Highly accurate filters to flag frequent hitters in AlphaScreen assays by suggesting their mechanism

Dipan Ghosh,[a] Uwe Koch,[a] Kamyar Hadian,[b] Michael Sattler,[b] Igor V. Tetko[b],[c]*

Abstract: AlphaScreen is one of the most widely used assay technologies in drug discovery due to its versatility, dynamic range and sensitivity. However, a presence of false positives and frequent hitters contributes to difficulties with an interpretation of measured HTS data. Although filters do exist to identify frequent hitters for AlphaScreen, they are frequently based on privileged scaffolds. The development of such filters is time consuming and requires deep domain knowledge. Recently, machine learning and artificial intelligence methods are emerging as important tools to advance drug discovery and chemoinformatics, including their application to identification of frequent hitters in screening assays. However, the relative performance and complementarity of the Machine Learning and scaffold based techniques has not yet been comprehensively compared. In this study, we analyzed filters based on the privileged scaffolds with filters built using machine learning. Our results demonstrate that machine-learning methods provide more accurate filters for identification of frequent hitters in AlphaScreen assays than scaffold-based methods, and can be easily redeveloped once new data are measured. We present highly accurate models to identify frequent hitters in AlphaScreen assays.

Keywords: Alphascreen, Frequent hitters, False Positives, Machine Learning, High Throughput Assays

1 Introduction

Machine learning has found numerous applications in the field of drug discovery[3-5] and is being constantly expanded, both methodologically and regarding applications. One such area of application is to build filters using machine learning that can identify potential false leads, false positives and frequent hitters from High Throughput Screenings (HTS)[2, 6-8]. Machine learning methods are of particular interest in such cases, because firstly, the models have high accuracy and statistical characteristics which frequently outperform other methods[6]. Secondly, models can be built from very large datasets with relatively little efforts, as compared to other methods. Finally, for a machine learning approach, no crystallographic or other structural data is necessary, as opposed to some other methods such as molecular docking or structure-based pharmacophore models.

AlphaScreen is a very versatile assay technology which is commonly used in drug discovery projects[9]. The assay relies on the intended biological interaction to bring two beads together; a donor and an acceptor bead. If a donor-acceptor interaction is present[10], upon excitation of the donor bead, singlet oxygens are transferred to the acceptor beads, which then emits light that is detected. AlphaScreen is particularly suitable for HTS, due to the high signal-to-background ratio, dynamic range and sensitivity, together with the homogenous assay format and reagent stability. However, there are multiple ways ligand molecules can interfere with various components of AlphaScreen assay technology. Based on the mechanism of action, there are three general categories: These are Singlet oxygen quenchers, color quenchers or inner filters and light scatterers. Efforts have been made to identify such bad actors, and filters have been published to isolate them in a high throughput screening setting[1, 2, 11].

In the previous study[2], we reported two classes of interfering compounds, one that interfered with the interaction of the protein His-tag moiety to nickel chelate (Ni2+-NTA) beads of the AlphaScreen detection system, and another generic class of compounds that interfered with the assays via unknown mechanisms. In a follow-up study[1] we analyzed a class of compounds interfering with the interaction of glutathione S-transferase (GST) to glutathione (GSH), and thus interfering with AlphaScreen assays involving beads containing Glutathione. Scaffolds were identified that were over-represented among the identified frequent hitters. Such scaffolds were then encoded using SMARTS[12] strings, and the ToxAlerts[13] platform was used to build a working filter. In this study, we used machine learning methods to build models using the OCHEM platform[14] from the same data, in order to compare the efficacy of machine learning and scaffold-based approaches.

2 Data

We already had the *in house* data that were used to create the SMARTS based filter from the previous studies[1, 2]. We needed to have a robust test set against which we can compare machine learing against scaffold based methods.

[a] Lead Discovery Center GmbH, Otto-Hahn-Straße 15,44227 Dortmund, Germany

[b] Helmholtz Zentrum München - German Research Center for Environmental Health (GmbH), Institute of Structural Biology, Ingolstädter Landstraße 1, D-85764 Neuherberg, Germany

[c] BIGCHEM GmbH, Valerystr. 49, D-85716 Unterschleißheim, Germany

* Dr. Igor V. Tetko,

Institute of Structural Biology, Helmholtz Zentrum München - German Research Center for Environmental Health (GmbH), Institute of Structural Biology, Ingolstädter Landstraße 1, D-85764 Neuherberg, Germany

itetko@vcclab.org

For this, we used publicly available data in PubChem BioAssay[15]. We searched for AlphaScreen confirmatory high throughput screenings, and then selected 15 HTS campaigns with the highest number of actives (see Table S1). HTS campaigns with very small number of actives are statistically less useful for identifying frequent hitters.

From the previous studies, we had two *in house* datasets corresponding to the two types of frequent hitters (FH) types identified. However, the identified assays from PubChem BioAssay used various combinations of donor and acceptor beads (Table S1). In order to be directly comparable, we had to identify compounds that were interfering with either the Ni-NTA beads, or GSH coated beads. However, without counter-screen information, separating compounds based on mode of action (MOA) was not possible. Therefore, we merged the various types of FH identified in the previous study into one category in order to compare them against the generic FH from the PubChem BioAssay sets. Although we did not have an external test set for the individual classes, we built models for each of the different classes, in an effort to provide models that could suggest mechanism of action of FHs. Thus, in total, we had four different datasets (Table 1).

Table 1: Descriptions of the datasets used in this study. In the PubChem-Combined dataset, we varied selection threshold, resulting in a variable number of FH.

Data Set abbreviation	Data Source	Total Compounds	Active Compounds	FH	FH specificity
OCHEM-Ni-NTA	*in house*	24988		77	Streptavidin-Ni-NTA/His
OCHEM-GST	*in house*	24988		53	GST/GSH
OCHEM-Combined	*in house*	24988		190	-
PubChem-Combined	PubChem	489951	55560	Variable*	-

**The number of FHs was determined by the enrichment threshold used.*

For identifying FH, we used statistical analysis of compound activity. First, compounds that were tested less than five times were omitted. The activity fractions associated with low-test count was much higher for such compounds and thus such data points would introduce noise in the analysis. For example, if a compound was tested once and found to be active, then its activity fraction was 1. Such compounds should not be considered a FH since it could be just an active one. After filtering, we calculated activity fraction $F_{obs} = n_i/k_i$ for each compound, where a compound i was tested in n assays and was active k times. Next, for each assay we shuffled the activity labels, keeping the total number of active compounds unchanged. This means that per compound, the activity was randomized. After this, we calculated the activity fraction again. Let us call this F_{calc}. To obtain a statistically significant result, we repeated the randomization and calculated F_{calc} 10,000 times, and calculated the 95% upper confidence interval $Fav_{(95\%)}$. Then for each activity fraction, we defined an enrichment factor E_n as the ratio of F_{obs} to $Fav_{(95\%)}$. This enrichment factor was the degree of overrepresentation for that activity fraction when compared to the one obtained randomly. For example, if for activity fraction 0.7 we calculated E_n = 10, then compounds that had an activity fraction of 0.7 were observed in the active group 10 times more, than it would be expected by chance. Considering that assays were not related, such overrepresentation was related to the non-specific activity of compounds, i.e., these compounds were frequent hitters (FH). Therefore, this metric served as an indicator of FH propensity (Table S2). Of course, only compounds with enrichment factor > 1 could be considered as FH.

For identifying FH, we selected a threshold for the enrichment. The larger the threshold the higher was probability of selected compounds to be FH. If the models and methods of prediction were accurate, the use of the larger threshold could increase their prediction score, i.e., the models should be able to predict compounds that are more likely to be frequent hitters with higher accuracy. This provided an additional measure for comparison of accuracy of the machine learning and the scaffold-based methods. Because of this reason the FH count for the PubChem-Combined set was mentioned to be variable in Table 1.

3 Methods

3.1 Machine learning methods

Using the freely accessible platform On-line Chemical and Modeling Environment (OCHEM), we performed comprehensive modeling for all three datasets. We applied different machine learning algorithms available in OCHEM, such as Associative Neural Network (ASNN)[16], Deep Neural Network (DNN)[17], Least Square Support Vector Machine (LSSVM)[18] for training the models. We also applied the newly proposed Transformer-CNN[19] method that uses the SMILES representation of molecules. The models were developed with default parameters of the methods as specified on the OCHEM web site and described at http://docs.ochem.eu.

We have used stratified cross-validation (CV) and stratified Bagging for developing our models. Both of these methods are used for internal validation or scoring of the model during the training phase. In cross-validation, the training data is randomly subdivided into n bins. One of the bins is used as a test set, and others are combined to form the training set. OCHEM by default uses 5-fold cross-validation, meaning the data is subdivided into 5 bins. Bootstrap aggregating[20] (bagging) is a variation of machine-learning ensemble meta-algorithm that relies on building multiple classification or regression models and averaging the results (for regression tasks) or voting on the result (for classification tasks) to obtain the final prediction. Due to this sample and ensemble approach, there is a good chance that each molecule in the training data will appear in one of the validation sets, thus providing better performance metric for the model. However, this also means that multiple models will have to be created and averaged, which increases the computational cost and size of such models significantly.

The term stratified, which can be applied both to CV and Bagging, indicates that the bins or the bootstrap samples were created to over-represent the smallest class to have the equal number of samples of each class in the respective training sets. Therefore, this can only be done in classification tasks.

3.2 Molecular descriptors

A variety of descriptors available within the OCHEM environment were used to develop the models.

ALogPS calculates two descriptors provided by the ALOGPS[21] program, which determine the water/octanol partition coefficient (logP_{calc}), and water solubility coefficient (logS_{calc})[22].

alvaDesc[23] is a relatively recent software package that calculates over 5000 descriptors that are divided in 33 logical blocks. Additionally, it also calculates different molecular fingerprints.

CDDD, which stands for Continuous and Data-Driven Descriptors[24] are descriptors derived from a molecular representation using a pre-trained deep learning model. The model that generates the descriptor uses the ability of deep neural networks to learn a feature representation from low-level encodings of molecules as SMILES.

CDK (3D) or the Chemistry Development Kit is an open source chemoinformatics project[25]. There are several types of descriptors available from the package that are integrated into the OCHEM environment. Descriptors calculated with the recently released 2.3 version of CDK were used in this study[26].

ChemAxon Descriptors (3D) are a set of descriptors provided by the ChemAxon company[27]. The available descriptors are subdivided into seven categories, namely Elemental Analysis, Charge, Geometry, Partitioning, Protonation, Isomers, and Others. Descriptors that return a Boolean or Numerical value were implemented into OCHEM.

Dragon[28] (3D) is a well-known software package, for the calculation of molecular descriptors, developed by the Milano Chemometrics and QSAR Research Group of Prof. R. Todeschini. It comprises perhaps one of the largest and most comprehensive molecular descriptor libraries available, with a total of 5,270 descriptors available. The descriptors are divided into 30 discrete blocks, such as Topological, Constitutional, Drug-like indices, etc. Dragon version 6 was used.

GSFRAG[29] belongs to the category of 2D fragment descriptors. It calculates the occurrence numbers of certain special fragments from k = 2 to 10 vertices in a molecular graph G that can be used as molecular descriptors in quantitative structure-property/activity studies.

MAP4[30] or MinHashed Atom Pair fingerprint of radius 2, is a fingerprint based on the topological distance between all atom pairs in a circular substructure within a given radius of the molecule. The original implementation sets the radius to be 2, but it can be customized for different purposes. If the radius is 2, the fingerprint is known as MAP4, which is used in his study.

MORDRED[31] is a python based open source descriptor package that can calculate more than 1800 two- and three-dimensional descriptors. The Mordred package is easy to install in any environment, and can be deployed as a webserver. It also has pre-processing built-in to ensure correctness of the descriptors.

RDKit[32] is a very popular python library for Cheminformatics. It offers ~40 2D descriptors, ~15 3D descriptors, and 8 fingerprints. These descriptors can be used from within OCHEM, and 2D descriptors can be used independently of 3D descriptors, which allows for comparison between the two.

PyDescriptor[33] is designed to be a PyMol plugin that calculates descriptors. The package calculates >16k descriptors, ranging from simple properties such as molecular weight to complex topological fingerprints that are based on the 3D structure of the molecule.

ISIDA descriptors are part of the ISIDA project, which stands for *In-SIlico* Design and data Analysis[34]. These fragment-like 2D descriptors are calculated from molecular graphs using three different methods, namely paths, trees, and neighbours. The descriptors are generated from the fragments by using different atom and bond labeling methods.

Mera and Mersy[35] (3D) are two related groups of descriptors. Mera provides a group of descriptors that deal with molecular area and surface. Mersy is abbreviated as Mera Symmetry, and the descriptors are calculated using 3D representations of molecules in the framework of the MERA algorithm.

Spectrophores[36] are 1D descriptors that encode the property fields surrounding the molecules. This provides a chemical-class-independent descriptor that can be used to build models.

QNPR (Quantitative Name Property Relationship) are 1D descriptors that are directly based on the SMILES representation of the molecules. The descriptors are calculated by splitting the respective string into all possible continuous substrings with a specified maximum length[37].

ToxAlert's[13] Extended Functional Group (EFG)[38] category are descriptors based on classification initially provided by the CheckMol software[39]. The coverage was extended to include new groups, particularly heterocycles[38]. ToxAlert covers total of 583 functional groups.

The assumption was that by using a different representation of chemical structures we could develop models covering different molecular aspects responsible for the FH activity of molecules.

3.3 Statistical coefficients

For internal validation of the generated models, we used 5-fold stratified cross validation[40]. Accuracy was defined as the percentage of correctly classified samples, given by the formula

$$ACC = (TP + TN) / (TP + FP + TN + FN) \qquad (1)$$

where TP, TN, FP and FN were the number of True Positive, True Negative, False Positive and False Negative, respectively. Due to the large imbalance of the active and the inactive populations, high ACC, e.g., ACC=0.993 for OCHEM-Combined, could be calculated for models without any prediction power, which predicted all compounds as non FHs. Therefore, instead of ACC we used Balanced Accuracy (BA) for determining the quality of the models. It was defined as:

$$BA = 0.5*(TP/P + TN/N) \qquad (2)$$

where P=TP + FN and N = TN+FP were the numbers of positive and negative samples, respectively.

Since we analyzed extremely unbalanced datasets, it was important to optimize the thresholds that were used to classify the predictions from the machine learning models. Usually such threshold is set to be 0.5, but such value may not be optimal in particular for models built with cross-validation, since such models are biased to provide the highest ACC. A selection of an optimal value of this threshold could increase BA but did not influence the ROC AUC (Receiver Operating characteristic Curve – Area Under the Curve) values, which is another traditional statistical coefficient to measure performance of classification methods. For stratified bagging due to bootstrapping, the data were balanced and threshold of 0.5 was frequently the optional one. OCHEM had a support for identifying the optimal threshold to maximize Balanced Accuracy: the threshold is changed until maximum BA is calculated for the training set. This feature was used to identify the optimal threshold to calculate the BA reported.

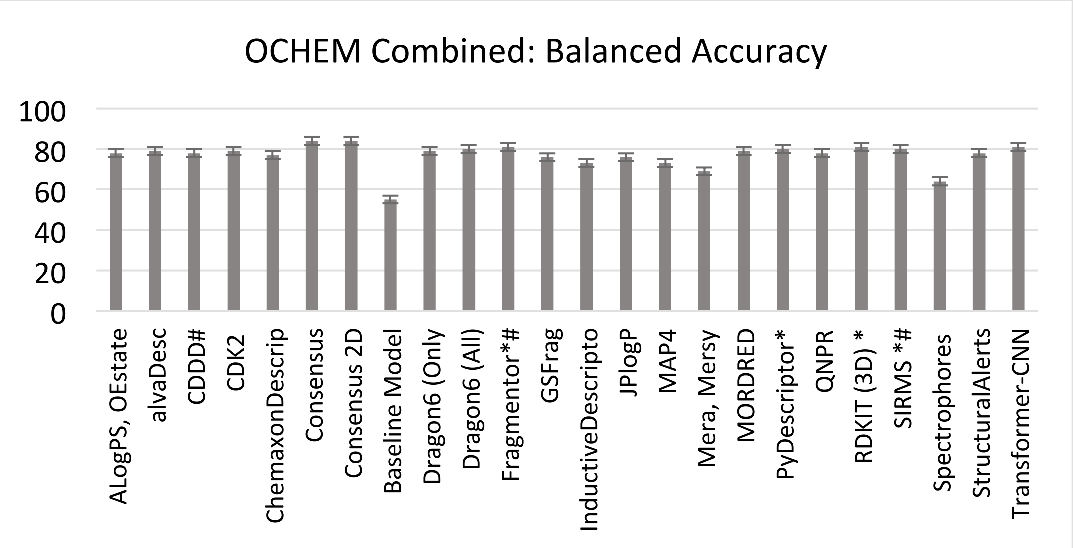

Figure 1: Comprehensive modelling with the OCHEM-Combined dataset. The models presented in the chart, with an exception of Transformer-CNN, were created with the Associative Neural Network. The Balanced Accuracy scores calculated with stratified bagging are plotted. Consensus models were calculated using simple average. Data corresponding to the chart is available in Table S3 of the supplementary information. *Models denoted were considered for the general consensus model. #Models were used in calculating the 2D only consensus model.

We also used the ROC-AUC metric from our models. ROC-AUC[41] is the area under the receiver operating characteristic curve. A receiver operating characteristic curve or ROC is a graphical plot that describes the variance in the discrimination power of a binary classification model. The curve is created by plotting the models' sensitivity against its false positive rate at various threshold levels Therefore, the ROC curve describes the sensitivity as a function of false positive rate. The area under this curve provides a measure for model accuracy. ROC-AUC can be used in unbalanced datasets. It ranges from 0 to 1, with 1 being perfect prediction.

4 Results

4.1 Frequent Hitter Analysis

In order to compare performances of previously developed ToxAlert filters[1, 2] and machine-learning model, we used independent test sets that were different from those utilized to develop the methods. As described in the Data section we selected these sets from the PubChem BioAssay[2, 15] and identified FHs based on the statistical analysis. If we considered compounds with an enrichment value > 10 to be a frequent hitter, then we identified 7633 FHs out of ~350K compounds, which is ~2.1%. In comparison, in the *in house* training set there were 190 FH compounds out of 24988, which is ~0.7%. The lower number of FHs in in the *in house* library could be due to explicit excluding of potential FHs compounds by using filters on reactive and unstable compounds available in the ToxAlert platform when selecting compounds for the library. While we did not use the PAINS filters for the library design, such selection still reduced the percentage of FH.

4.2 Machine-learning Models:

For determining comparative performance of machine-learning models against scaffold based FHs, we performed comprehensive modeling with the combined *in-house* data using a variety of descriptors in OCHEM (Figure 1).

We used Least Square SVM, ASNN, and deep learning algorithms available in OCHEM. Out of all methods, ASNN showed best overall performance, the results are summarized in Figure 1. The five best performing models were chosen for building the consensus model. For this, outputs reported by each individual model is averaged and the average score is considered as the output of the consensus model. The consensus model thus developed were applied to the test set.

We used a variety of descriptor packages available in OCHEM. Some of the descriptors required the 3D structure of the molecule, for which the Corina software package[42] was used as a part of the modelling pipeline in OCHEM. We decided to make one consensus model from the best performing models with descriptors that did not require 3D structures, and another consensus model with the best performing models regardless of the descriptor package used. The consensus model using 2D+3D descriptors outperformed the 2D only consensus model, but only by a small margin with an AUC score of 0.9 for the 2D only model versus 0.91 for the general consensus model. A similar trend was also observed while modeling with other datasets in the study (Figure 3 and 4). This indicates that models derived from only the 2D structure of molecules can be used with almost equal effectiveness to models based on 3D structures. As 3D structure calculation is computationally expensive, models derived from only the 2D structure of molecules may be used as a faster and lighter solution.

We also trained a Transformer-CNN model with this data. This method requires no descriptors to be calculated, and therefore, is agnostic of any bias that may arise from using one descriptor over the other. The model is also very lightweight. It required only 8MB of disk space, whereas some of the other models that depend on large descriptor sets such as Dragon needed 300MB.

4.3 Comparison between Scaffold based Filter and Machine-learning Models:

To compare the performance of the developed machine learning filter, we applied the general consensus model developed from the combined training set to the test set collected from PubChem BioAssay data. As discussed in the data section, for classifying FH, we used a threshold value for the enrichment that we calculated. As we increased this threshold, we compared performances for machine learning consensus model, the PAINS filter developed by Baell *et al*[11], as well as the combined filter created from previous AlphaScreen studies[1, 2] using *in house* data.

With a very low enrichment value, the entire library was marked as frequent hitters, and therefore, it resulted in very poor balanced accuracy from all the methods. As the threshold value increased, there was a marked increase in the BA for both PAINS and the machine learning methods, with PAINS outperforming machine learning. However, the balanced accuracy for the AlphaScreen FH based on *in house* data barely increased (Figure 2). At an enrichment threshold of >25, the consensus machine learning model calculated a BA of 64%, and the PAINS filter calculated a BA of 65%. It should also be noted that increasing the threshold also reduced the number of identified FH significantly. At an enrichment threshold of 25, the number of FHs was down to 4k from 7.6k compounds calculated for threshold of 10.

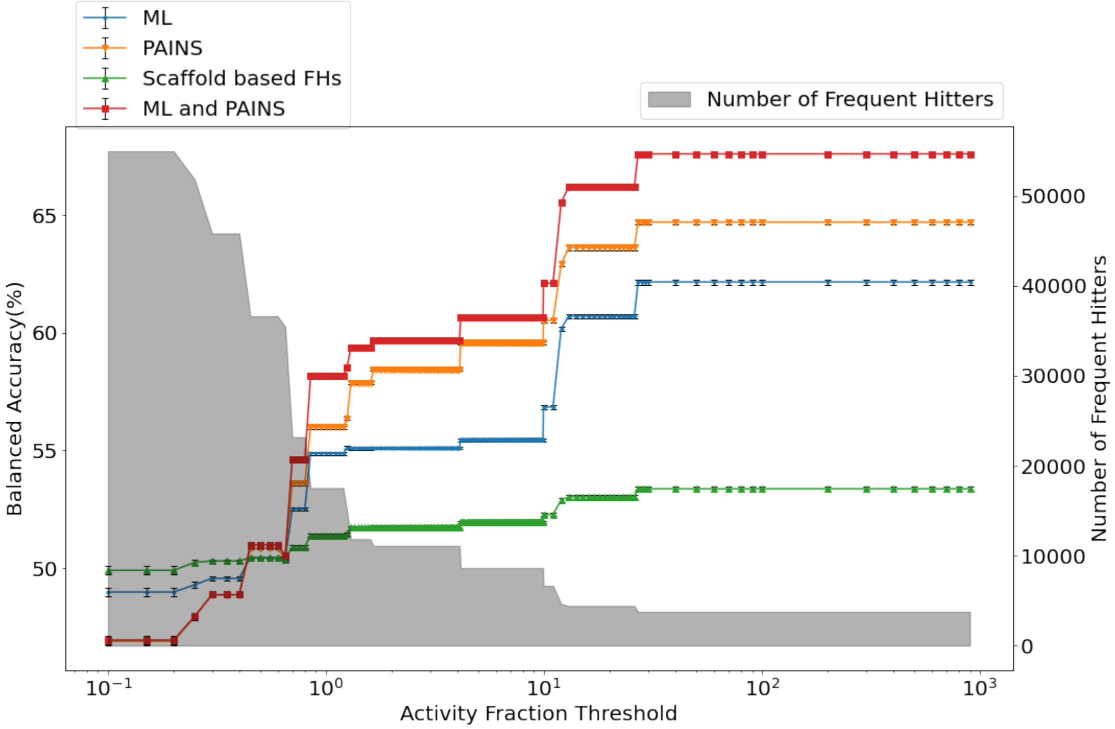

Figure 2: Comparative performance of filters based on the chemical scaffolds groups (Scaffold-based FHs from previous studies[1, 2]) and machine learning model (ML) developed from the same data across a range of selectivity thresholds. The PAINS filter and a combined filter (ML+PAINS) is also shown. Number of compounds identified as FHs in PubChem are shown as char bar diagram for the different enrichment thresholds.

The reason for the poor performance of the scaffold-based FHs was the limited nature of the *in house* dataset. A scaffold-based method could be very effective, but the scaffolds identified must cover a wide chemical space. This was evident from the fact that PAINS, which is also a scaffold-based filter, had significantly better performance. The PAINS scaffolds were identified from six AlphaScreen based assays with a total compound pool of 93212. They covered 2062 FH (2.2%) that were active in four or more out of the six assays. Compared to that, as explained below the *in house* data had reduced fraction (0.9%) of FHs due to the library design. Therefore, the scaffold-based filter derived from the *in-house* data performed poorly compared to the PAINS filters. However, interestingly, the machine-learning model developed from such a limited dataset was still able to perform almost comparable to the PAINS filter, demonstrating the effectiveness of this approach.

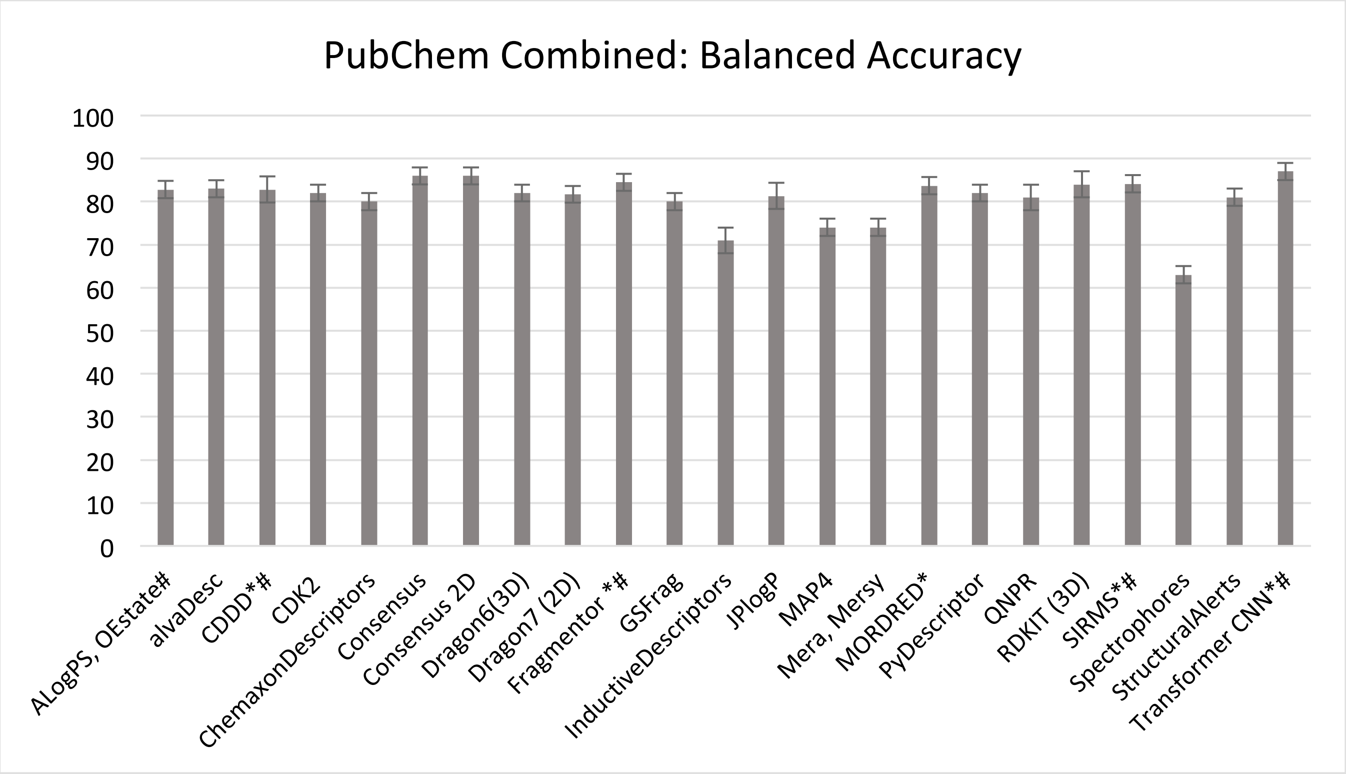

Figure 3: Comprehensive modelling using PubChem-Combined dataset. The models presented in the chart, with an exception of Transformer-CNN, were created with the Associative Neural Network. The ROC-AUC scores calculated using 5-fold cross-validation are plotted. Consensus models were calculated using simple average. Data corresponding to the chart is available in Table S4 of the supplementary information. *Models denoted were considered for the general consensus model. #Models were used in calculating the 2D only consensus model.

As the machine learning and the PAINS filter both performed well and were derived from independent datasets, we combined both of them and considered compound as a FH when it was predicted by any of the approaches. The combined filter improved balanced accuracy by ~5 % over the entire range (ML+ PAINS as shown in Figure 2) and showed 68.3% balanced accuracy at an enrichment threshold of >25. Using bootstrapping, we determined that the improvement was statistically significant ($p<0.05$).

As the performance of the machine-learning model is always limited by the training dataset, we wanted to explore whether better results can be calculated by developing a model with a larger dataset. Therefore we developed models using PubChem-Combined and applied it to the OCHEM-Combined. The rational was that as the PubChem-Combined set was much larger. Therefore a model trained with such a set should be able to pick up FHs from the comparatively limited OCHEM-Combined set.

We performed comprehensive modelling using OCHEM (Figure 3), and decided to use 5-fold cross-validation instead of bagging, since performing bootstrap aggregation on such a large dataset contributed very large models. For this dataset, the Transformer-CNN method contributed the model with the highest AUC-ROC=0.94 and BA=87%. We built a consensus model with the Transformer-CNN model and four other best performing model that were built using ASNN. Applied to OCHEM-Combined, the consensus model identified 139 out of 190 frequent hitters, resulting in a sensitivity score of 73% and BA of 79% Compared to this, the PAINS filter applied to OCHEM-Combined only identified 65 out of 190 frequent hitters, with a sensitivity score of only 34% and BA of 65%. Since both PAINS and consensus model were developed with different sets, we cannot compare their performances directly. Still, this clearly demonstrates that machine learning models, when trained with appropriate dataset, could provide very good accuracy.

4.4 Machine learning models to identify mechanism of action of FHs

As machine learning models was determined to be effective, we also developed models that could suggest mechanism of action (MoA) for FHs (Figure 3). For example, the frequent hitters in the OCHEM-Ni-NTA and OCHEM-GST datasets (Table 1) interfered with Histidine binding to Ni-NTA and interaction of glutathione S-transferase (GST) to glutathione (GSH), respectively. Since the machine learning method provided better performance compared to scaffold-based filters for the combined set, the models developed for each of the sets could be more efficient to identify MoA of new FHs. Therefore, we joined all datasets and developed a multitask model that simultaneously predicts whether a compound is a frequent hitter, and its MoA. The training set had 371604 molecules spanning three different target properties, i.e., one for if the molecule is a frequent hitter, and two for the two modes of action. We used both a multitask learning approach[43], and multiple single task learning approaches using stratified cross validation. For this study the combined single task learning performed better. Among the three properties predicted, PubChem-Combined showed best performance, followed by OCHEM-Ni-NTA, and OCHEM-GST showed worse performance consistently. This is due to the amount and nature of data involved. PubChem-Combined is a much larger dataset than OCHEM-GST or OCHEM-Ni-NTA, and this produces a better model. OCHEM-GST has fewer compounds marked as active (53 compared to 77). All other compounds between the two sets are the same, so the lesser number of actives results in worse performance. We have built other models from the PubChem-Combined set in this study (Figure 3). The performances of the two sets of models were very similar (Consensus 0.93 vs 0.94), which is expected because of the multiple single task learning approach used. The transformer neural network showed the best performance amid individual models, with average AUC of 0.87 and BA of 84.7% (Figure 4). A consensus model with only 2D descriptors improved the AUC and BA score to 0.90. Including all types of descriptors in the consensus produced the final model, with an AUC of 0.91 and BA of 84.7%. The model is publicly available on the OCHEM web site at https://ochem.eu/article/125278

We were curious to find out what improvements did the model actually achieve, apart from the statistical scores. So, we applied this final model to OCHEM-Combined, and it was able to identify 140 out of 190 frequent hitters. This is very similar to the performance shown by models built from the PubChem-Combined dataset (Figure 3), and this is expected. As explained previously, the PAINS filter identified only 65 of these 190 compounds, so we filtered out the 75 compounds that our model identified, but PAINS did not. The full list of these compounds are presented in the supplementary information (Table S6), and a few scaffolds of interest are presented in Figure 5. In general, the collection is quite heterogeneous, and no prominent scaffold could be identified. This is expected, because a well-known scaffold-based filter such as PAINS, were not able to identify these compounds. There are a few toxoflavines, as well as toxoflavin mimics. We also noted the presence of multiple condensed polyaromatic moieties, cyanodithiines, aminothazoles and picolyamines (Figure 5). It should be noted here that many of these scaffolds

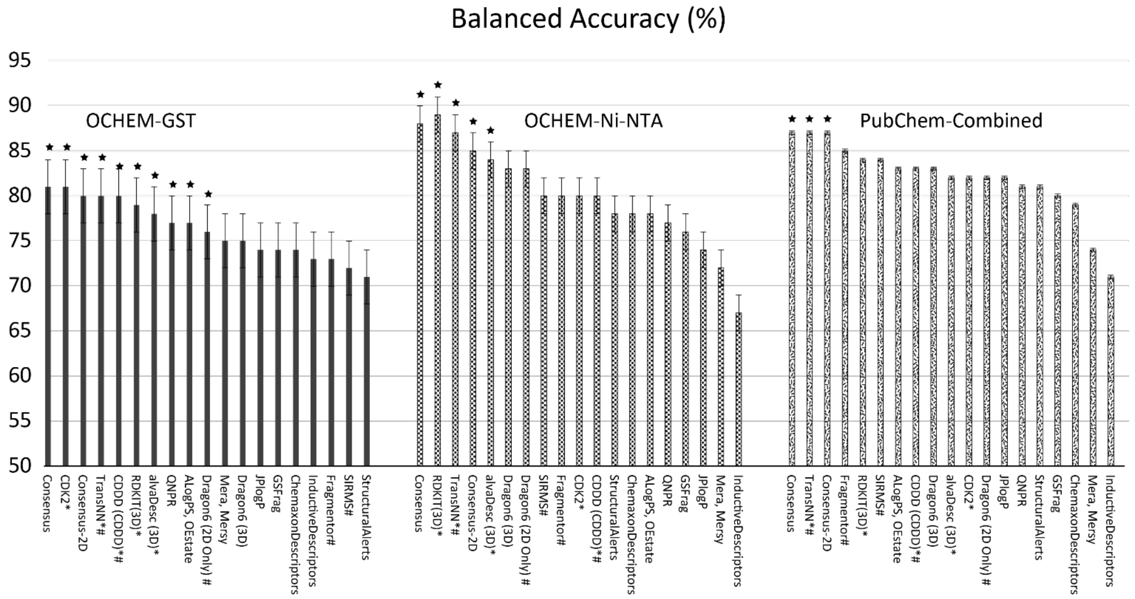

Figure 4: Multiple Single Task learning for predicting frequent hitter and possible Modes of Action. Different datasets were used to create the three submodels, as appropriate. For example, OCHEM-GST dataset was used to create the model for predicting compounds interfering with GST. The models presented in the chart, with an exception of Transformer-CNN, were created with Associative Neural Network. The Balanced Accuracy calculated using 5-fold stratified cross-validation are plotted. Consensus models were calculated using simple average. Data corresponding to the chart is available in Table S5 of the supplementary information. The model with best performance that are statistically significantly different from the rest is marked with a large star (★). *Models denoted were considered for the general consensus model. #Models were used in calculating the 2D only consensus model.

has been identified manually before, and is already present in the scaffold-based filter. However, as we have discussed in detail, that scaffold-based filter performs very poorly against a broad test set. It is able to pick up these particular scaffolds, but suffers greatly in overall performance. On the other hand, the machine learning model was able to identify these scaffolds without being exposed to them before, and provides much better overall performance as well (79% BA compared to 65% of PAINS, when the final model is applied to OCHEM-Combined). Therefore, in the final model, we are able to detect the presence of a frequent hitter with better accuracy, we are able to detect scaffolds and compounds that a traditional method such as PAINS overlooks, and we are also able to comment on a possible mode of action. This makes our model an excellent tool for filtering frequent hitter compounds in AlphaScreen assays.

5 Conclusion:

In this study, we compared scaffold-based methods and ML methods for identifying AlphaScreen frequent hitters. The result showed that ML models outperformed scaffold-based methods by a large margin, when both approaches originated from the same data. We demonstrated that a ML model trained on a large dataset collected from PubChem BioAssay outperformed PAINS[11] filter for prediction of the public and *in house* data. However, combining PAINS with a ML improved the prediction outcome. We also developed ML models from the individual *in-house* datasets, for which the mechanism of action (MoA) was known. These models should be effective in identifying compounds that interferes through particular MoAs and should have higher accuracy compared to the existing scaffold-based filters developed in the previous studies[1, 2]. Finally, we also contributed a ML model based on the PubChem dataset. As this set was much larger and more diverse than the training set used to develop PAINS, it provided better coverage of the chemical space, and was able to identify frequent hitters in the *in-house* AlphaScreen assays with higher accuracy than the PAINS filter. We also demonstrated that the new ML model is able to identify scaffolds that are not identified by PAINS, without introducing false positives, and thus providing better balanced accuracy scores.

In summary, the scaffold-based methods were limited to the identified scaffolds only, and therefore had lower accuracy compared to ML when both types of models were used to screen large heterogeneous datasets. Identifying scaffolds from large data sets, such as the PubChem-Combined and after that manually inspecting them to develop representative SMARTS patterns, as we did in our previous studies[1, 2], would be a complicated task that would take a very long time. ML on the other hand benefited from very large datasets. Re-training the models with new datasets is trivial, and adding new molecules could further improve the accuracy of the model. If a scaffold-based filter is too specific, it is ineffective in large sets, if it is too general and has many scaffolds, then it may produce too many false positives. Machine learning can identify much more sophisticated and complex pattern in the data, and therefore provide better prediction accuracy.

The models developed in this study are freely available on the OCHEM platform (https://ochem.eu/article/125278).

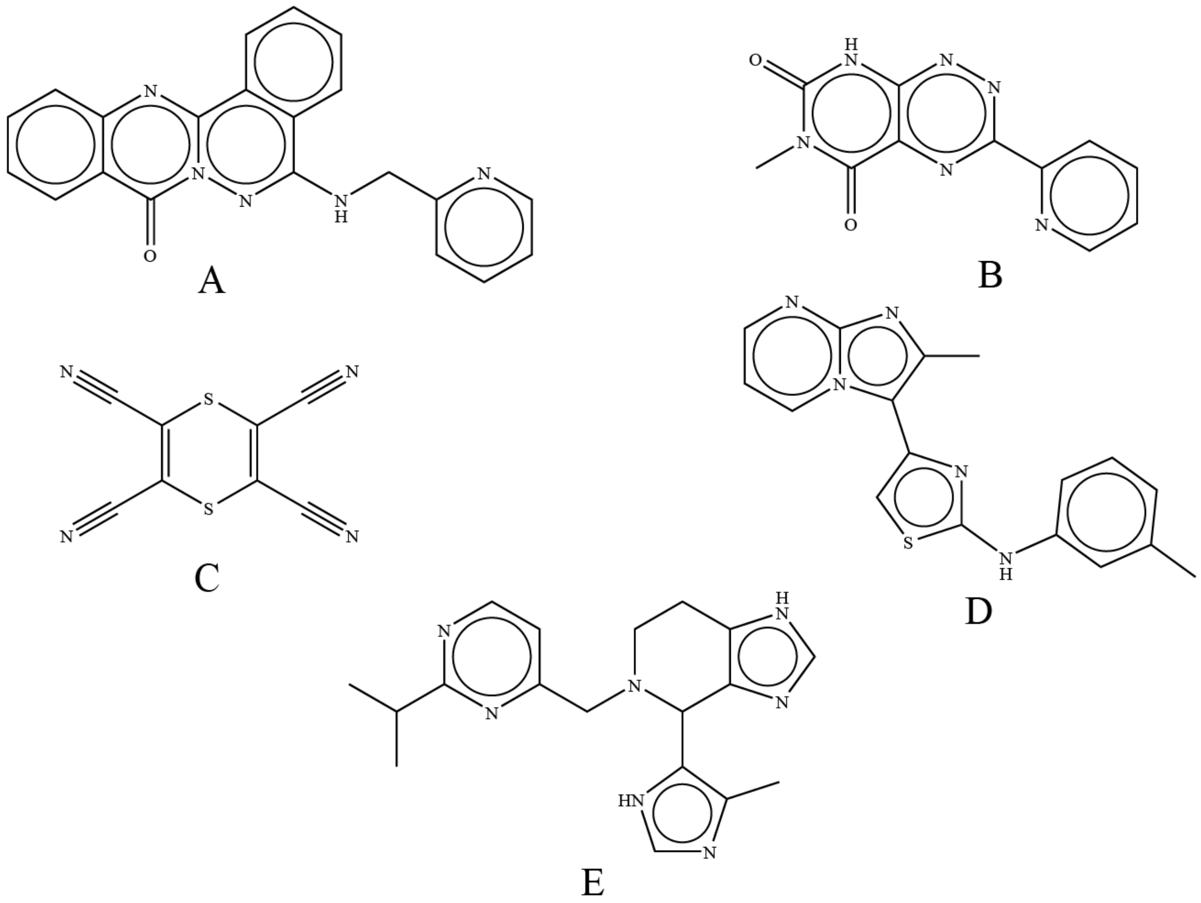

Figure 5: Examples of confirmed AlphaScreen frequent hitter compounds identified by the ML filter but not by PAINS. **A.** Scaffold resembling toxoflavin **B.** Toxoflavin **C.** cyanodithiine **D.** aminothazole and **E.** picolylamine.

6 Acknowledgments

This study was partially funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Innovative Training Network European Industrial Doctorate grant agreement No. 676434, "Big Data in Chemistry". The authors thank Alvascience Srl, ChemAxon and Molecular Networks GmbH for a possibility to use descriptors and Corina programs in their study.

7 References

[1] Brenke, J. K.; Salmina, E. S.; Ringelstetter, L.; Dornauer, S.; Kuzikov, M.; Rothenaigner, I.; Schorpp, K.; Giehler, F.; Gopalakrishnan, J.; Kieser, A.; Gul, S.; Tetko, I. V.; Hadian, K., *J. Biomol. Screen.* **2016,** *21*, 596-607.

[2] Schorpp, K.; Rothenaigner, I.; Salmina, E.; Reinshagen, J.; Low, T.; Brenke, J. K.; Gopalakrishnan, J.; Tetko, I. V.; Gul, S.; Hadian, K., *J. Biomol. Screen.* **2014,** *19*, 715-26.

[3] Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T., *Drug Discovery Today* **2018**.

[4] Lavecchia, A., *Drug Discovery Today* **2015,** *20*, 318-331.

[5] Baskin, I. I.; Winkler, D.; Tetko, I. V., *Expert Opin. Drug Discov.* **2016,** *11*, 785-795.

[6] Ghosh, D.; Koch, U.; Hadian, K.; Sattler, M.; Tetko, I. V., *J. Chem. Inf. Model.* **2018,** *58*, 933-942.

[7] Stork, C.; Wagner, J.; Friedrich, N. O.; de Bruyn Kops, C.; Šícho, M.; Kirchmair, J., *ChemMedChem* **2017,** *13*, 564-571.

[8] David, L.; Walsh, J.; Sturm, N.; Feierberg, I.; Nissink, J. W. M.; Chen, H.; Bajorath, J.; Engkvist, O., *ChemMedChem* **2019,** *14*, 1795-1802.

[9] Yasgar, A.; Jadhav, A.; Simeonov, A.; Coussens, N. P., *Methods Mol. Biol.* **2016,** *1439*, 77-98.

[10] Eglen, R. M.; Reisine, T.; Roby, P.; Rouleau, N.; Illy, C.; Bossé, R.; Bielefeld, M., *Current chemical genomics* **2008,** *1*, 2-10.

[11] Baell, J. B.; Holloway, G. A., *J. Med. Chem.* **2010,** *53*, 2719-40.

[12] Mayfield, J. W.; Sayle, R. A., *J. Cheminformatics* **2017,** *9*, 10.

[13] Sushko, I.; Salmina, E.; Potemkin, V. A.; Poda, G.; Tetko, I. V., *J. Chem. Inf. Model.* **2012,** *52*, 2310-2316.

[14] Sushko, I.; Novotarskyi, S.; Korner, R.; Pandey, A. K.; Rupp, M.; Teetz, W.; Brandmaier, S.; Abdelaziz, A.; Prokopenko, V. V.; Tanchuk, V. Y.; Todeschini, R.; Varnek, A.; Marcou, G.; Ertl, P.; Potemkin, V.; Grishina, M.; Gasteiger, J.; Schwab, C.; Baskin, I. I.; Palyulin, V. A.; Radchenko, E. V.; Welsh, W. J.; Kholodovych, V.; Chekmarev, D.; Cherkasov, A.; Aires-de-Sousa, J.; Zhang, Q. Y.; Bender, A.; Nigsch, F.; Patiny, L.; Williams, A.; Tkachenko, V.; Tetko, I. V., *J. Comput. Aided Mol. Des.* **2011,** *25*, 533-54.

[15] Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.;

Zaslavsky, L.; Zhang, J.; Bolton, E. E., *Nucleic Acids Res.* **2019,** *47*, D1102-D1109.

[16] Tetko, I. V., *Neural Process. Lett.* **2002,** *16*, 187-199.

[17] Sosnin, S.; Karlov, D.; Tetko, I. V.; Fedorov, M. V., *J. Chem. Inf. Model.* **2019,** *59*, 1062-1072.

[18] Suykens, J. A. K.; Vandewalle, J., *Neural Processing Letters* **1999,** *9*, 293-300.

[19] Karpov, P.; Godin, G.; Tetko, I. V., *J. Cheminformatics* **2020,** *12*, 17.

[20] Breiman, L., *Machine Learning* **1996,** *24*, 123-140.

[21] Tetko, I. V.; Tanchuk, V. Y., *J. Chem. Inf. Comput. Sci.* **2002,** *42*, 1136-1145.

[22] Tetko, I. V.; Tanchuk, V. Y.; Kasheva, T. N.; Villa, A. E. P., *J. Chem. Inf. Comput. Sci.* **2001,** *41*, 1488-1493.

[23] Mauri, A., alvaDesc: A Tool to Calculate and Analyze Molecular Descriptors and Fingerprints. In *Ecotoxicological QSARs*, Roy, K., Ed. Springer US: New York, NY, 2020; pp 801-820.

[24] Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A., *Chem. Sci.* **2019**.

[25] Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E., *J. Chem. Inf. Comput. Sci.* **2003,** *43*, 493-500.

[26] Willighagen, E. L.; Mayfield, J. W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliazkova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O.; Torrance, G.; Evelo, C. T.; Guha, R.; Steinbeck, C., *J. Cheminformatics* **2017,** *9*, 33.

[27] Myrdal, P. B.; Manka, A. M.; Yalkowsky, S. H., *Chemosphere* **1995,** *30*, 1619-1637.

[28] Todeschini, R.; Consonni, V., In *Handbook of Molecular Descriptors*, Wiley-VCH Verlag GmbH: 2000.

[29] Skvortsova, M. I.; Baskin, I. I.; Skvortsov, L. A.; Palyulin, V. A.; Zefirov, N. S.; Stankevich, I. V., *J. Mol. Struct.: THEOCHEM* **1999,** *466*, 211-217.

[30] Capecchi, A.; Probst, D.; Reymond, J.-L., *J. Cheminformatics* **2020,** *12*, 43.

[31] Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T., *J. Cheminformatics* **2018,** *10*, 4.

[32] Landrum, G. A. RDKit, Open-Source Cheminformatics. http://www.rdkit.org.

[33] Masand, V. H.; Rastija, V., *Chemometrics Intellig. Lab. Syst.* **2017,** *169*, 12-18.

[34] Varnek, A.; Fourches, D.; Horvath, D.; Klimchuk, O.; Gaudin, C.; Vayer, P.; Solov'ev, V.; Hoonakker, F.; Tetko, I. V.; Marcou, G., *Curr Comput Aided Drug Des* **2008,** *4*, 191-198.

[35] Potemkin, V. A.; Grishina, M. A., *J. Comput. Aided Mol. Des.* **2008,** *22*, 489-505.

[36] Gladysz, R.; Dos Santos, F. M.; Langenaeker, W.; Thijs, G.; Augustyns, K.; De Winter, H., *J. Cheminformatics* **2018,** *10*, 9.

[37] Tetko, I. V.; M. Lowe, D.; Williams, A. J., *J. Cheminformatics* **2016,** *8*, 2.

[38] Salmina, E. S.; Haider, N.; Tetko, I. V., *Molecules* **2015,** *21*, E1.

[39] Haider, N., *Molecules* **2010,** *15*.

[40] Tetko, I. V.; Sushko, I.; Pandey, A. K.; Zhu, H.; Tropsha, A.; Papa, E.; Oberg, T.; Todeschini, R.; Fourches, D.; Varnek, A., *J. Chem. Inf. Model.* **2008,** *48*, 1733-46.

[41] Hajian-Tilaki, K., *Caspian. J. Intern. Med.* **2013,** *4*, 627-635.

[42] Sadowski, J.; Gasteiger, J.; Klebe, G., *J. Chem. Inf. Comput. Sci.* **1994,** *34*, 1000-1008.

[43] Sosnin, S.; Vashurina, M.; Withnall, M.; Karpov, P.; Fedorov, M.; Tetko, I. V., *Mol. Inform.* **2019,** *38*, e1800108.