start position of a deletion, relative to the position of the DSB. For insertions, this indicates the relative position where nucleotides were inserted.
delRelativeEnd	int	The end position of a deletion, relative to the position of the DSB. For insertions, this is equal to delRelativeStart.
delRelativeStartTD	int	The start position of a tandem duplication, relative to the position of the DSB. If no TD is present in the repair product, this value equals delRelativeStart.
delRelativeEndTD	int	The end position of a tandem duplication, relative to the position of the DSB. If no TD is present in the repair product, this value equals delRelativeStart.
primerDist	int	The number of nucleotides between the two primers used.

Table 4. Features used by models predicting the genotype of a cell in which repair products are found. Features in cursive are added in preprocessing. None-cursive features are retrieved directly from the dataset. OHE type means data is one-hot encoded.

A.2. Cell genotype prediction

Cell cultures are described using summary features. These features primarily focus on the mean and standard deviation of a repair products features. Similar cell cultures are expected to have similar summary features, with not too much variance. Similar cell cultures would be cultures with the same target sites and genotypes. Features used are shown in Table 5.

Feature Name	**Type**	**Describes**
TypeRatios	float $0 <= x <= 1$	For each type a repair products can be (see Table 4), the fraction it occurs in the cell culture. E.g SNVRatio.
AVGdelLen STDdelLen	float	The average and standard deviation deletion lengths of repair products in a cell culture. Uses deletions with length 0 for calculation.
AVGdelLenNoZero STDdelLenNoZero	float	The average and standard deviation deletion lengths of repair products in a cell culture. Only calculated over deletion events, thus excludes deletions of length zero.
AVGdelATRatio STDdelATRatio	float	The average and standard deviation in the AT ratio of deletions seen in repair products of a cell culture. Calculated over products with delLen > 0.
AVGdelCGRatio STDdelCGRatio	float	The average and standard deviation in the CG ratio of deletions in the repair products of a cell culture. Calculated over products with delLen > 0.
AVGinsSize STDinsSize	float	The average and deviation of inserted nucleotides. Calculated over all repair products in a cell culture. Uses insertions of length 0 for the calculation.
AVGinsSizeNoZero STDinsSizeNoZero	float	The average and deviation of inserted nucleotides. Calculated over all repair products in a cell culture. Only uses insertion types, thus excludes insertions of length 0.
AVGinsATRatio STDinsATRatio	float	The average and standard deviation in the AT ratio of insertions seen in repair products of a cell culture. Calculated over products with insLen > 0.
AVGinsCGRatio STDinsCGRatio	float	The average and standard deviation in the CG ratio of insertions in the repair products of a cell culture. Calculated over products with insLen > 0.
AVGhomLen STDhomLen	float	The average and standard deviation in the homology seen in repair products of a cell culture. Calculated over all repair products.
AVGhomLenNoZero STDhomLenNoZero	float	The average and standard deviation in the homology seen in repair products, where homology is actually present. Calculated over repair products with homLen $>= 1$.
AVGhomATRatio STDhomATRatio	float	The average and standard deviation in the AT ratio of homology seen in repair products of a cell culture. Calculated over products with homologyLength > 0.
AVGhomCGRatio STDhomCGRatio	float	The average and standard deviation in the CG ratio of homology in the repair products of a cell culture. Calculated over products with homologyLength > 0.
AVGmod3 STDmod3	float	The average and standard deviation in modulo3(insertion - deletion) of all repair products in a culture.
AVGdelRelStart STDdelRelStart	float	The average and standard deviation in the relative start position of repair products in a cell culture.
AVGdelRelEnd STDdelRelEnd	float	The average and standard deviation in the relative end position of repair products in a cell culture.
AVGdelRelStartTD STDdelRelStartTD	float	The average and standard deviation in start position of repair products with tandem duplications.
AVGdelRelEndTD STDdelRelEndTD	float	The average and standard deviation in end position of repair products with tandem duplications.

Table 5. Features used by models predicting the genotype of a cell in which repair products are found. All features are calculated during cross-validation using the NumPy library.

B: Target sites

Throughout this thesis double strand breaks are studied originating from ten different target sites. These target sites are shown in Table 6. Furthermore, this table shows the different number of primers used for sequencing. For each primer set used, a cell culture with the genotype Ku80-/-, PolQ-/- and WT is present in the dataset. This means, that for TS_9 and TS_10 the dataset contains two cell cultures for each of the genotypes. Note that sequence data of repair products contains about 200 to 250 base pairs, but only the 40 bp around the DSB are shown.

TargetSite	Primer sets used	DNA Sequence at relative position to DSB				
		−20	−10	ϟ	10	20
TS_1	1	aaggagatgg	gaggccatca		cattgtggcc	ctctgtgtgc
TS_2	1	ctataagttc	tttgctgacc		tgctggatta	cattaaagca
TS_3	1	ttgtatacct	aatcattatg		ccgaggattt	ggaaaaagtg
TS_4	1	catcacattg	tggccctctg		tgtgctcaag	gggggctata
TS_5	1	gtggccctct	gtgtgctcaa		ggggggctat	aagttctttg
TS_6	1	aaatagtgat	agatccattc		ctatgactgt	agattttatc
TS_7	1	taaaagttat	tggtggagat		gatctctcaa	ctttaactgg
TS_8	1	taattaacag	cttgctggtg		aaaaggacct	ctcgaagtgt
TS_9	2	atttgttttg	tatacctaat		cattatgccg	aggatttgga
TS_10	2	aagacttgct	cgagatgtca		tgaaggagat	gggaggccat

Table 6. The different target sites used during training. Primer sets used shows how many different primers were used to sequence the data. Note that when using different primers, different repair products will be found. The lightning strike indicates the position of the Cas9-induced DSB. The numbers above the DNA sequences indicate the relative position with regard to the DSB. This is a value used by several features, as shown in supplementary materials A.

C: Model parameters search grid

This section briefly describes the hyperparameters supplied for the search grid. Each combination of hyperparameters is tested during the grid search. Since different learning algorithms use different hyperparameters, each classifier is described separately.

C.1. Logistic regression

The following set is supplied for selecting the logistic regression hyperparameters. What hyperparameters do can be found in the scikit-learn documentation of logistic regression. Keep in mind that the max iteration hyperparameter is set, but all models trained have been stopped early due to a stopping criteria.

’model__max_iter’: 5000,
’model__tol’: 1e-4, 1e-5,
’model__class_weight’: balanced,
’model__C’: .001, .01, .05, .1, .5, .1, 5, 10, 50, 100,
’model__fit_intercept’: True, False,
’model__intercept_scaling’: 0.01, 0.1, 1, 10, 50, 100,
’model__penalty’: ’l2’, ’l1’, ’elastinet’,
’model__l1_ratio’: .1, .2, .3, .4, .5, .6, .7, .8, .9

C.2. Random forest

The following set is supplied for selecting the random forest hyperparameters. What hyperparameters do can be found in the scikit-learn documentation of the random forest classifier.

’model__class_weight’: balanced, .33, .5,
’model__criterion’: ’gini’, ’entropy’,
’model__n_estimators’: 200,
’model__max_depth’: 5, 10, 15, 20, 25, 30,
’model__min_samples_split’: 2, 4, 6, 10, 20, 30
’model__min_samples_leaf’: 1, 2, 5, 10, 15, 20
’model__max_features’: ’sqrt’,
’model__ccp_alpha’: 0.0001, 0.001

C.3. K-Nearest neighbour

The following set is supplied for selecting the K-NN hyperparameters. What hyperparameters do can be found in the scikit-learn documentation of the K-Nearest neighbour classifier.

’model__class_weight’: balanced,
’model__n_neighbors’: 3, 5, 10, 100, 200, 500,
’model__algorithm’: ’ball_tree’, ’kd_tree’,
’model__leaf_size’: 20, 30, 40,
’model__metric’: ’euclidean’, ’manhattan’

D: Over fitting in the first predictive task

Models trained for predicting in which genotype a repair product can be found using all target sites have been discussed in subsubsection 3.1.1. As shown in Figure 3, models performance can be described as modest. A possible explanation given is that over fitting leads to diminished results, also reducing any differences seen in the different training sample weights used. Evaluation on the training set is shown in Figure 9. This shows the training error for different thresholds. Again, an increase is seen in AUCs when the evaluation set (the training data in this case) is equally or fully weighted, shown respectively by the upper and lower row. This increase indicates that more frequent repair products are predicted correctly more often, correspondingly as said in subsubsection 3.1.1. When comparing Figure 9 to Figure 3, all of the K-NN, logistic regression and random forest classifiers show clear and significant difference in AUCs. This is especially true or weighted evaluation. This indicates models are over fitted. Interestingly, a clearer difference can be seen for the different training sample weights used in Figure 9. However due to this difference is only seen in the training error, this fact cannot be used to make conclusions about the different training sample weight methods. Over fitting was combated with tuning some model hyperparameters, but this leaded to diminished results compared to those shown in Figure 3.

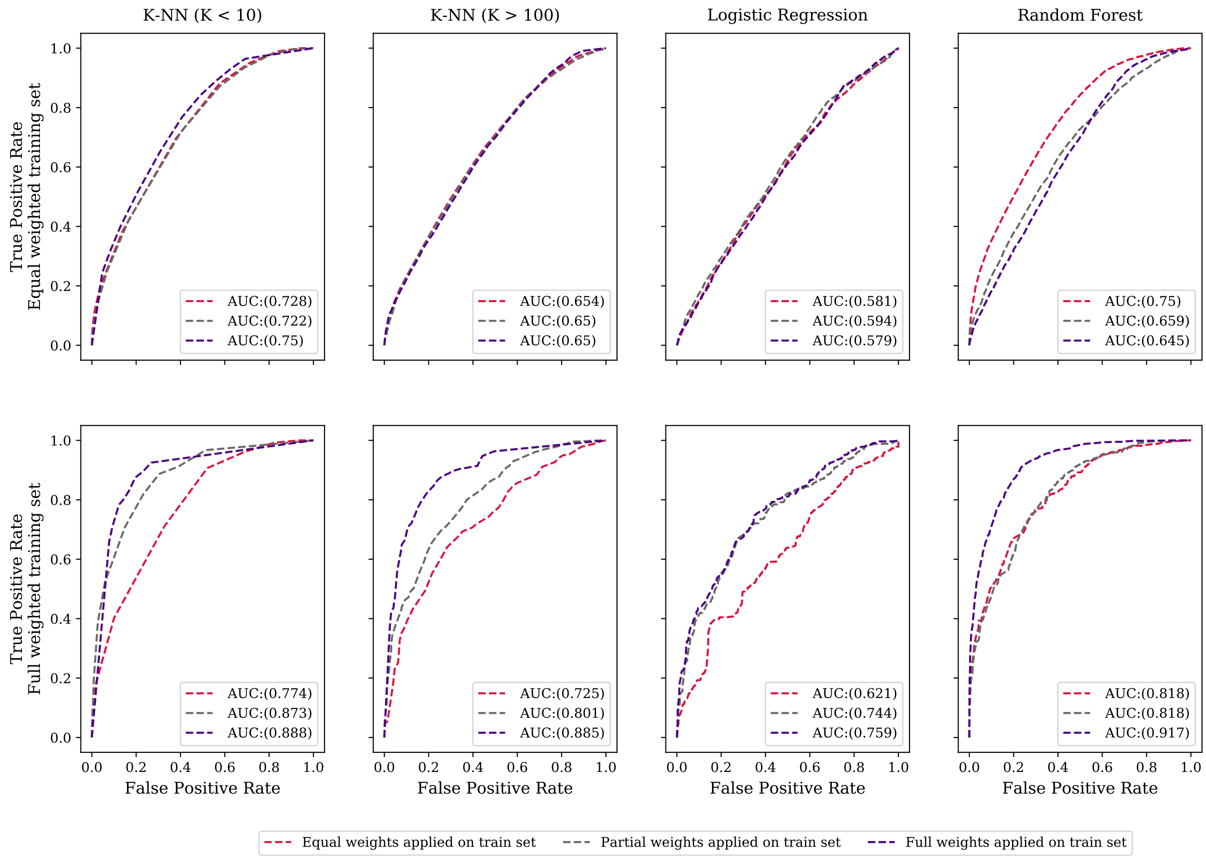


Fig. 9. Receiver operator characteristic curves of each models trained shown in subsubsection 3.1.1 while evaluated on the training data. This information shows the training error and can be used to identify over fitting.