Computational and Structural Biotechnology Journal 20 (2022) 5453–5465

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

Unleashing high content screening in hit detection – Benchmarking AI workflows including novelty detection

Erwin Kupczyk [a,b], Kenji Schorpp [c], Kamyar Hadian [c], Sean Lin [c], Dimitrios Tziotis [a], Philippe Schmitt-Kopplin [a,b,*], Constanze Mueller [a,*]

[a] Research Unit Analytical BioGeoChemistry, Helmholtz Zentrum München, Ingolstaedter Landstr. 1, 85764 Neuherberg, Germany
[b] Comprehensive Foodomics Platform, Chair of Analytical Food Chemistry, TUM School of Life Sciences, Technical University of Munich, Maximus-von-Imhof-Forum 2, 85354 Freising, Germany
[c] Institute for Molecular Toxicology and Pharmacology, Cell Signaling and Chemical Biology, Helmholtz Zentrum München, Ingolstaedter Landstr. 1, 85764 Neuherberg, Germany

ARTICLE INFO

Article history:
Received 30 June 2022
Received in revised form 16 September 2022
Accepted 16 September 2022
Available online 27 September 2022

Keywords:
High-content screening
Machine learning
Deep learning
Classifier
Novelty detection
Bioactives
Hit detection
Cell painting

ABSTRACT

Complex mixtures containing natural products are still an interesting source of novel drug candidates. High content screening (HCS) is a popular tool to screen for such. In particular, multiplexed HCS assays promise comprehensive bioactivity profiles, but generate also high amounts of data. Yet, only some machine learning (ML) applications for data analysis are available and these usually require a profound knowledge of the underlying cell biology. Unfortunately, there are no applications that simply predict if samples are biologically active or not (any kind of bioactivity). Within this work, we benchmark ML algorithms for binary classification, starting with classical ML models, which are the standard classifiers of the scikit-learn library or ensemble models of these classifiers (a total of 92 models tested). Followed by a partial least square regression (PLSR)-based classification (44 tested models in total) and simple artificial neural networks (ANNs) with dense layers (72 tested models in total). In addition, a novelty detection (ND) was examined, which is supposed to handle unknown patterns. For the final analysis the models, with and without upstream ND, were tested with two independent data sets. In our analysis, a stacking model, an ensamble model of class ML algorithms, performed best to predict new and unknown data. ND improved the predictions of the models and was useful to handle unknown patterns. Importantly, the classifier presented here can be easily rebuilt and be adapted to the data and demands of other groups. The hit detector (ND + stacking model) is universal and suitable for a broader application to support the search for new drug candidates.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Seeking for new bioactive compounds, natural product containing complex mixtures are an interesting sources [1,2]. Therefore, many medicines are based on natural products [3] and it is still possible to find novel biologically active substances in complex mixtures as e.g. shown by Furner-Pardoe *et al.* [4] Thanks to technological developments in the *omics* disciplines, comprehensive screenings of complex mixtures became more efficiently and faster [5]. One of these technical developments is high-content screening (HCS), a useful tool that for a comprehensive bioactivity analysis.

HCS has become very popular in *in-vitro* drug screenings or development [6], as it allows to observe how compounds or mixtures affect the cellular morphology in high-throughput (HT) campaigns. For this purpose, cells are cultured in multi-well plates (96, 384 or 1536 well format) and treated with different samples in parallel. The cells are then stained with different fluorescent dyes or antibodies to detect changes in cellular substructures and organelles. Depended on the HCS instrument several dyes can be multiplexed, which leads to a growing complexity and size of the obtained data. Due to the rapid development of artificial intelligence (AI) in the field of image recognition [7], it is possible to analyse the generated microscopic images and convert them into a multitude of numerical features [8] allowing an efficient screening for new compounds. However, the HT and high number of features per sample results in large and multidimensional data sets resulting complex data for subsequent data analysis. These data

* Corresponding author at: Research Unit Analytical BioGeoChemistry, Helmholtz Zentrum München, Ingolstaedter Landstr. 1, 85764 Neuherberg, Germany.
E-mail addresses: Schmitt-Kopplin@helmholtz-muenchen.de (P. Schmitt-Kopplin), constanze.mueller@helmholtz-muenchen.de (C. Mueller).

https://doi.org/10.1016/j.csbj.2022.09.023
2001-0370/© 2022 The Author(s). Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

sets are often analysed with classical statistical methods [3,9–12], or more recently also with machine learning (ML) approaches [13–17], a subfield of AI. An example for an application using ML is the classification of morphological differences. This can be done using image data [8,18–21] or extracted morphological features [6,22–23]. For such purposes, support vector machines (SVM) [24], random forest classifiers [25] or voting models [25] are often used. Applications using image data for analysis have an artificial neural network (ANN) algorithm upstream of the classifier in order to convert the images into numerical features. Based on the distinction of morphologies, there are applications that classify different mechanism of action [26] or modes of action [27] with the help of SVM. However, the major drawback for ML models is their lack in recognition of new and unknown patterns. This is a major problem for screening approaches, as the models force the data in previously defined categories leading to missing or false detections. One way to tackle this issue is to use novelty detection (ND) [24,28–30]. ND can be understood as an outlier test, which tests how similar new samples/data are to the already known samples/-data. Unlike a classical outlier test, there is no need to adjust the alpha error and therefore there is no need to know the final number of performed tests. This makes it suitable for an application that will be used countless times.

To best of our knowledge, this is the first paper benchmarking algorithms for fast and easy hit detection. A proper hit detection and selection takes a key position in any screening and allows a sample prioritisation. Fig. 1 shows a general overview of the workflow of data analysis in HCS [31–33]. With the Hit Detector highlighted in green. We present a systematic analysis of different models, from classical ML models (95 models) over an alternative partial least square regression (PLSR) (44 models) to simple ANNs with dense layers only (72 models). For the classical models (9 models), all classifiers of the scikit-learn [34] library were checked. These included the already mentioned SVM and random forest. Various voting models (15 models) were also examined, which can be counted to the group of ensemble methods. For voting models, different models are combined to predict the class, each model gives a vote and the prediction with the most votes wins. Stacking models (68 models) which are also ensemble models were also examined. A stacking model consists of two levels, the first consists of several different models and are called the base estimators. It is up to the user to decide how many models and which models to use. These models are trained with the data and produce outputs which are the input for the last level. The last level is a single model that learns to combine the output/information of the lower level in a meaningful way to improve performance. We also decided to use PLSR (44 models) for classification, since it is the most commonly used approach in different *omics* fields [35,36]. PLSR is similar to principal component analysis (PCA), which is often used to discover classes and groups in the data. It also reduces the dimension of the variables. The ability to discover classes in data is ideal for the task of classification, as it supports it. The PLSR can separate the data in such a way that mayor class differences can be found, which can facilitate the classification of the HCS data. The possibility of reducing the features has an additional advantage, because it decrease the required computing resources of the computers. In the investigation of neural networks (72 models), only dense layers were used, as other layers would go beyond our scope. Dense layers are those layers where each unit is connected to all units of the previous layer. Other layers include the dropout layer (where random connections are removed), flatten layers (where multi- dimensional data matrices are converted into a one-dimensional one) and many more. Even with the use of dense layers, there are countless architectures that can be built. For example, one can vary the number of layers and the number of nodes within a layer. In addition, neural networks with dense layers are the classical variants, which is why these architectures are usually used as a starting point.

All built models were evaluated using five different metrics (accuracy, precision, recall, specificity and optimised accuracy based on Ranawana and Palade [37]).

2. Methods and materials

2.1. Hardware and software

The study was calculated on two computers. The first computer uses an Intel processor i7 7800X with 16 Gb RAM and an Nvidia RTX 1060 graphics card. The second computer has an AMD Ryzen 9 5900HX processor with 64 Gb RAM and an Nvidia Geforce RTX 3080 graphics card.

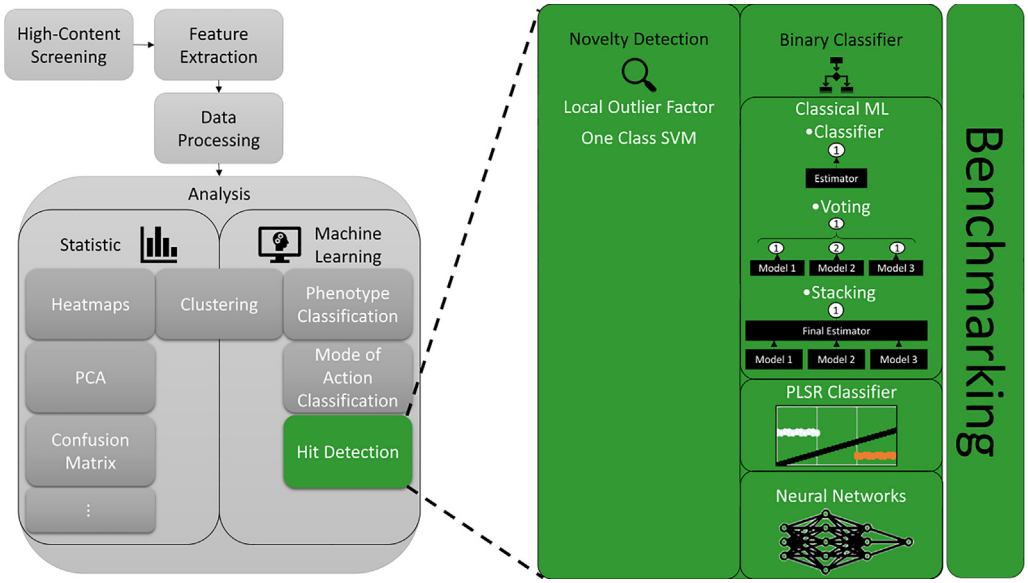

Fig. 1. A common workflow of high content screening data analysis and how this study fits in the workflow (marked in green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The open-source software Python, scikit-learn, Jupyter Notebook and Tensorflow were used in the Anaconda application [34,38–41].

2.2. Experimental procedures and data sets

To capture a broad but defined set of phenotypes we decided to analyse the SELECTCHEM Target Selective library in the Standard Cell Painting procedure [42]. The library is a unique selection of 641 validated highly selective pharmaceutically relevant inhibitors covering over 123 targets. Their selectivity is at least 100-fold higher relative to their non-primary targets, leading to minimal off-target activity. Furthermore, the compounds are structurally diverse and highly cell permeable. We used U2OS cells for all analysis and dimethyl sulfoxide (DMSO) treated cells served as negative controls. The samples were treated following the procedure of Brey *et al.* [42] except that the stains were divided into two groups to avoid fluorescents overlapping. Each sample was measured in triplicate using 10 μM as final concentration. The DMSO controls were positioned randomized between the three replicates, but were in the same position on both plates for the staining pair. Images were acquired on the PerkinElmer Operetta® High Content Imaging System, and Perkin Elmer software Columbus (2.9.1532) was used for feature extraction. In total 979 features were recorded. These features consisted of the mean value of morphological (STAR and Shape) and texture (SER) characteristics, and if possible their standard deviations.

Two additional data sets were used for the final verification of the models. Both data sets were generated using the same experimental procedure. The first contained drugs that affect the cell cycle in different stages (mebendazole, nicolsamide and cladribine), each in a final concentration of 10 μm. The second validation set contained staurosporines at different concentrations (10 nM and 30 nM).

2.3. Data Processing

To process the data we used a Jupyter notebook as described below [39]. First of all, the not a number (NaN) values were replaced by 0. Then only the control samples were used and Z-score transformed. For each sample, it was calculated how many features were outliers using the Grubbs outlier test. If there were more than 30 outlier features, the control sample was removed. The complete data set, with cleaned controls, was then Z-score transformed using the mean and standard deviation of the controls [43]. Finally, single replicate data was kept and combined into one data set, containing 1762 bioactive samples and 306 controls for the first data set. For the cell cycle interfering compounds there were 151 data with 32 controls and for the staurosporine data set there were 146 values of which 41 are controls.

2.4. Metrics for evaluation

Accuracy, precision, recall and specificity were calculated to check the quality of the models. In addition, an optimised accuracy was used based on Ranawana and Palade [37]. This allows a better evaluation of the performance of imbalanced data sets. Accuracy indicates how well the model is predicting, but has a problem with unbalanced data. For instance, a model that predicts everything as the majority class automatically has a higher accuracy due to the imbalance of the data. However, this falsely gives the impression of having a good model, even though it cannot predict the minor class. This problem is better handled with optimised accuracy. The precision indicates how many of the predicted bioactive substances are indeed bioactive, whereas the recall gives how many bioactive substances were indicated as bioactive substances. In a screening approach, it is therefore more important to have high precision and, in contrast, the recall rate can be lower. Specificity represents how many controls were actually predicted to be controls, which is also important in a screening approach.

Our focus at the beginning of the model selection was on high optimised accuracy. In the later tests, more attention was paid to precision and specificity. As mentioned, for a screening approach it is important that a model predict controls well (high specificity) and only recognises biologically active samples as such when they are (high precision).

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$OptimizedAccuracy = (Specificity * N_n) + (Recall * N_p) - \frac{|Specificity - Recall|}{Specificity + Recall}$$

- N_n: proportion of negative samples
- N_p: proportion of positive samples
- TP: True Positive
- TN: True Negative
- FP: False Positive
- FN: False Negative

2.5. Splitting the data

When working with small data sets, the problem is how to train the model and check it. There are several ways to tackle this problem, some examples can be seen at Gowen *et al.* [44] We decided to use a classic train/test split. Splitting the data is important to verify the performance of an ML model. Because two extreme cases can occur when learning the models, the overfitting and the underfitting. Underfitting can be recognized quickly, since the model cannot even predict the training data. The model did not recognize any meaningful patterns in the data. The other case, overfitting, can only be recognized with an additional data set. Here the model perfectly predicts the training data but cannot predict the test data. This shows that the model has learned too specific patterns from the training data that do not contribute to solving the problem. An ideal ML model learns patterns that are necessary to solve the problem and that are applicable to all data in the domain.

For this, scikit-learn offers stratified shuffle splitting, where a fixed random state can be set. This was done so that the generated data sets were always the same. The data was divided into three sets, a training set (70 % of the data), a validation set (10 %) and a test set (20 % of the data). By keeping the data sets fixed, it was easier to compare the models with each other. It also prevents leakage of information. If one were to use randomly split data sets repeatedly, all data is offered to the model during training at some point. The test for generalisation would no longer be guaranteed, since the information from all data went into the model.

2.6. Modelling workflow

To give an overview, the modelling workflow is illustrated schematically (Fig. 2). First, the data is pre-processed using a Z-transformation and the Grubbs outlier test. Then the data was divided into three sub-sets, the training set (70 % of the original data), the validation set (10 %) and the test set (20 %). The training set was used for hyperparameter optimisation with a grid search cross-validation (GSCV). The models with adjusted parameters were tested with the validation data. If the performance of the models fulfilled certain criteria, the optimised models were used for further investigation. For this purpose, the training set was merged with the validation set and used to train the optimised models. The performance of the models was checked with the remaining test set. If the performance also fulfilled certain criteria, the model was checked with a final cross-validation (CV). For this purpose, all data sets were merged and used. This last validation was done because the performance of the models depends strongly on the data size and the used data. Since all data is usually used to train the final model, a CV was performed with all data. This gives a better overview of the performance of the final model.

2.7. Data availability

The extracted raw and processed data, as well as the Python scripts that support the findings of this study are available in Mendeley Data and found at: https://doi.org/10.17632/hctkwwzx5z.1 [45].

3. Results

3.1. Classical machine learning approaches

We began by testing the classifiers from the scikit-learn library, which includes nine models. In the paper, we refer to these standard classifiers from the library as the classical classifiers, as they are the most common approaches. The first step was to check the models in their default settings to see how they perform. Models such as stochastic gradient descent (SGD) classifier, ridge classifier or logistic regression were already performing acceptably, but there was still much potential for improvement (Fig. 3). A GSCV with CV = 5 and fixed random state was therefore performed for the hyperparameter optimisation. A GSCV takes a list of different settings for the different parameters and tests all possible combinations with a CV. It then returns the parameter setting that gave the best performance. The models were thereafter tested with the optimised parameters (Supplementary Materials 1). Apart from k-nearest neighbours (KNN), SGD and the Gaussian Process Classifier (GPC), all models improved. However, the performance was still not sufficient, ranging between minimal 43 % (decision tree classifier) to maximal 79 % (Ridge Classifier) optimised accuracy (Fig. 4).

As a consequence, we decided to evaluate the performance using different ensemble models. Ensemble approaches combine multiple models and can improve predictions. In the tested standard models of scikit-learn, an ensemble model is already included, namely random forest. It is a model that uses multiple decision trees to make its prediction. After optimizing the parameters, this model gave the largest performance increase of almost 50 % for optimised accuracy (Figs. 3 and 4). This shows the advantage of ensemble models, when set up correctly; different models can support each other and improve performance. For the new ensemble models we first investigated voting models, where one can choose between hard voting (9 models) and soft voting (6 models). Hard voting works by selecting the prediction with the highest number of votes. In contrast, soft voting sums the probabilities of all the predictions in each model and selects the prediction with the highest probability achieved. Nevertheless, there was neither an increase in performance with the hard voting nor with the soft voting approach (Supplementary Material 1).

The next ensemble approach that was tested were stacking models. A total of 68 different models were analysed in this study. To this end, different combinations of basic models and final estimators were built. For the stacking models, an improvement of almost 7 % in performance was observed (Supplementary Material 1). The models were also checked with reduced features where the standard deviations were omitted. For further testing, we selected all models with a very high optimised accuracy (greater than80 %),

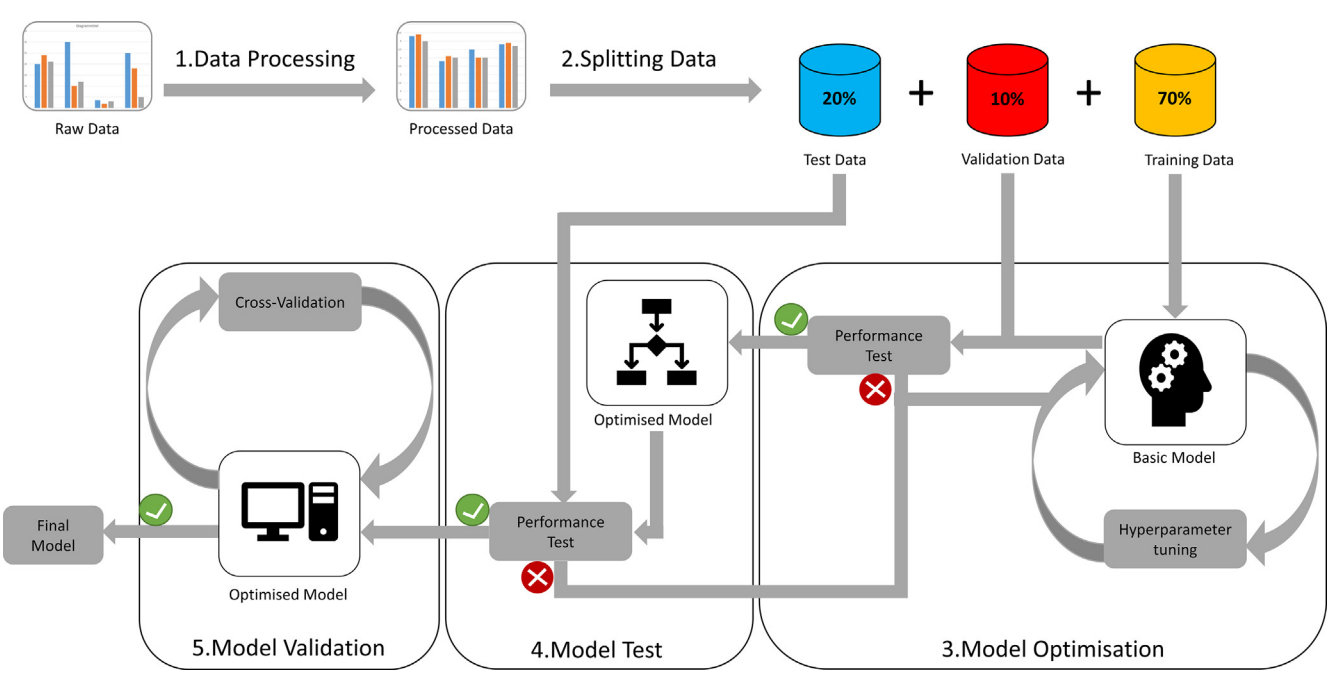

Fig. 2. Schematic representation of the modelling workflow used in this work.

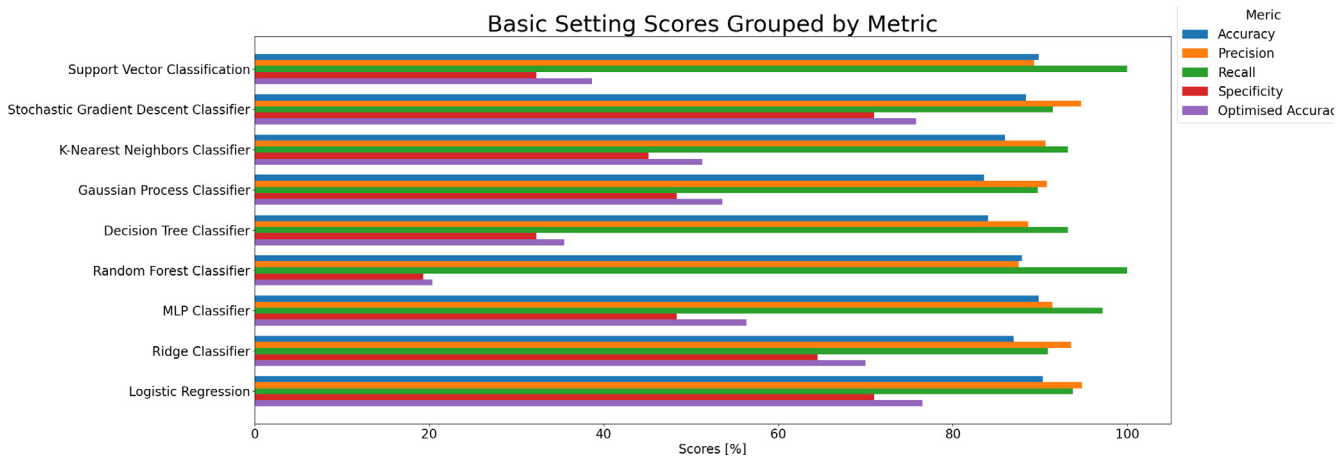

Fig. 3. Comparison of the performance of the standard models of the Scikit-learn library grouped by the metrics accuracy (blue), precision (orange), recall (green), specificity (red) and optimised accuracy (purple). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

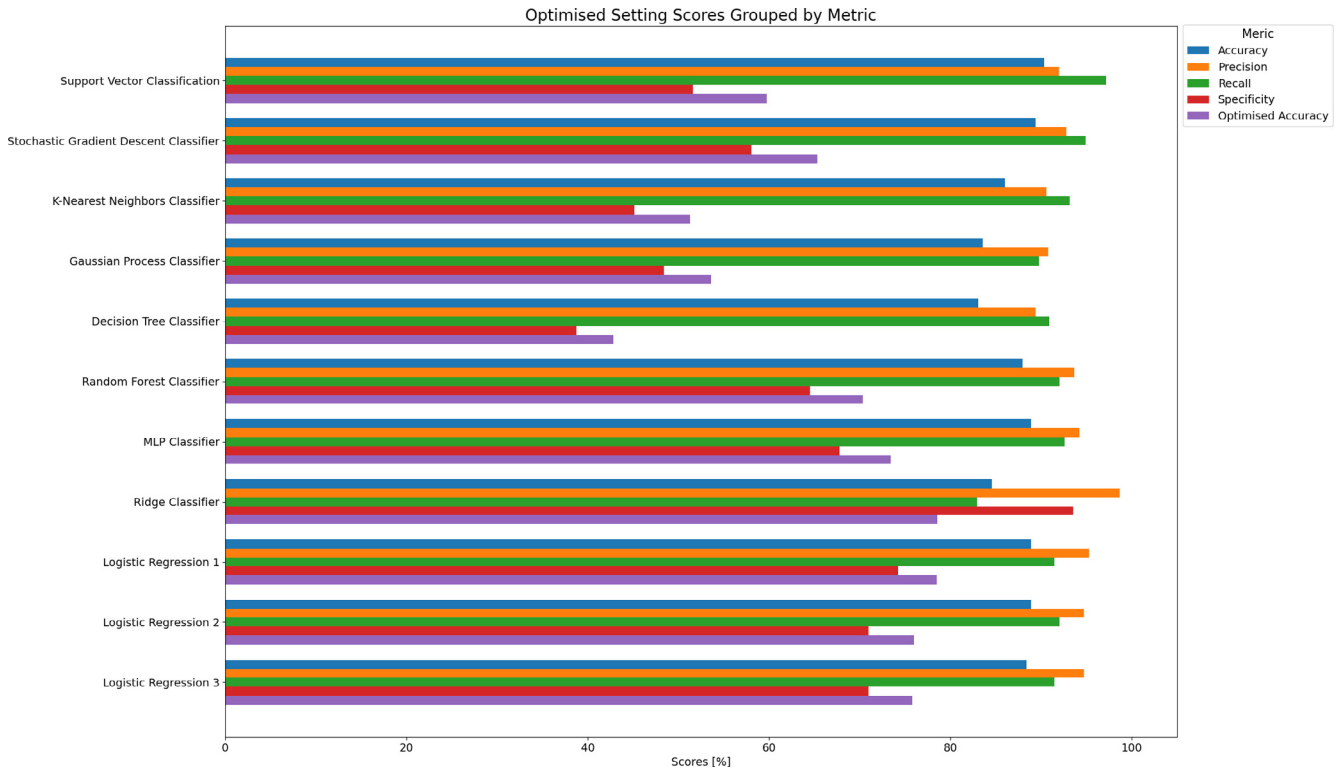

Fig. 4. Comparison of the performance of the optimised models of the Scikit learn library grouped on the metrics accuracy (blue), precision (orange), recall (green), specificity (red) and optimised accuracy (purple). For the optimized settings see Supplementary Materials 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

in total 16 models for all features and 15 models for the features without standard deviation. For these tests, the training and validation data were combined into one large training set to train the models. They were then validated with the test set (Supplementary Material 1). The best 16 stacking models of the approach with all features were also examined with reduced features obtained by different algorithms. The used algorithms were e.g. select from model, select k best or the recursive feature elimination, which were all used with different settings (Supplementary Material 1). A total of 13 reduced feature sets were investigated, which were obtained through the algorithms. The tests were performed with the training set and tested against the validation set. Only the 16 best models of the approach with all features were checked, since the algorithms, including the standard deviations, retain all possible features. In this way, 208 (16 best Models $\times$ 13 reduced feature) additional examinations were carried out. For the further investigations of these models, only those were selected that showed an improvement in their optimised accuracy (Supplementary Material 1). The models with reduced features were not significantly better or worse than the models with all features. There are many approaches where more features lead to better ML models. This can be seen in the work of Siegismund *et al.* [16] Nevertheless, this does not always have to be

the case, as some features, unless excluded, could potentially lead to noisy data.

Since the final model is intended to be integrated in a work process where one wants to find strong bioactive substances. This means that only substances that are really bioactive should be predicted as bioactive and that false positive samples should be kept as low as possible (high precision). In addition, controls should be correctly predicted (high specificity) in a searching process where many samples are tested. It is more important to get only strong hits and it can be neglected if some minor bioactive substances are predicted as controls (small recall). As the performance of the models strongly depends on the selected training data and usually the whole/unfiltered data set is used for the final model, one wants to get a better estimate of the performance of the final model. For this purpose, after the further tests (training with the large training set and tests with the test set), the models were finally validated by a 10-fold CV with the full data. In this way, the models are trained with 9-folds (90 % of the data) and are thus closer to the final models. When selecting the models for the final CV, not only the optimised accuracy was considered, but also the precision and specificity. Only models with an optimised accuracy of over 80 % or with precision and specificity over 98 % were used.

The 10-fold CV was performed with a fixed random state, so that all models can be compared. A total of 84 models (with and without reduced features) were examined with the final CV.

For the choice of the final model, only models with a standard deviation below 4 % for all metrics were considered. This left only nine candidates out of the original 84 models. The best model, Stack 1, was a stacking model with support vector classifier, SGD classifier and KNN Classifier as basic models with basic settings and logistic regression as the final estimator with the optimised settings from Logistic Regression 2 (Supplementary Material 1). The model was trained with reduced features obtained by recursive feature elimination with the model logistic regression. The settings were from Logistic Regression 2 except for the tolerance (tol) which was set to tol = 0.001. This classifier Stack 1 was chosen because it has the highest optimised accuracy, the smallest standard deviation in optimised accuracy and the smallest standard deviations in specificity and precision (see Fig. 5).

3.2. Partial least square regression as a classifier

Since partial least square algorithms are one of the most widely used algorithms in the field of *omics* [35,36], an approach using PLSR was tried. Also because it is similar to PCA, which is a suitable method to detect classes in data. The PLSR is not designed as a classifier in the scikit-learn library [34]. Moreover, it does not output discrete numerical values, but only continuous numbers. Thus, a threshold value was used to classify the data. Since GSCV cannot be used for classification problems with PLSR, a separate GSCV was written. This sought for the most adjustable parameters of the PLSR and a threshold at which the best classification results for optimised accuracy, precision or specificity were obtained. For precision and specificity, we looked for the maximum values that were less than 100 %. Since the search is time intensive, all cores of the processor were used in the computation through the multiprocessing library [41]. In addition, the principal components were divided into several sets and analysed separately on two computers. For the own GSCV, 5-folds and a fixed random state were used again. The results from each set were then used for further investigation. For the outcomes that had the best optimised accuracy, precision and specificity, an additional grid search with finer thresholds was performed (Supplementary Material 2).

In addition to the own GSCV, a simple search for the number of principal components was carried out. Here, we looked which number of principal components achieved the highest coefficient of determination. The result was then used to search for a suitable threshold value in the own GSCV. Since the number of iterations or the tolerance was the same for all previous results, these settings were used. The search was only for the appropriate threshold value (Supplementary Material 2).

All parameter combinations, in total 44 models, were then checked by training the models with the training set and testing them with the validation set. For further investigations, models were taken that had an optimised precision greater than 80 % or where precision and specificity were above 90 % with a recall above 60 %. Again, a large data set from the training and validation set was used to train the classifier. The model was verified with the test set (Supplementary Material 2).

For the final test, the entire data set was used for a CV with 10-folds. The models that were tested also had either an optimised accuracy of over 80 % or specificity and precision of over 90 % and recall of over 60 %. Altogether seven models were examined in this way. The best model, PLSR 1, was the one with a principal component number of 13 and a threshold of 0.65. Which was the result of the simple principal component search with the aim of achieving the highest coefficient of determination. It had the highest value in optimised accuracy with the smallest standard deviation. It also had a smaller standard deviation for specificity and precision than other models with comparable optimised accuracy (Fig. 6).

3.3. Artificial neural networks

The ANNs were all created with the GPU supporting library of Tensorflow [40]. Only dense layers were used for the architectures, as other types of layers would go beyond the scope of this work. We have tested different combinations of number of layers and the units they contain. In addition, as with PLSR, a threshold was used where a sample belongs to a class, since ANNs only output probabilities of class membership.

The training of the ANNs was carried out with the training data and tested with the validation data. Different numbers of layers were systematically examined. For a certain number of layers, loops were used to analyse which number of units is the best per layer. Callbacks were used to optimise the training in the loop. The training was designed to stop after 100 training epochs, when the accuracy in the validation set no longer increased, and save the best model. This best model is then loaded to check which threshold gives the highest optimised accuracy. The model must be reloaded because the old model was over trained with the 100 additional epochs. After the loop has been run, the settings are obtained that have achieved the highest optimised accuracy. These are used to check which thresholds give the highest optimised accuracy, precision and specificity. Further examinations were also carried out with the ANNs using the large training set and the test set for validation. For this, the different models were tested with the different settings obtained from the loop runs. In the case of ANNs, 72 different architectures were tested (Supplementary Material 3).

In order to be able to compare the ANNs better with the other models, the final test was also carried out under the same criteria. Even though it is unusual in the field of ANNs, there are a few studies that have used a CV [23,46–49]. Models that over fitted in the CV were rejected, this left 16 models (Supplementary Material 3). The best model, ANN 1, was one that consisted of 6 layers with 5,5,5,7,8 and 1 unit in the layers. Only 47 epochs were needed to train the model, and the threshold for the best result was 0.83 (see Fig. 7).

If you compare all the previous models (see Fig. 8), you can see that the ANN has the best performance with 95 % Optimised Accuracy. While the stacking model (87 % Optimised Accuracy), the

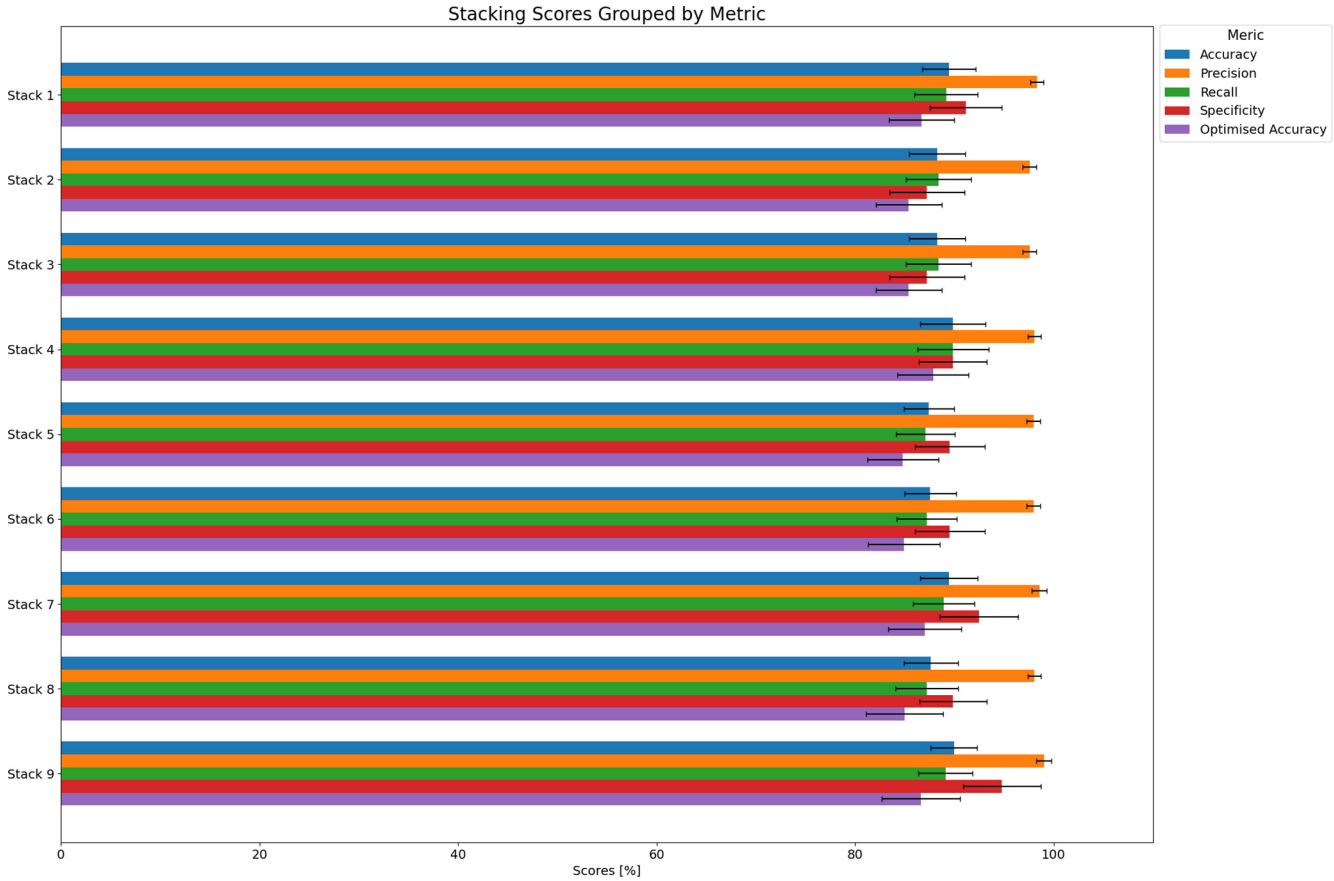

Fig. 5. The best 9 stacking models of the cross-validation where the standard deviations were less than 4 %. The performance of the models is grouped by the metrics accuracy (blue), precision (orange), recall (green), precision (red) and optimised accuracy (purple). For the structure of the stacking models see Supplementary Material 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

classic ML approach, and the PLSR (86 % Optimised Accuracy) are similarly good. The ANN has better scores in all the used metrics. Looking at the optimised accuracy in Fig. 8, we see that even the outliers in the boxplot are within the range of the other models.

3.4. Novelty detection

A well-known problem with supervised ML is that the models cannot handle new, unknown patterns. They make a prediction for new patterns, but assign them to one of the known classes. But it is precisely the target to find novel patterns in the search for new, unknown bioactive compounds. So the idea was to use a kind of outlier test. In a routine work process with a fixed pipeline, it is not possible to use an outlier test. This is because the alpha error would have to be adjusted for each outlier test performed. Also, one would need to know how many tests are performed in total to correctly adjust the alpha error. Since this is not possible, another solution must be chosen. In the field of ML, there is such a solution, namely ND [50].

In contrast to the classic outlier test, which is an unsupervised procedure, ND is a supervised procedure. It learns patterns of the data and compares new samples with the learnt pattern. A density function is used to determine whether or not new samples belong to the known data set.

The scikit-learn library offers two different NDs, the OneClassSVM and the LocalOutlierFactor (LOF) [34]. For our approach, two novelty detectors were used, one trained with the data from the control groups and one trained with the bioactive samples. A new sample does only account as “novel” if it is classified as such by both detectors. As for the classifier, the same training, validation and test set was used for verification. One detector was trained only with the controls from the training set and the other only with the compounds. When checking the ND, it was important that as few samples as possible were counted as outliers. The focus was on the control samples, which should be kept as good as possible, because outliers are handled as possible unknown bioactivities. This is because outliers are different from the bioactive compounds, but also from the control groups. Since they are different from the controls, it means that they are bioactive.

For the first analysis of the models, the models were trained with the training set and checked with the validation set. This showed that the LOF performed better than the OneClassSVM. In LOF only 5 % of the validation data were classified as outliers in the best case; while in OneCLassSVM it was 59 % (Supplementary Material 4). Therefore, further tests were only carried out with the LOF. For this purpose, the data set for training was extended with the validation set and tested against the test set. The best setting was obtained with a LOF for the control groups with the parameters p = 1 and n_neighbors = max. For the LOF trained with the compounds, the best setting was p = 1 and n_neighbors = 1000. In these settings, there were only 9.66 % outliers, of which all belonged to the bioactive compounds (Supplementary Material 4). The LOF did keep all the controls.

3.5. Validation with independent data

For the final verification of the models, two new data sets were used, with compounds that were previously unknown to the mod-

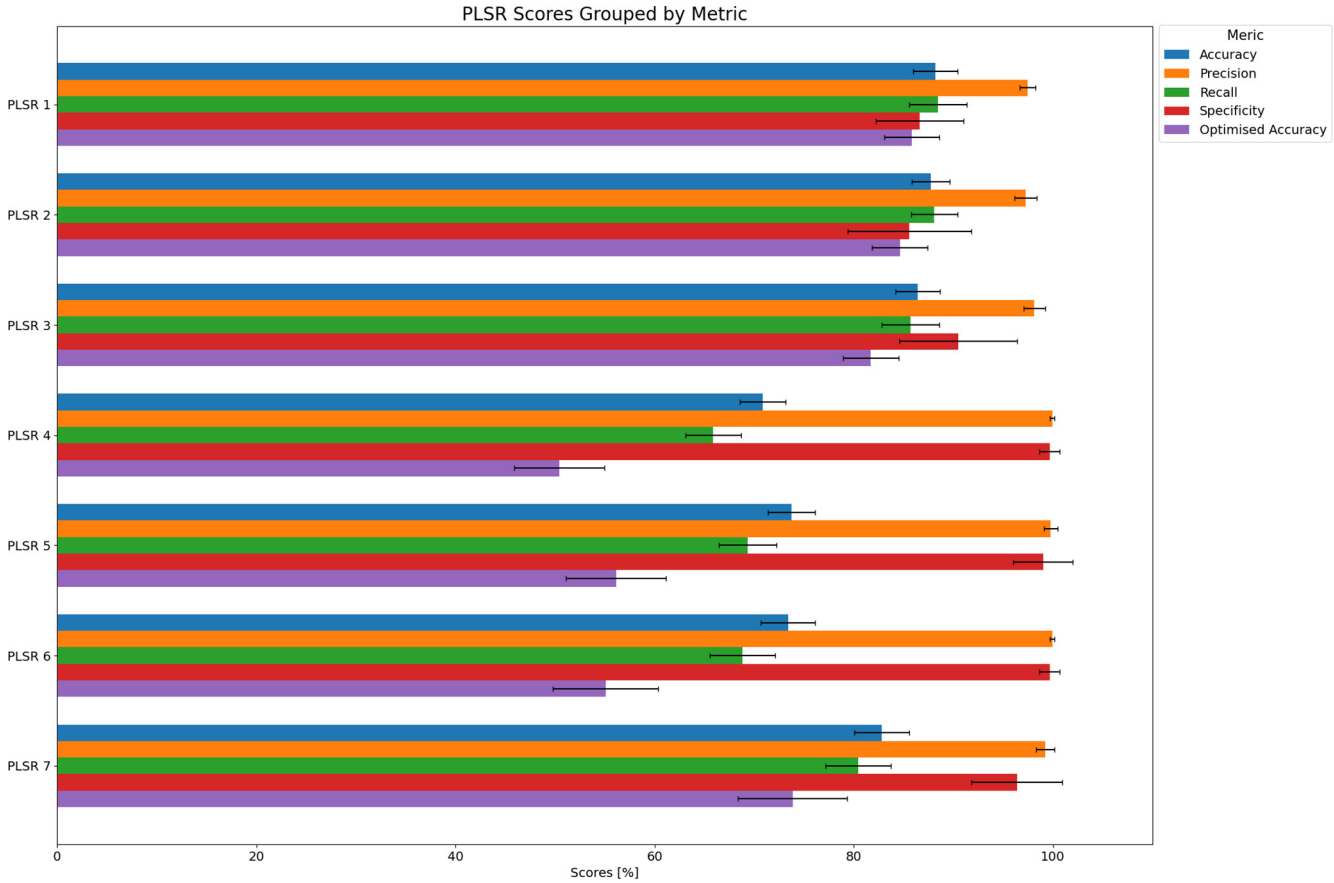

Fig. 6. The best 7 PLSR models in the cross-validation. The performance of the models is grouped on the metrics of accuracy (blue), precision (orange), recall (green), precision (red) and optimised accuracy (purple). For the settings of the PLSR models see Supplementary Material 2. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

els, except for mebendazole. One data set contained cell cycle interfering drugs (mebendazole, nicolsamide and cladribine) at a fixed concentration of 10 μM, the other set consisted staurosporine in different concentrations (10 nM and 30 nM). The best models of the above presented approaches were tested without and with upstream ND. This was done to test how the models generally handle new, unknown patterns and how the ND performs.

During the first evaluation, which was without ND (Fig. 9), ANN performed worst. It had a performance for both data sets that is more like a random predictor. Looking at the PLSR, almost all samples were predicted as controls for the cell cycle interfering drugs. For the staurosporine data, the predictions were much better and within an acceptable range. In contrast, the stacking model performed well for both data sets. The stacking model is best suited to search for new compounds, as it was able to handle unknown data. In addition, the accuracy of the predictions is very high, which is ideal for the screening approach. It achieved an optimised accuracy of 78 % for the cell cycle data. The precision is 97 % and the specificity is 91 %. For the staurosporine data, the model achieved an optimised accuracy of 92 %. The precision is 96 % and the specificity is 90 %. All values are ideal for a screening approach (Supplementary Material 5).

The models were thereafter tested with an upstream ND. All models benefited from the detector and their performance increased (Fig. 10). In ANNs the performance improved for the bioactivity predictions, as these are recognised by the ND. The controls remain in the pipeline and are predicted by the ANN, so the predictions here are the same as above and are more like those of a random predictor. The PLSR benefited greatly from the ND because the predictions of cell cycle interfering agents perform well. The unknown drugs are detected through the ND and the controls remain in the pipeline. The controls are predicted by the PLSR, which works well. For staurosporines, the performance also improved thanks to the ND. The performance of the stacking model also increased and achieved the best results in the predictions. For the cell cycle data, the optimised accuracy increased by 15 % to 93 %. Specificity and precision remained at their values, but recall improved from 82 % to 100 %. For the staurosporine data, the performance of the model remained the same (Supplementary Material 5). There is a great advantage in using the ND as all models are improved and it helps to handle known bioactivities. This is clearly visible in the PLSR and ANN and their predictions for bioactivity.

4. Discussion

Thanks to HCS in conjunction with the suitable data analysis workflow, there is a new possibility to discover bioactive substances quickly and easily. This involves a lot of data that can have many variables. In many studies, these large data sets are analysed with classical statistics, but ML is finding more and more applications such as the classification of morphology [22,23,29,51–55], mechanism of action [26], modes of action [27] or clustering [28–30,56–57]. These classifications require deeper knowledge of cell biology. To best of our knowledge, there are no ML applications that are suitable for simple hit detection of whether samples are biologically active or not. In this work, we have systematically tested different ML models for use as binary classifier. The used data were HCS data acquired following the cell painting protocol

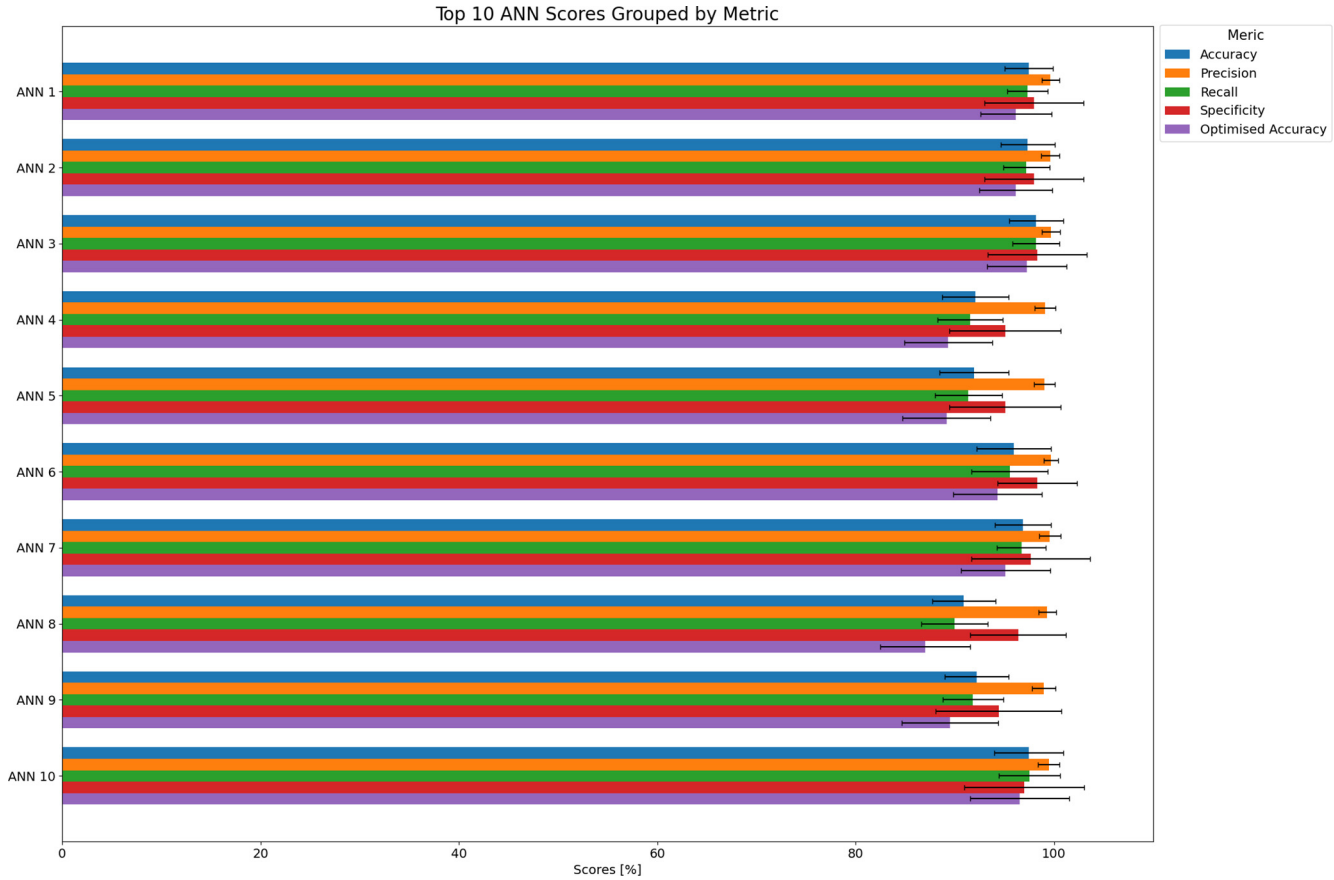

Fig. 7. The best 10 ANN models in the cross-validation. The performance of the models is grouped on the metrics of accuracy (blue), precision (orange), recall (green), precision (red) and optimised accuracy (purple). For the construction of the ANN see Supplementary Material 3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

[42]. In total, we had 645 compounds with known pharmaceutical activities, which were labelled as the bioactive class. The other class, labelled as the control group, were the control groups of the HCS measurements. The data were pre-processed (Z-transform and Grubbs outlier test) and divided into three data sets (train, validate and test set). Different metrics were used as quality criteria for the models (accuracy, precision, recall, specificity and optimised accuracy [37]).

The first models examined were classical ML models. For the basic settings of the classical models, the SGD classifier had the best result. In the default settings, it corresponds to a linear SVM and the achieved results fit to other SVM models in a similar task area [55]. After the hyperparameter optimisation, the optimised accuracy decreased. Since the hyperparameters are determined with a CV, the settings are taken that gave the best result in the average of all tests. In the case of the used test set, it may turn out worse. However, it is usually more stable for a larger variety of other data sets. We found that the simple models Logistic Regression, Ridge Classifier, MLP Classifier or Random Forest Classifier produced the best results. However, since the results were not yet satisfactory, ensemble methods were tested. This showed that voting models, whether hard or soft voting, did not improve the performance. This is probably due to the fact that the models made the same predictions and thus no additional information was gained from the variation. The next approach was to use stacking models, which gave the best results. The achieved performances are so good that the models can be used for screening. The best model was one that was trained with reduced features. However, it was only slightly better than the best model that used all features (Supplementary Material 1). As a rule, it is assumed that more features are more useful [16], but more features are only useful if they really provide information and not noise. Since feature selection has helped somewhat, many features are probably orthogonal to each other or cause unnecessary noise. Ensemble models often perform well because they use the information of several models and thus models help each other with their weaknesses.

The second model examined was the PLSR. Actually, it is only used for regression problems but in this work it was used for classification. This was possible through a threshold that served to separate the class membership. All results greater than the threshold were classified as bioactive and all results less than or equal to the threshold were classified as controls. We decided to test PLSR because it is the most frequently used algorithm in the different *omics* diciplines [35,36]. Also, good results were obtained with PCA for HCS [58], which is very similar to PLSR. For hyperparameter optimisation, two different approaches were evaluated. One was a search for the principal components that should give the largest coefficient of determination and the other a self-written grid search CV, looking for the highest optimal accuracy, precession or specificity. The best result was obtained with the model that had the largest coefficient of determination. One possibility for the result is that the model was forced to have the smallest average error. Which means that it is generally forced to make the smallest deviations in the predictions. This allows the model to better generalise, which is an important attribute in ML.

The last model examined was an ANN, which consisted only of dense layers. The best model found consisted of 6 layers with

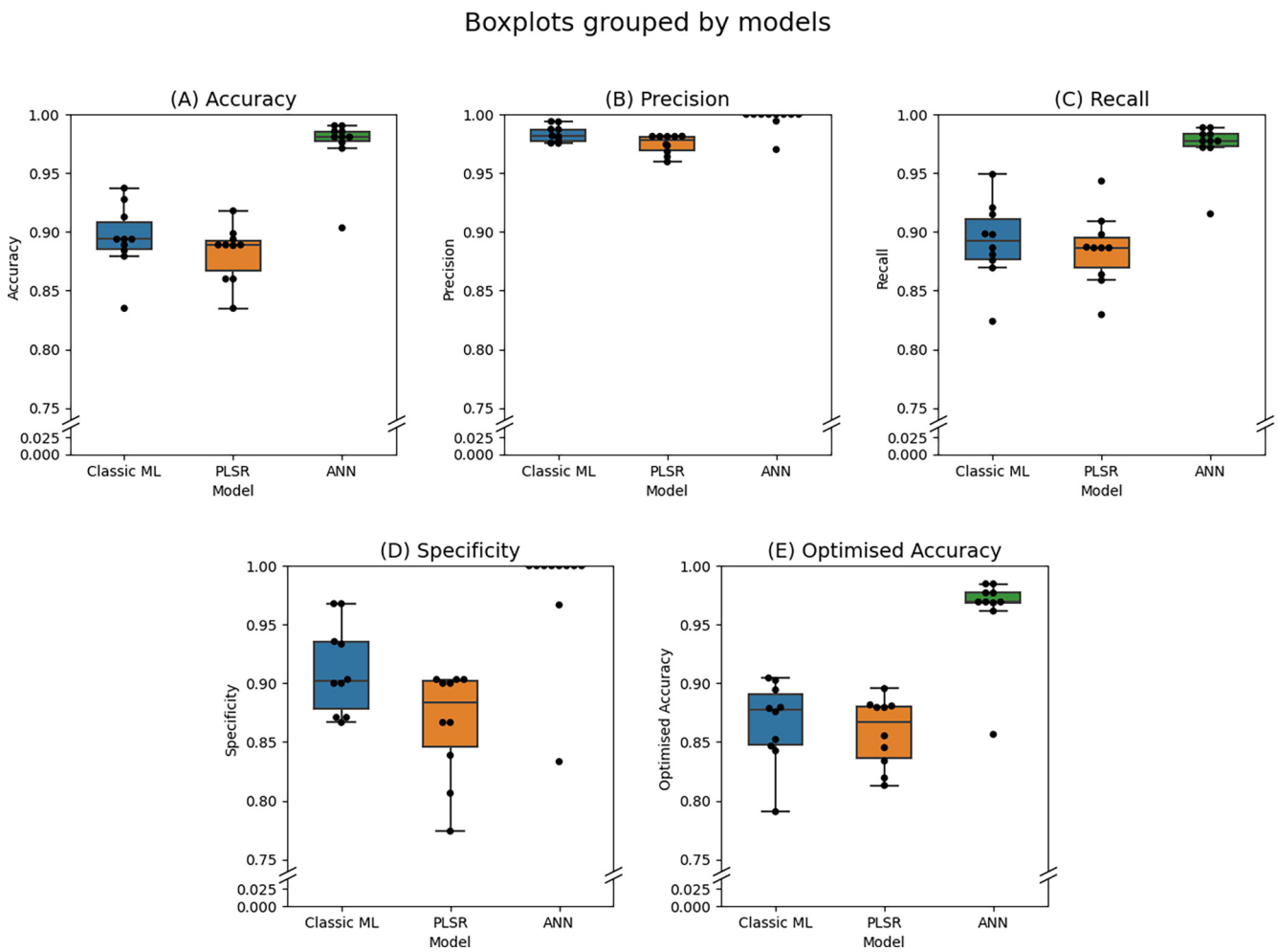

Fig. 8. Comparison of the different model approaches in their achieved scores for (A) Accuracy, (B) Precision, (C) Recall, (D) Specificity and (E) Optimised Accuracy.

5,5,5,7,8 and 1 units. The ANN had the best performance of the found models with the highest optimised accuracy, which was 96 %. This is probably caused by the very flexible architecture of the ANN, which can be adapted very well to different problems. ANNs can learn patterns very well, but one has to be careful. Because they are very good at learning patterns, they can quickly tend to overfit [59]. They may learn very specific patterns that are disadvantageous for generalising. As a result, they may not predict new, unknown data as well as the others. With ANNs, one must therefore pay close attention to the balancing act between very good predictions on the training data and the ability to generalise and predict new data well. On the other hand, learning specific patterns can also be beneficial, as with samples that don't vary much or don't show batch effects. This behaviour can be helpful during routine investigations.

Since the classifier is to be used to search for new active compounds, a ND was tested to be connected upstream of the classifier. It should enable the model to deal with new and unknown patterns, similar approaches existed in other HCS applications [24,28–30]. Patterns should be filtered out that are very different from the known patterns of bioactive substances and controls. Since such patterns differ from the controls, they are also bioactive. Two detectors were trained, one with the active substances and the other with the controls. If a sample was detected as an outlier with both, it was first counted as an outlier and was classified as unknown bioactivity. Two NDs were tested that were in the scikit-learn library, the OneClassSVM and the LOF. For HCS data, the best ND was the LOF. Here, most samples could be kept and, most importantly, all controls could be maintained. The most important thing is that the controls are maintained. Since all outliers are classified as bioactive. So, it is not tragic for bioactive substances to be detected via the ND instead by the classifier. The LOF is a general algorithm that uses a density function to determine the outliers. Samples that have a density lower than a certain number of neighbours are counted as outliers. It is a function that compares how much a new sample differs from its neighbours. It could be counted as one of the Nearest Neighbour algorithms and they work well for similar problems [27,28].

For the final testing of the models, two data sets were used with completely new compounds which are unknown to the models. One of the data sets contained drugs that affect the cell cycle (mebendazole, nicolsamide and cladribine) and the other contained staurosporines in different concentrations (10 nM and 30 nM). Two approaches were tested, one with and one without ND. Without the ND, the stacking model performed best in the predictions. The PLSR predominantly predicted all samples for the cycle data as controls. The staurosporine data predicts the PLSR well. The ANN was very poor and made predictions almost randomly. The ANN's poor discrimination is because it has learned patterns that are too specific. For biological data, batch problems are common [60]. Results of experiments that are conducted on different days with other batches of reagents, cells and multiwell plates can slightly differ from previous experiments. The ANN has learned the patterns of the training data too specifically

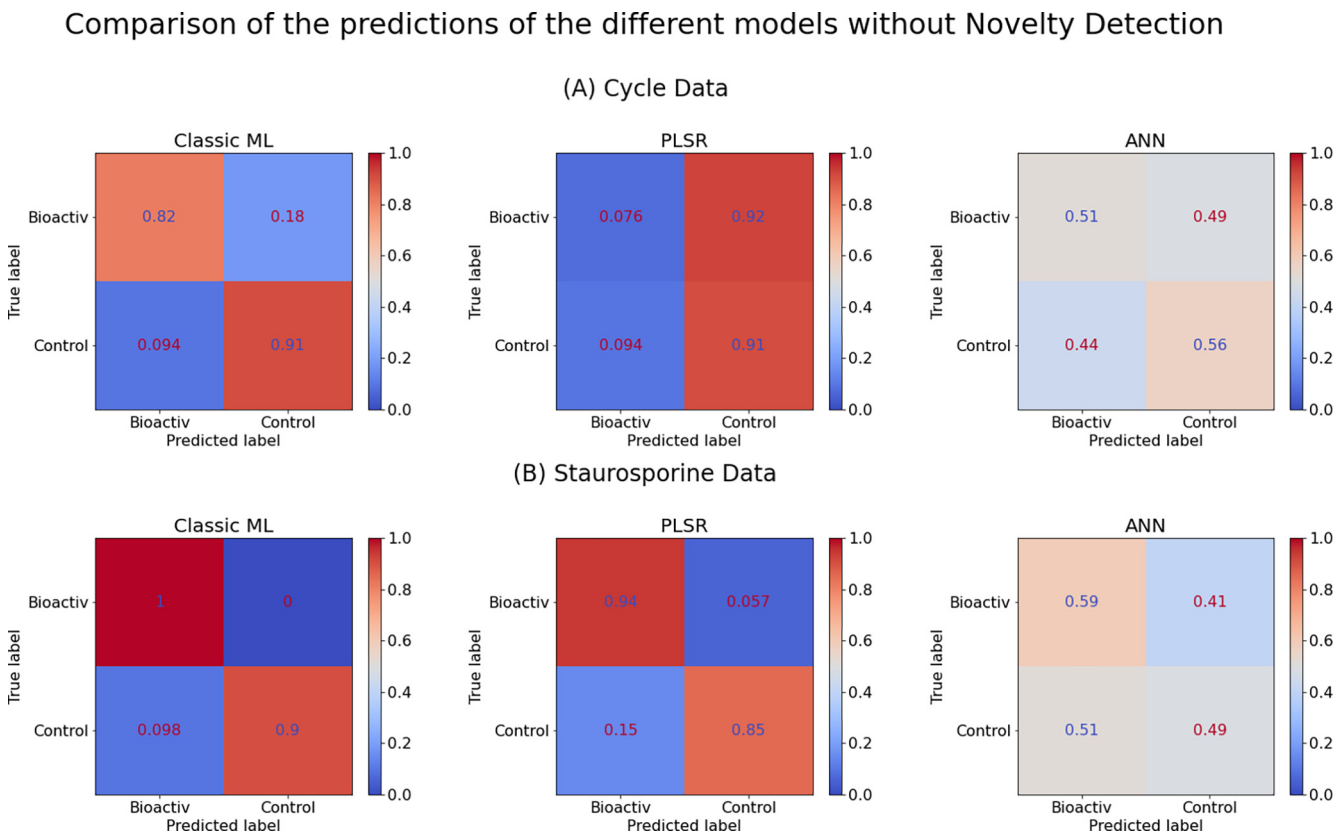

Fig. 9. Comparison of the predictions of the different models, without Novelty Detection, for (A) Cell Cycle Data and (B) Staurosporine Data.

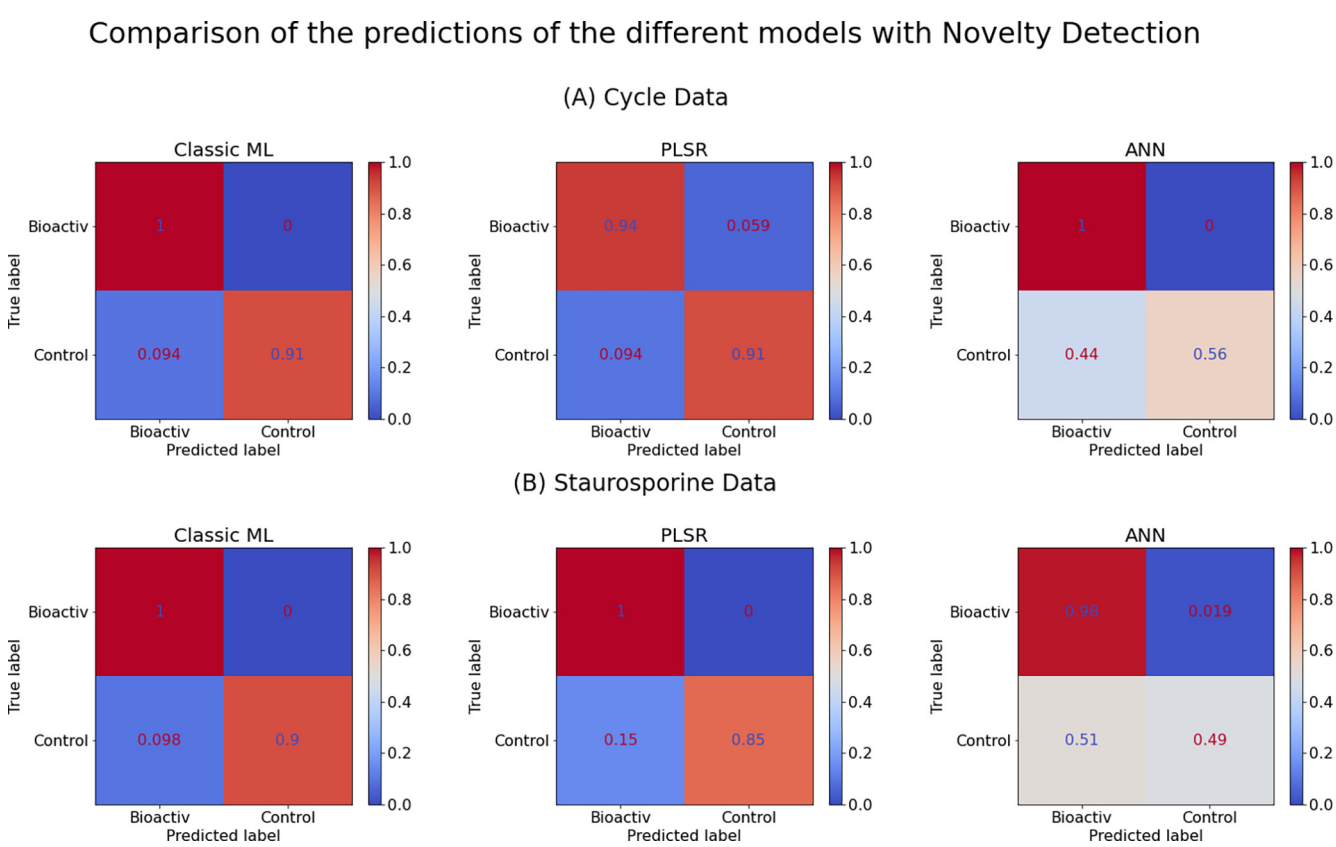

Fig. 10. Comparison of the predictions of the different models, with Novelty Detection, for (A) Cell Cycle Data and (B) Staurosporine Data.

and can no longer handle the deviations due to the batch effect. This is evident in the controls, which cannot be predicted correctly. The new bioactive substances are different from the previously learned ones, which is why it cannot handle them properly either. The ANN is not able to generalise well. With the PLSR, you can see that it is also worse at dealing with unknown patterns. The cell cycle data is different from the previous learned patterns and for PLSR it is probably closer to the controls. Unlike the staurosporines, these are closer to the bioactive compounds for the PLSR and that is why they are predicted well. The stacking model shows very good results. It shows that it can handle unknown patterns and generalise well. This is probably due to the used base models, as the SVM [22,26,29,52–55] or KNN [27,52] were often used in other works for similar problems. With the ND upstream, the performance of all models improved. Again, the stacking model was the best. The performance of the PLSR has greatly improved thanks to the upstream ND and comes within the range of the stacking model. The ANN is also the worst model here, as it still predicts the control groups like a random predictor. The test with the ND shows that the ND filters out patterns that are unknown to the models well. The reasons why the ANN is also bad here are the same as above. Since the ND only filters out bioactive substances, the controls still enter the model for prediction. As explained above, the ANN has learned the patterns of the controls too specifically and can no longer generalise to new controls. But the tests with the ND also show how well it supports the models and makes them more robust. The predictions of all models became better with the ND.

The results of the models also provide an insight into the structure of the data. Usually, non-linear models such as SVM [22,24,26,29,52–55] or Random Forest [25] are used for HCS data. Here too, non-linear models have been found that perform well, such as Random Forest. Two other models that allow to approximate non-linear relationships are MLP and the ANNs, which also performed well. This suggests non-linear relationships in the data. In this work, however, there were also linear models, such as PLSR, ridge regression or logistic regression, that performed good. It gets interesting when we look at the best model, the stacking model. In this two basic models are non-linear (SVM and KNN) and the other linear (SGDC). These mainly non-linear models produced an output that served as input for the final model (Logistic Regression), which is a linear model. This shows that HCS data are very complex and have an ambivalent behaviour, because they can be solved well with linear models and non-linear ones.

It should be noted that the results are only very certain for the data used here. The evaluation and verification of the stability of the stacking model requires a lot of additional data. While the hit detector is more robust thanks to the ND, one should consider the hit detector as a complementary analysis method for the moment.

When the results are summarised again, it is clear that the Stacking Model is best suited for screening. It has the best ability to generalise and can also deal well with unfamiliar patterns. In addition, the ND was found to serve very well and to support and improve the models in their predictions. The stacking model in combination with the ND provides a solid pipeline to analyse HCS data whether they are biologically active or not. The current results show that ANNs with dense layers are not well suited for a screening/untargeted approach. Perhaps other architectures would be better suited. However, the good results of the ANNs in modelling suggest a good application in a targeted approach. In which the samples are known, do not vary greatly and do not have batch effects.

For future work, it would be worth checking how the stacking model performs for other HCS data that have different features. Of course, one would have to adapt the hyperparameters of the classifiers to the other data. The stacking model might be a way to improve the existing ML models that only use the SVM for morphology or MoA prediction. Since the performance of the basic models, where an SVM was also present, has improved in the stacking approach. In addition, a deeper investigation of the HCS data would be interesting, as they can be predicted well with linear and non-linear models. As a better understanding of the data leads to better models, and therefore it can improve the development of drugs.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2022.09.023.

References

[1] Helfer M et al. The root extract of the medicinal plant Pelargonium sidoides is a potent HIV-1 attachment inhibitor. PloS one 2014;9:e87487.

[2] Kremb S, Müller C, Schmitt-Kopplin P, Voolstra CR. Bioactive potential of marine macroalgae from the central red sea (Saudi Arabia) assessed by high-throughput imaging-based phenotypic profiling. Marine drugs 2017;15. https://doi.org/10.3390/md15030080.

[3] Mueller C et al. Advanced identification of global bioactivity hotspots via screening of the metabolic fingerprint of entire ecosystems. Sci Rep 2020;10:1319. https://doi.org/10.1038/s41598-020-57709-0.

[4] Furner-Pardoe J et al. Anti-biofilm efficacy of a medieval treatment for bacterial infection requires the combination of multiple ingredients. Sci Rep 2020;10:12687. https://doi.org/10.1038/s41598-020-69273-8.

[5] Schmitt-Kopplin P et al. Systems chemical analytics: introduction to the challenges of chemical complexity analysis. Faraday discussions 2019;218:9–28. https://doi.org/10.1039/c9fd00078j.

[6] Smith K, Horvath P. Active learning strategies for phenotypic profiling of high-content screens. J Biomol Screen 2014;19:685–95. https://doi.org/10.1177/1087057114527313.

[7] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44. https://doi.org/10.1038/nature14539.

[8] Carpenter AE et al. Cell Profiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol 2006;7:R100. https://doi.org/10.1186/gb-2006-7-10-r100.

[9] Ko M et al. Screening of FDA-approved drugs using a MERS-CoV clinical isolate from south korea identifies potential therapeutic options for COVID-19. Viruses 2021;13. https://doi.org/10.3390/v13040651.

[10] Schulze CJ et al. "Function-first" lead discovery: mode of action profiling of natural product libraries using image-based screening. Chem Biol 2013;20:285–95. https://doi.org/10.1016/j.chembiol.2012.12.007.

[11] Shinozawa T et al. High-fidelity drug-induced liver injury screen using human pluripotent stem cell-derived organoids. Gastroenterology 2021;160:831–846.e10. https://doi.org/10.1053/j.gastro.2020.10.002.

[12] Sridhar, S. *et al.* High-Content Imaging to Phenotype Antimicrobial Effects on Individual Bacteria at Scale. *mSystems* **6**, doi: 10.1128/mSystems.00028-21 (2021).

[13] Chao JT, Roskelley CD, Loewen CJR. MAPS: machine-assisted phenotype scoring enables rapid functional assessment of genetic variants by high-content microscopy. BMC Bioinform 2021;22:202. https://doi.org/10.1186/s12859-021-04117-4.

[14] Kraus OZ et al. Automated analysis of high-content microscopy data with deep learning. Mol Syst Biol 2017;13:924. https://doi.org/10.15252/msb.20177551.

[15] Piccinini F et al. Advanced cell classifier: user-friendly machine-learning-based software for discovering phenotypes in high-content imaging data. Cell Syst 2017;4:651–655.e5. https://doi.org/10.1016/j.cels.2017.05.012.

[16] Siegismund D, Fassler M, Heyse S, Steigele S. Benchmarking feature selection methods for compressing image information in high-content screening. SLAS Technol 2021. https://doi.org/10.1016/j.slast.2021.10.015.

[17] Tao CY, Hoyt J, Feng Y. A support vector machine classifier for recognizing mitotic subphases using high-content screening data. J Biomol Screen 2007;12:490–6. https://doi.org/10.1177/1087057107300707.

[18] Boland MV, Murphy RF. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. Bioinformatics (Oxford, England) 2001;17:1213–23. https://doi.org/10.1093/bioinformatics/17.12.1213.

[19] Boutros M, Heigwer F, Laufer C. Microscopy-based high-content screening. Cell 2015;163:1314–25. https://doi.org/10.1016/j.cell.2015.11.007.

[20] Neumann B et al. High-throughput RNAi screening by time-lapse imaging of live human cells. Nature Methods 2006;3:385–90. https://doi.org/10.1038/nmeth876.
[21] Bakal C, Aach J, Church G, Perrimon N. Quantitative morphological signatures define local signaling networks regulating cell morphology. Science (New York N.Y.) 2007;316:1753–6. https://doi.org/10.1126/science.1140324.
[22] Misselwitz B et al. Enhanced cell classifier: a multi-class classification tool for microscopy images. BMC Bioinform 2010;11:30. https://doi.org/10.1186/1471-2105-11-30.
[23] Horvath P, Wild T, Kutay U, Csucs G. Machine learning improves the precision and robustness of high-content screens: using nonlinear multiparametric methods to analyze screening results. J Biomol Screen 2011;16:1059–67. https://doi.org/10.1177/1087057111414878.
[24] Sommer C, Hoefler R, Samwer M, Gerlich DW. A deep learning and novelty detection framework for rapid phenotyping in high-content screening. Mol Biol Cell 2017;28:3428–36. https://doi.org/10.1091/mbc.E17-05-0333.
[25] Rose F et al. Compound functional prediction using multiple unrelated morphological profiling assays. SLAS Technol 2018;23:243–51. https://doi.org/10.1177/2472630317740831.
[26] Berg EL, Yang J, Polokoff MA. Building predictive models for mechanism-of-action classification from phenotypic assay data sets. J Biomol Screen 2013;18:1260–9. https://doi.org/10.1177/1087057113505324.
[27] Reisen F et al. Linking phenotypes and modes of action through high-content screen fingerprints. Assay Drug Development Technol 2015;13:415–27. https://doi.org/10.1089/adt.2015.656.
[28] Manning S, Shamir L. CHLOE: a software tool for automatic novelty detection in microscopy image datasets. J Open Res Softw 2014;2. https://doi.org/10.5334/jors.bg.
[29] Yin Z et al. A screen for morphological complexity identifies regulators of switch-like transitions between discrete cell shapes. Nature Cell Biol 2013;15:860–71. https://doi.org/10.1038/ncb2764.
[30] Yin Z et al. Using iterative cluster merging with improved gap statistics to perform online phenotype discovery in the context of high-throughput RNAi screens. BMC Bioinform 2008;9:264. https://doi.org/10.1186/1471-2105-9-264.
[31] Caicedo JC et al. Data-analysis strategies for image-based cell profiling. Nat Methods 2017;14:849–63. https://doi.org/10.1038/nmeth.4397.
[32] Scheeder C, Heigwer F, Boutros M. Machine learning and image-based profiling in drug discovery. Curr Opin Syst Biol 2018;10:43–52. https://doi.org/10.1016/j.coisb.2018.05.004.
[33] Sommer C, Gerlich DW. Machine learning in cell biology - teaching computers to recognize phenotypes. J Cell Sci 2013;126:5529–39. https://doi.org/10.1242/jcs.123604.
[34] Pedregosa F et al. Scikit-learn: machine learning in python. J Mach Learn Res 2011;12:2825–30.
[35] Gromski PS et al. A tutorial review: Metabolomics and partial least squares-discriminant analysis—a marriage of convenience or a shotgun wedding. Anal Chim Acta 2015;879:10–23. https://doi.org/10.1016/j.aca.2015.02.012.
[36] Mendez KM, Broadhurst DI, Reinke SN. The application of artificial neural networks in metabolomics: a historical perspective. Metabolomics 2019;15:142. https://doi.org/10.1007/s11306-019-1608-0.
[37] Ranawana, R., Palade, V., Optimized Precision - A New Measure for Classifier Performance Evaluation. In *2006 IEEE International Conference on Evolutionary Computation* (IEEESunday, July 16, 2006), pp. 2254–2261.
[38] Anaconda Inc. *Anaconda Software Distribution* (Anaconda Inc., 2021).
[39] Loizides F, Schmidt B. Positioning and Power in Academic Publishing: Players, Agents and Agendas. Electronic Publishing (IOS Press; 2016.
[40] . *TensorFlow* (Zenodo 2021.
[41] van Rossum G, Drake FL. Python 3. Scotts Valley, CA: CreateSpace; 2021.
[42] Bray M-A et al. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. Nat Protoc 2016;11:1757–74. https://doi.org/10.1038/nprot.2016.105.
[43] Kremb S, Voolstra CR. High-resolution phenotypic profiling of natural products-induced effects on the single-cell level. Sci Rep 2017;7:44472. https://doi.org/10.1038/srep44472.
[44] Vabalas A, Gowen E, Poliakoff E, Casson AJ. Machine learning algorithm validation with a limited sample size. PloS one 2019;14:e0224365.
[45] Erwin Kupczyk. Benchmarking AI Workflows for Hit Detection in High-Content Screening, 2022.
[46] Hu MY, Zhang G, Jiang CX, Patuwo BE. A cross-validation analysis of neural network out-of-sample performance in exchange rate forecasting. Decis Sci 1999;30:197–216. https://doi.org/10.1111/j.1540-5915.1999.tb01606.x.
[47] Huynh, T. Q. & Setiono, R. Effective neural network pruning using cross-validation. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005* (IEEESunday, July 31, 2005), pp. 972–977.
[48] Srinivasan Kathiravan, Cherukuri Aswani Kumar, Vincent Durai Raj, Garg Ashish, Chen Bor-Yann. An efficient implementation of artificial neural networks with K-fold cross-validation for process optimization. J Int Technol 2019;20:1213–25.
[49] Setiono R. Feedforward neural network construction using cross validation. Neural Comput 2001;13:2865–77. https://doi.org/10.1162/089976601317098565.
[50] Pimentel MA, Clifton DA, Clifton L, Tarassenko L. A review of novelty detection. Signal Process 2014;99:215–49. https://doi.org/10.1016/j.sigpro.2013.12.026.
[51] Jones TR et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. Proc Natl Acad Sci USA 2009;106:1826–31. https://doi.org/10.1073/pnas.0808843106.
[52] Raemoe, P., Sacher, R., Snijder, B., Begemann, B. & Pelkmans, L. CellClassifier: supervised learning of cellular phenotypes, doi: 10.3929/ETHZ-B-000015925 (2009).
[53] Neumann B et al. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. Nature 2010;464:721–7. https://doi.org/10.1038/nature08869.
[54] Held M et al. Cell cognition: time-resolved phenotype annotation in high-throughput live cell imaging. Nat Methods 2010;7:747–54. https://doi.org/10.1038/nmeth.1486.
[55] Conrad C, Gerlich DW. Automated microscopy for high-content RNAi screening. J Cell Biol 2010;188:453–61. https://doi.org/10.1083/jcb.200910105.
[56] Liberali P, Snijder B, Pelkmans L. A hierarchical map of regulatory genetic interactions in membrane trafficking. Cell 2014;157:1473–87. https://doi.org/10.1016/j.cell.2014.04.029.
[57] Kümmel A et al. Differentiation and visualization of diverse cellular phenotypic responses in primary high-content screening. J Biomol Screen 2012;17:843–9. https://doi.org/10.1177/1087057112439324.
[58] Feng Y, Mitchison TJ, Bender A, Young DW, Tallarico JA. Multi-parameter phenotypic profiling: using cellular effects to characterize small-molecule compounds. Nat Rev. Drug Discov 2009;8:567–78. https://doi.org/10.1038/nrd2876.
[59] Bilbao, I. & Bilbao, J. Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks. In *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)* (IEEETuesday, December 5, 2017 - Thursday, December 7, 2017), pp. 173–177.
[60] Luecken MD et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods 2022;19:41–50. https://doi.org/10.1038/s41592-021-01336-8.