Bioinformatics
doi.10.1093/bioinformatics/xxxxxx

OXFORD

On a journey to explore the machine learning landscape to predict breast cancer subtypes from aCGH data

Priya Gangaram-Panday [1,2576802], Sifan Liu [1,2690273], Zhexin Liu [1,2689786] and Ruben Wapperom [1,2545415]

[1] Bioinformatics, Vrije Universiteit, Amsterdam, Netherlands

Abstract

Motivation

Breast cancer is since 2020 the most diagnosed type of cancer among women. Because of the heterogeneous nature of cancer it is necessary to come up with subtypes of breast cancer to get better treatments after diagnosis and get clearer prognosis. Instead of histological examination this can be done much more efficiently using machine learning algorithms. In this paper we aim to label tumor biopsies by training machine learning models to predict subtypes. We used k-nearest neighbours (kNN), random forest (RF), gradient boosting (GB) and extreme gradient boosting (XGB) as machine learning models and for feature selection we used the mutual information (MI) approach combined with forward filtering (FF). All of this is implemented into a double cross validation (DCV) scheme.

Results and impact

The best performing model is XGB when feature selection is performed on the split training set, with an accuracy of 0.788. Feature 2184 on chromosome 17 contributed most in all models. This feature consists of nine genes, from which the gene ERBB2 produces the HER2 protein. Also chromosomes 6 and 12 contain clusters of genes that contributed a lot to the predictions. These results could impact the clinic by speeding up subtype prediction.

1 Introduction

1.1 Incidence and mortality

In 2020 11.7 percent of all cancer diagnoses were females diagnosed with breast cancer. This makes breast cancer right now the most common form of cancer, worldwide. The disease also ranks fifth in the number of cancer deaths [WHO, 2020]. Since cancer is a disease of the DNA it would be helpful to have a technique to take a look at the copy numbers of chromosomes in cancer cells. In this way the number of deaths could be reduced by applying the right type of medication for every breast cancer patient.

1.2 Breast cancer subtypes

Breast cancer consists of diverse histological profiles. The most important indicators for specific breast cancer types are progesterone receptor (PR), estrogen receptor (ER) and human epidermal growth factor receptor 2 (HER2) [Ravi et al., 2020]. There are three subtypes of breast cancer distinguished. The most severe one being the triple negative variant. None of the aforementioned receptors are present in this cancer, so there is no target. This results in a poor prognosis for patients diagnosed with triple negative breast cancer, because the only treatment option is chemotherapy [Yin et al., 2020]. The second worst prognosis would be HER2 positive breast cancer. For this subtype, treatment with monoclonal antibodies is possible to catch away the abundance of proliferation inducing HER2 protein [Cesca et al., 2020]. In case both or either ER or PR is overexpressed then prognosis for breast cancer patients is best due to the availability of plenty of drugs to target these hormone receptors. This variant is therefore called hormone receptor positive breast cancer. In this case no overexpression of the HER2 protein is detected. [McAndrew and Finn, 2020, Ravi et al., 2020]

1.3 Experimental background data set

Array comparative genomic hybridization (aCGH) is a combination of two existing techniques. One of these techniques is CGH, which was used to identify copy number aberrations in genomes, in which metaphase chromosomes were studied. To improve the resolution of the method it was combined with a technique used in RNA microarray analysis, which made it possible to detect much smaller regions of DNA aberrations [Lichter

et al., 2000]. Known DNA clones of interest were mounted on a slide of glass, each having its own spot. Fluorescently labeled reference and tumor samples, each of different color, were mixed and deposited on the slide of glass to stimulate hybridization of sample DNA with the probes. Now intensity of fluorescence is read by a computer, resulting in raw data [Schena et al., 1995].

1.4 Pre-processing of data set

Before we got the dataset, there was some preprocessing done, namely the log2-ratios were calculated for tumor DNA versus reference DNA. These log2-ratios are plotted as a function of probes in chromosomal order, which results in an aCGH profile. Further preprocessing usually contains normalization, segmentation and calling. Normalization is performed to take into account different hybridizations. Next, segmentation is necessary to get regions by merging consecutive probes of equal log2-ratios. Finally, calling is carried out by categorizing each region into a loss, normal copy number, gain or amplification [Van de Wiel et al., 2011].

1.5 Aim of the paper

For microarray data a specific pipeline has been created to analyze gene expression. Though for aCGH data there exists no known plan of action yet [Van de Wiel et al., 2011]. Ribeiro et al. [2021], explained that a gene signature was revealed for patients with oral squamous cell carcinoma to predict prognosis after treatment. In this paper we will explore this ground for breast cancer samples by performing a double cross validation on aCGH data. In the process we will make use of four models, that is RF, kNN, GB and XGB.

Then feature selection is carried out inside the cross validation loop for each model; trained on the whole data data set or on a split part of the training data set. The difference between the latter two is whether the rank of the features is decided using all the training set and validation set, or just using the training set of the outer cross validation loop. Finally, for each model we will end up with an accuracy score. It is likely that including the validation set to rank the features would have biased validation estimate, and features need to be eliminated to get a generalized model, so in this study, we focus on the questions: What effect will the possibly biased feature selection methods have on the accuracy score, if the feature elimination is very necessary in improving the prediction and which features of our aCGH data set contribute most to predicting subtypes of breast cancer?

2 Methods

2.1 Data description

For this research project, we were provided with a pre-processed data set. This data set consists of 100 samples of breast cancer patients from three subtypes. A high-resolution array CGH platform was used with 244,000 probes to measure the quantity of chromosomal DNA. Each of the 100 samples specifies the quantity of the chromosomal DNA. The data set consists of 2834 features/regions and the sample label is the associated breast cancer subtype. Each feature is labelled whether this region is a loss (-1), a normal region (0), a gain (1) or an amplification (2) of the DNA. We were provided with two raw data files, the first one consists of the pre-processed aCGH data of the breast cancer samples and the other one consists of the associated clinical outcome of the samples. These two files were merged together to make the data set that will be used to perform training and prediction on using the classification methods.

2.2 Validation protocol

We generally follow this predictor methodology construction: (1) preprocess the data and filter the genes to a smaller number. (2) Decide appropriate prediction methods suitable for the data set and feature selection methods. (3) Use reasonable training-validation scheme to train the model, in the meantime output the optimal reporters and tune the parameters for the predictors. (4) Make predictors validate on the test set to measure the performance. (5) Combine all the models based on the previously validated parameters. (6) Make a final prediction on the validation set.

Since the preprocessing is already done, we can start with implementing the validation approaches to set the parameters. We generally follow the scheme from Wessels et al. [2005]. The data set contains far more features (2834) than samples (100), so we implemented two loops of cross validation to get more representative estimates for the prediction performance and to do feature selection. This is also known as double cross validation. We start by splitting the data set into K folds. To make sure that we have sufficient data for testing, K is set to 3 [Dudoit et al., 2002]. We then take one of these sub-data sets as a validation set, and the remainder k-1 data sets are used for training. In this iteration, the training set (K-1 sub-data set) is further split into 10 folds [Kohavi and John, 1997], which is known as the inner cross validation loop. The purpose of the inner loop is to find the optimal number of reporters and optimal combination of hyperparameters for the classifier. Next, we use a certain number of features to build the model, test this on the validation set, and repeat this process 10 times so that the model is tested upon all the different folds of the validation sets. Then the final estimate is the average of the 10 different accuracy scores. Like this we then try another set of features and would get an averaged accuracy score. We continuously add features until we find the reporters with the maximum performance.

Till now the features selection is done. We are still left with the hyperparameters for the model to choose. So we repeat the same 10 Fold cross validation on the same training set to find the optimal hyperparameter for the predictor. After the reporters and the parameters for the classifier are decided, we build a final classifier using this information and predict on the validation set in the outer loop. Now we get an appropriate estimate of the performance for this one time of outer loop validation. Then we repeat the procedure above two more times until the validation is done. In order to make a more accurate estimate, this one time of double loop cross validation is repeated 100 times. And the total accuracy is taken as the average for 100 iterations. Figure 1 shows how the whole process works in general. It should be noted that samples from all classes are chosen in the same ratio in each fold (outer 3 folds and inner 10 folds) to avoid the biased prediction on different classes. This scheme is chosen so that the feature selection and parameter tuning occurs totally independent of the validation set.

2.3 Feature selection method

Since the data set contains many more features than samples, we will have to select only the important features. If we do not do this, we will most likely overfit the classifier which will lead to unreliable results. Another important reason for feature selection is that we often assume that machine learning algorithms are smart, but they can easily be influenced because they are strongly dependent on the data. Hence, it is very important to do some preprocessing steps like feature selection. This reduces the noise that can be in the data set and it helps to only work with variables that are relevant and remove collinear variables. Some of the most commonly used feature selection methods are the wrapper methods, filtering methods and embedded methods [Kohavi and John, 1997]. For this paper we decided to

use filtering methods because these methods are not based on the machine learning algorithm but on the original data set. We did not implement, for example, the wrapper method, since it has intense computation complexity [Kohavi and John, 1997]. Furthermore, it only takes into account statistical characteristics from the data. The filtering method used is the Forward Filtering (FF) method.

Before performing the filtering, the features are firstly ranked in the outer cross validation loop based on the train set split from *train.csv* or based on the whole data of *train.csv* for which it is to create the possible biased estimates. We will use the entropy-based filtering method mutual information (MI) to determine which features have to be included. MI measures the dependency between the variables and outputs a number between 0 and 1. Higher number means higher dependency between two features. The method boils down to an entropy estimation from K-nearest neighbours distances [Kraskov et al., 2011]. According to Beraha et al. [2019], the MI is useful because it assesses both the redundancy of other variables and the relevancy of features that are important in predicting the target and orders the features according to these two criteria. In other words, MI measures any kind of relation between variables (also non-linear relationships). Furthermore, transformations do not affect the order of the features [Cover and Thomas, 1991].

We apply the FF method in the inner loop of the double cross validation, where we start with the first feature with the highest MI score. Then after ten times of iteration we will get an average accuracy score. Second, we add the second feature to the set and test these two features. In this way we keep extending the feature set. We stop when the prediction performance is maximum. Then the features are selected as the combination that we want. Due to the large number of features, if we use backward filtering it might take a lot of time before we find the best feature set.

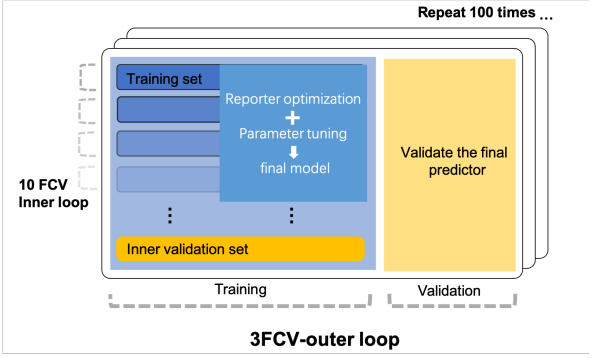

Fig. 1. Training and validation scheme. In each outer loop cross validation, the data is split into three folds. The blue box represents the training set, and the orange box represents the validation set. The number of features and parameters are validated inside the 10-fold inner loop in the training set. Finally, the general performance of the selected reporters and predictor's parameters are tested on the validation set by training a final model using all the training set. The whole double loop cross validation is repeated 100 times.

2.4 Machine learning classifiers

2.4.1 The k-Nearest Neighbours

The k-Nearest Neighbors (kNN) is a non-parametric method that is often used for regression problems; however, we will use it for this research project as a classification method [Peterson, 2009]. According to Cunningham and Delany [2020], it is considered to be a lazy learning technique due to the delay of generalization of the training data until a query is made in the system. In other words, there is either no explicit training phase or it is very minimal. The advantage of this model is its simplicity and the speed of the training part. However, the downside of this is that all the data is needed for the training phase which is time consuming. Furthermore, the kNN is sensitive to irrelevant features which can also cause problems in our case since we are dealing with many features. We will adapt the kNN machine learning to make a prediction due to its simplicity and versatility. When using the kNN as a classifier, it classifies an object based on the neighboring objects. So, the kNN groups samples based on their similarity level. We need to set the hyperparameter n_neighbors, which is set to 2, which is the value that can produce the highest accuracy score in the inner loop and the mode of those values from the outer loop. Besides, the 300 feature sets formed in the outer cross validation are merged and sorted by their occurrence time in sets. Then the FF method combined with a 3-fold cross validation is used to decide the final number of features used in the model, which can produce the optimal accuracy score. Finally, the seven most frequent features are screened. Besides, tuning parameters and feature selection for the other three models are implemented on the basis of the same idea.

2.4.2 RandomForest Classfier

The second machine learning classifier we will train the data on is the RF classifier, which is the most used supervised machine learning algorithm. This method constructs multiple decision trees and uses these as classification trees [Breiman, 2001]. However, this approach can, just like the kNN, also be used for regression. The difference between RF classifier and RF regression is that the output will be the mean instead of the mode. The output of RF in case of a classification is the mode of the classes. The advantage of RF is that it corrects for overfitting compared to the decision tree classification method [Friedman et al., 2008]. Also, the interpretation of the RF is relatively simple and it is a fast algorithm. Furthermore, the RF tries to combine and maximize the performance of multiple decision trees. The most important reason for us to implement this algorithm is its speed when identifying significant information from the features. As repeatedly mentioned, we are dealing with many features, so this advantage is very useful. We set the number of estimators equal to 100, maximum depth of the tree equal to 10, the minimum number of samples required to be at a leaf node equal to 1, the maximum samples to 80% and bootstrap as True.

2.4.3 Gradient Boosting

Boosting is a machine learning algorithm that converts weak learners into strong learners. The GB algorithm, originally devised by Breiman [1997], can be used for both regression and classification problems just like the kNN. This method makes an ensemble of weak prediction models which are often decision trees. According to Piryonesi and El-Diraby [2020], the GB algorithm outperforms the RF if the decision tree is a weak learner. The GB algorithm builds a model, just like other boosting approaches, stage-wise and making them general by allowing optimization of a differential loss function [Hastie et al., 2009]. The GB can be implemented in supervised machine learning problems where the aim is to find a function that minimizes a loss function. The loss function is a measure which indicates how good a model's coefficient is in fitting the data. In our case the loss function is a measure of how good the predictive model is in classifying the breast cancer subtypes. The key reason for using the GB is that it is capable of optimizing a specified function rather than only a lost function. The downside, however, of this method is that it is a greedy algorithm that can quickly overfit data. The parameters we used are: maximum depth = 60, minimum samples =2, minimum samples split = 2, maximum features = square root.

2.4.4 Extreme Gradient Boosting

Chen and Guestrin [2016] have proposed an advanced version of the gradient boosting method, the eXtreme Gradient Boosting (XGBoost). This version has, compared to the GB, higher computational power and is better in dealing with the overfitting problems. The key difference between GB and XGBoost is that the GB has no regularization which the XGBoost has. Regularization is also performing variable selection since it tries pushing the weights for many variables to zero. Regularization can especially be useful for high-dimensional data. XGBoost regularizes the weights of variables or shrinks them to zero. When the weights decrease, the variance of the predictions will also decrease but the bias will increase which is a commonly known trade-off. The important parameters that are required by XGBoost are: maximum number of iterations, size of the tree/maximum depth and the learning rate. Because we are dealing with many features the XGBoost can give us nice results since it was designed for this type of data. The parameters are set to: Number of estimations: 91 (the optimal number of rounds/trees required), Maximum dept = 4 (number of splits in each tree), Minimum child weighs =0.8 (minimum sum of the weights of all the observations that are required in tree child), Subsample = 0.6, Learning rate = 0.1 (how quickly algorithms adapt tree to the growing model).

3 Results

The average accuracy scores and the standard deviations are calculated for the four models and are shown in Table 1. The accuracy score for kNN is 0.757 when using the complete training set to select features, 0.703 when feature selection is performed on a split training set and 0.433 when no feature selection is performed. All of the accuracy scores are lower than the accuracy scores obtained from the RF model and GB model (0.807, 0.761, 0.689; 0.816, 0.759,0.779, respectively). The accuracy scores of XGB are the highest (0.841, 0.788, 0.826). The biases are further tested and are significantly different (the third panel in Figure 2). These results are as expected, because the kNN model is a lazy classification algorithm, and it just assigns the label based on the distances between the nearby samples. On the other hand, RF, GB and XGB are ensemble models that build multiple decision trees and are based on different features, so it should have boosted prediction performance. The standard deviations of RF are overall the highest (0.077 and 0.071), and XGB has the lowest standard deviations (0.062 and 0.057), which also confirms the superiority of the XGB and why it is the most popular machine learning method nowadays. The standard deviations of regular GB and kNN are (0.068 and 0.063 for GB, 0.064 and 0.063 for kNN)

When compare the accuracies between the two ways that feature selection was done, an obvious trend could be shown, all the models have significantly better accuracy scores when the feature selection includes the validation set (middle and right panel in 2). From Figure 2, because we observed that each model has similar variances when in the two cases, hence the Leven's test was implemented to assess the equality of the variances. We obtained that all models have p-values higher than 0.05 and then a two sample t-test was done to check the significant level. This gave us the result that every model has reached p-value < 0.01. Therefore, we could conclude that using the validation set, to help rank the features, truly leads to higher visible accuracy score and overestimates the performance.

As expected, the accuracy scores for all the models, when features are not selected, are much lower than when features are selected by the FF (the first panel in Figure 2). The possible explanation is that too many unimportant features make the classifier memorize the noise instead of making a generalized prediction. Note that kNN has the most decrease in performance (towards 0.43). Which is, again, due to simply assigning classes based on its neighbours, but when the instance is in a far higher dimensional space, the prediction is more likely to get affected by those noisy features. The other three ensemble models show obvious better performance because they could adjust the depth of trees and use regularization to prevent overfitting the unrelated noise.

Method	Model	Acc	Std	Acc_2184	Features	Entropy
No feature selection	kNN	0.443	0.075			
	RF	0.689	0.075			
	GB	0.779	0.070			
	XGB	0.826	0.057			
Feature Selection based on splited train set	kNN	0.703	0.064	0.642	7	4.68
	RF	0.761	0.077	0.675	49	5.50
	GB	0.759	0.068	0.649	33	5.42
	XGB	0.788	0.645	0.62	43	5.42
Feature Selection based on whole train set	kNN	0.757	0.063			
	RF	0.807	0.071			
	GB	0.816	0.063			
	XGB	0.841	0.057			

Table 1. Accuracy scores and standard deviations of the test data set for the four models. The middle column is the accuracy scores of models only based on 2184. The two columns on the right are the information of the optimal feature numbers selected for four models and the entropy of 50 features. The row names have the same meaning as shown in the box plot in figure 2.

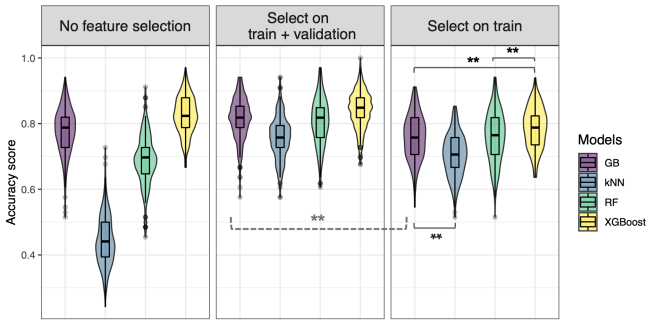

Fig. 2. Box plot of accuracy scores for four models (kNN, RF, GB and XGB). The left panel represents the accuracy scores of each model without feature selection. The middle panel represents the accuracy scores of models with feature selection that is based on both the training and validation set. The right panel represents accuracy scores of models with feature selection that is based on only the training set in the outer loop. The legend refers to the four models used and their corresponding colours in this plot. The two-sample t-test is done for models when feature selection is conducted only on the training set. XGB is significantly the best, while kNN performs the worst. "*": p-value < 0.05, "**": p-value < 0.01.

Now that the classifiers are built and we see that they show decent prediction performance, we could further study what features are selected most times during the 100 times of cross validation. Important reporters could be of great value and regarded as biomarkers. For each aforementioned model, the 300 feature sets in the outer cross validation loop are merged and sorted by their occurrence time in sets, then a new ranking from each model is formed. Next, we applied the FF method according to the ranked feature list and 3-fold cross validation to select the ultimate top N features as the parameter for the final model. This narrowed feature number to only 43 features for RF model, 7 features for kNN model, 33 features for GB and 43 features for XGB (Table 1).

In order to compare the specific features selected by different models, we extracted the top 50 features that occurred in all the models (Figure

3). The first ranked feature (with index 2184) got selected all 300 times. But the second best one decreases sharply to less than 100 times. From Figure 3a, we observe a long tail and only very few features are dependent on the subtypes. In this case the feature 2184 is no doubt the strongest predictor. If we were to build a classifier with one feature, this feature should be considered. In order to see how much prediction it is able to support, all the models are trained using this one feature. It turns out that it could explain 78% - 91% of the accuracy (kNN the most and XGB the least, Table 1.), which confirms its dominant dependence. Furthermore, the occurrence heatmap (grey bars below the coloured bars in Figure 3a), shows that only top 15 features were selected in all the models, and predictors afterwards are specifically picked out by different models, which means these features are model-biased. This also indicates there exits very few critical predictors.

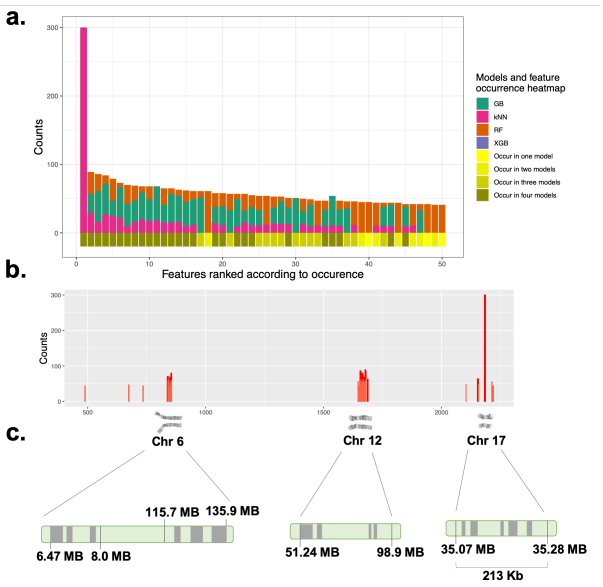

Fig. 3. a)Features from different models are ranked according to how many times in total they are selected in the 300 times of inner loop cross validations. X axis stands for each feature and the y axis is the number of occurrences. Here the top 50 features are shown. Below the y = 0 horizontal line, the heatmap of how many models the feature was selected is displayed. A darker bar indicates the feature picked out by more models. It is apparent that the top one feature performed as a reporter all the 300 times, and the rest of the features have sharply decreased predictive power. Also note that from the heatmap, only a few features play important roles in all the models. b) Top 50 features mapped to chromosomes. On the horizontal axis the genome can be seen chromosomally ordered. From this plot the positions of the biomarkers and how wide they expand can be clearly seen. The best performing feature can be found as the highest bar on the right on chromosome 17. This involves 9 genes on a 213 kb region. Also on chromosome 6 and 12 aggregated biomarkers can be seen. c) Base pair location of features on chromosomes. The best performing feature 2184 on a 213 kb long short segment. Features on chromosome 12 (202 genes) and 6 (205 genes) are in a more scattered manner distributed on a wider range.

To reveal the connection between the biomarkers and its biological explanation, the top 50 features were mapped to the reference genome (NCBI36, Ensemble 54) by their position coordinates and the corresponding genes were mined. Figure 3b and 3c show their locations on chromosomes. It is interesting that they are mainly only distributed on three human chromosomes, with the strongest biomarker 2184 on chr 17, some on chr 12 and the rest on chr 6. And note that on each chromosome they are closely located in a short segment. In Figure 3c, for example, feature 2184 has the highest peak on chr 17 which corresponds to nine genes on a 213 kb long region. This indicates that the most dependent

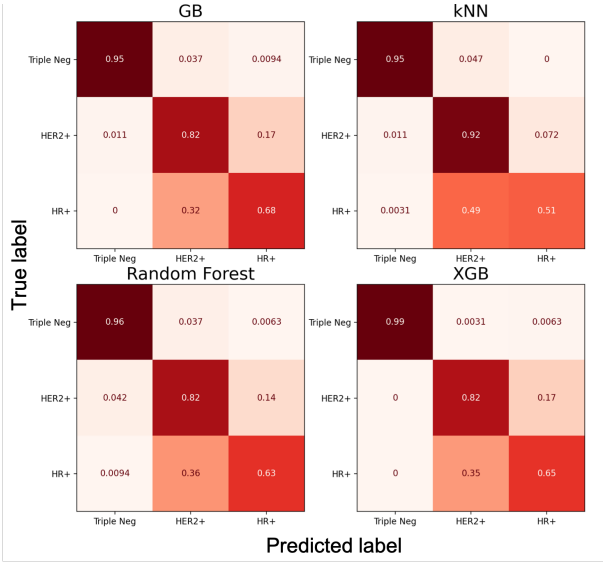

Fig. 4. For every model a confusion matrix is produced. In each submatrix the y axis stands for true subtypes, and the x axis represents predicted labels by each model. The percentage of samples in each subtype that got predicted correctly could be told from the heatmap and the ratio in each cell. Notable is that for every model at least one third of what should be predicted as HR+ is predicted falsely as being HER2+. Colors also indicate a heatmap with lightest colors indicating low values and darker colors representing higher values.

predicted variables come from certain specific regions instead of multiple scattered segments.

4 Discussion and conclusion

4.1 Issues that might affect the result

First, prediction methods need to be carefully chosen for a data set with high dimension and a low sample size. Naive models such as kNN or linear models need to neatly select predictors to reduce biases, and data hungry models like neural networks could easily overfit. So here an ensemble model with regularization and repeating cross-validation iterations would better generalize the data. Second, the order of filtering the features would have effects on the selection of best-performed reporters. Insignificant variables with misranking would affect all the models built later. Thus trying different ranking methods and repeating multiple times would lead to an unbiased selection of features. Plus, in the inner loop of cross-validation to select features, even though we split 10 folds and are left with a small size of test set, this is still used for scoring the features before the FF iteration starts. So to avoid overestimation the implementation of feature ranking should be independent of validation data, and this is the part that could still be improved.

4.2 Biological interpretation biomarker

For all four models, feature 2184 was number one in the feature ranking. This feature may act as a good biomarker for use in a clinical setting. The genes inside this region are TCAP, PNMT, PERLD1, ERBB2, C17orf37, GRB7, IKZF3, AC079199.2 and ZPBP2. The most important gene seems to be ERBB2, which produces the HER2 protein [Joshi et al., 2020]. This most probably affects the predictive power of HER2+ samples, because in TN and HR+ samples the HER2 protein is not present. Genes c17orf37 and GRB7 are related to diverse cancers. Gene c17orf37 is also known as MIEN1, which is a gene that is connected to cell migration and invasion,

which may cause metastases. MIEN1 is involved in apoptosis as well, which is an important factor in tumor development [Dasgupta et al., 2009]. GRB7 plays a role in cell migration and breast cancer, because it generates a protein that binds with growth factor receptors [Wang et al., 2020].

4.3 Research question

It is shown that for a data set with this high dimensionality, training without feature selection has the worse prediction performance. And although ranking the features on the whole data set has higher overall accuracy score, this would be explained by overfitting the validation set. So performancing ranking independently of validation data should be emphasised and would lead to much unbiased estimates and better models. For the second research question, it was vividly demonstrated that feature 2184, a 213 kb-long region on chromosome 17 no doubt has the strongest prediction power.

4.4 Future research

When taking the 50 best performing features and plotting them onto a chromosome then three relatively short distinct locations can be found with a high density of features. These can be found on chromosomes 6, 12 and 17. Further gene function analysis could be performed for the features that are clustered on chromosomes 6 and 12. In Figure 4, a confusion matrix can be seen for every model. In it the number in each cell represents the ratio of how many samples are correctly predicted in each subtypes. What is most noteworthy is the fact that for every model the HR+ subtypes mostly are predicted as HER2+ subtype. This could be investigated in more depth in another research to find out why this specific pattern appears in every model.

References

M. Beraha, A. M. Metelli, M. Papini, A. Tirinzoni, and M. Restelli. Feature selection via mutual information: new theoretical insights. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2019.

L. Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at, 1997.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

M. G. Cesca, L. Vian, S. Cristóvão-Ferreira, N. Pondé, and E. D. Azambuja. Her2-positive advanced breast cancer treatment in 2020. *Cancer Treatment Reviews*, 88, 2020.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

T. M. Cover and J. A. Thomas. Entropy, relative entropy and mutual information. *Elements of information theory*, 2(1):12–13, 1991.

P. Cunningham and S. J. Delany. k-nearest neighbour classifiers–. *arXiv preprint arXiv:2004.04523*, 2020.

S. Dasgupta, L. Wasson, N. Rauniyar, L. Prokai, J. Borejdo, and J. Vishwanatha. Novel gene c17orf37 in 17q12 amplicon promotes migration and invasion of prostate cancer cells. *Oncogene*, 28(32):2860–2872, 2009.

S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

T. Hastie, R. Tibshirani, and J. Friedman. Boosting and additive trees. In *The elements of statistical learning*, pages 337–387. Springer, 2009.

S. K. Joshi, J. M. Keck, C. A. Eide, D. Bottomly, E. Traer, J. W. Tyner, S. K. McWeeney, C. E. Tognon, and B. J. Druker. Erbb2/her2 mutations are transforming and therapeutically targetable in leukemia. *Leukemia*, 34(10):2798–2804, 2020.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

A. Kraskov, H. Stögbauer, and P. Grassberger. Erratum: estimating mutual information [phys. rev. e 69, 066138 (2004)]. *Physical Review E*, 83(1):019903, 2011.

P. Lichter, S. Joos, M. Bentz, and S. Lampel. Comparative genomic hybridization: uses and limitations. In *Seminars in hematology*, volume 37, pages 348–357. Elsevier, 2000.

N. P. McAndrew and R. S. Finn. Management of er positive metastatic breast cancer. In *Seminars in Oncology*. Elsevier, 2020.

L. E. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

S. M. Piryonesi and T. E. El-Diraby. Data analytics in asset management: Cost-effective prediction of the pavement condition index. *Journal of Infrastructure Systems*, 26(1):04019036, 2020.

R. Ravi, G. Haider, K. Ahmed, S. Zahoor, A. Sami, and R. Lata. Frequency of hormone receptors and her-2/neu receptor positivity in different histology in breast cancer patients. *Journal of Ayub Medical College Abbottabad*, 32(3):323–326, 2020.

I. P. Ribeiro, L. Esteves, A. Santos, L. Barroso, F. Marques, F. Caramelo, J. B. Melo, and I. M. Carreira. A seven-gene signature to predict the prognosis of oral squamous cell carcinoma. *Oncogene*, pages 1–11, 2021.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270 (5235):467–470, 1995.

M. A. Van de Wiel, F. Picard, W. N. Van Wieringen, and B. Ylstra. Preprocessing and downstream analysis of microarray dna copy number profiles. *Briefings in bioinformatics*, 12(1):10–21, 2011.

Z. Wang, Y. Wang, Y. He, N. Zhang, W. Chang, and Y. Niu. Aquaporin-1 facilitates proliferation and invasion of gastric cancer cells via grb7-mediated erk and ras activation. *Animal Cells and Systems*, 24(5):253–259, 2020.

L. F. Wessels, M. J. Reinders, A. A. Hart, C. J. Veenman, H. Dai, Y. D. He, and L. J. v. Veer. A protocol for building and evaluating predictors of disease state based on microarray data. *Bioinformatics*, 21(19):3755–3762, 2005.

WHO. Latest global cancer data: Cancer burden rises to 19.3 million new cases and 10.0 million cancer deaths in 2020. (292):1–3, 2020.

L. Yin, J.-J. Duan, X.-W. Bian, and S.-C. Yu. Triple-negative breast cancer molecular subtyping and treatment progress. *Breast Cancer Research*, 22(1), 2020.