Active Learning of Compounds Activity – Towards Scientifically Sound Simulation of Drug Candidates Identification

Wojciech Marian Czarnecki[1], Stanisaw Jastrzebski[1], Igor Sieradzki[1], and Sabina Podlewska[2]

[1] Faculty of Mathematics and Computer Science,
Jagiellonian University, Krakow, Poland

[2] Institute of Pharmacology,
Polish Academy of Sciences, Krakow, Poland

wojciech.czarnecki@uj.edu.pl, stanislaw.jastrzebski@uj.edu.pl,
igor.sieradzki@uj.edu.pl, smusz@if-pan.krakow.pl

Abstract. Virtual screening is one of the vital elements of modern drug design process. It is aimed at identification of potential drug candidates out of large datasets of chemical compounds. Many machine learning (ML) methods have been proposed to improve the efficiency and accuracy of this procedure with Support Vector Machines belonging to the group of the most popular ones. Most commonly, performance in this task is evaluated in an offline manner, where model is tested after training on randomly chosen subset of data. This is in stark contrast to the practice of drug candidate selection, where researcher iteratively chooses batches of next compounds to test. This paper proposes to frame this problem as an active learning process, where we search for new drug candidates through exploration of the compounds space simultaneously with the exploitation of current knowledge. We introduce the proof of concept of the simulation and evaluation of such pipeline, together with novel solutions based on mixing clustering and greedy k-batch active learning strategy.

Keywords: active learning, tanimoto coefficient, compounds activity prediction, cheminformatics, clustering, virtual screening

1 Introduction

Cheminformatics is a rapidly growing field at the intersection of computer science and chemistry. Due to the rapid growth of the amount of experimental data, the need for efficient, statistical methods for their deep and systematic analysis emerged. Classification models, such as Support Vector Machines are widely adapted [21, 25] to many problems in the field, in particular to the tasks connected with the prediction of biological activity of chemical compounds, on

which we focus in our research. The main contribution of this paper is proposing realistic scenario for evaluating machine learning method performance in the above problem.

Active learning is a relatively young paradigm [19], finding its applications mainly in natural language processing [24] and image recognition [23]. Its aim is to minimize the cost of preparing labeled training sets for supervised machine learning models, while preserving the resulting model efficiency. Surprisingly, such an approach is not common in cheminformatics, where the process of samples labeling is extremely expensive due to the cost of biological experiments (buying/synthesizing chemical compounds and performing *in vitro* experiments). Even though, there are examples of application of active learning in the evaluation of compounds biological activity [26], we argue that considered setting is unrealistic and thus obtained results are not reliable.

In this paper we try to build common language for machine learning and cheminformatics research communities. The paper is structured as follows. First, we introduce some basic concepts and notations from active learning paradigm. Then, we briefly describe the task of chemical compound activity prediction. In the next sections, we introduce proposed experimental setting and active learning strategies used, whereas final parts include experimental evaluation and conclusions.

2 Active Learning

The classic supervised machine learning setting assumes that one is given a training set by some sampling process completely independent on the training procedure. However, in real life problems it is often the case that one has access to enormous amounts of unlabeled examples, and only obtaining labels is an expensive, time consuming "sampling process". In particular, one can guide this process through selection of samples which should be labeled in order to maximize model efficiency while in the same time – minimize the number of samples requiring labeling. One example of such case is huge amount of unlabeled text available on the Internet, which can be downloaded without any problem, but if labeling of any type is needed – it requires a time-consuming process of linguists annotations. If one provides a closed loop between ML model and the process of training set construction, then an active learning method is obtained [19].

From more theoretical point of view, active learner is often defined in terms of utility function

$$u : \mathcal{X} \to \mathbb{R},$$

such that $u(x)$ denotes the valuability of the knowledge of x label. Consequently, in each iteration, one adds to the training set point x, maximizing $u(x)$ over $\mathcal{U}$ – set (often called *pool*) of unlabeled samples.

One important generalization of the above problem is so called k-batch scenario, where in each iteration learner has to select a subset of k points instead of just a single one, what is done analogously through definition of utility function

over subsets

$$u : 2^{\mathcal{X}} \to \mathbb{R}.$$

Such approaches are proven to extremely reduce the number of labels required for the construction of strong predictive models [17, 18, 14]. In this paper, we focus on using such a method in the field of cheminformatics, in particular for the problem of chemical compounds activity prediction.

3 Chemical compounds activity prediction

The increasing amount of data in the fields of cheminformatics make machine learning tools more and more popular. These methods are often used to predict whether a given chemical compound is active towards a given protein target. From ML perspective, this can be interpreted as a binary classification where inputs samples are compounds (represented in an appropriate way), and labels denoting whether a compound could be a drug candidate (positive) or not (negative), that is whether it is able to bind with the target protein or not.

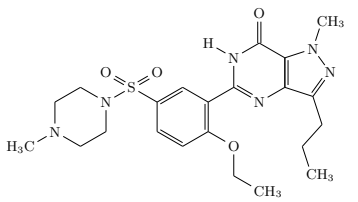
There are two very important aspects of such problem. The first one, is connected with the way data are collected and the second with how the data are represented. We briefly investigate both of these issues.

One of the fundamental assumptions of most of the ML methods is that data are generated iid from some underlying distributions. Unfortunately for cheminformatic problems, it is not the case. There are two main reasons leading to heavy violation of this assumption [10]. First, researchers look for possible drug candidates in selected parts of chemical space which is the most probable to contain such objects (potential drug candidates). In other words, they often investigate *neighbourhoods* of known drugs, as well as exploit other expert/biological and chemical knowledge. Consequently, space of input sample is extremely skewed and does not represent the actual distribution of compounds (nor active ones). Second problem comes from positive result bias common in science – databases contain mostly record regarding active compounds (as such results can be relatively easily published), as well as inactive compounds which are highly similar to the active ones (so their inactivity is an interesting fact). Unfortunately, as the result, we lack enormous amount of information regarding inactive compounds.

Most of the ML approaches require data to be a subset of $\mathbb{R}^d$. In other words, we need to embed chemical compounds, which are very complex structures, into such space. Researchers proposed multiple ways of such transformations (fingerprints) [9, 22, 11, 6]. One popular family of such objects is constituted by binary fingerprints, consisting of a sequence of d predicates $\phi_i(\cdot)$ (descriptors), which project compounds onto the vertices of the d-dimensional hypercube. For a given compound $x \in \mathcal{X}$, such embedding is given by

$$\varphi : \mathcal{X} \ni x \to [1_{\phi_1(x)}, 1_{\phi_2(x)}, \ldots, 1_{\phi_d(x)}]^T \in \{0,1\}^d \subset \mathbb{R}^d,$$

where $1_{\phi_i(x)}$ equals 1 if $\phi_i(x)$ is true and 0 otherwise. See Fig. 1 for an example of multiple types of possible predicates used.



x'

	i	1	2	3	...	d
predicates $\phi_i(x)$		$\subset x$	$\subset x$	$\lvert\mathrm{N}\rvert_c \geq 3$	...	$\subset x$
fingerprint $\varphi(x')$		1	0	1	...	1

Fig. 1. Sample fingerprint of the chemical molecule x'. $|A|_x$ denotes the number of atoms/substructures A in x, so in particular $A \subset x \iff |A|_x \geq 1$.

Due to the characteristics of the binary representation, one needs a specific methods of measuring similarity between objects described in such a way. In particular, in order to use Support Vector Machines (SVM), one should use a kernel designed for binary sequences. One of the well known methods, which is very successful in cheminformatic applications [1, 4] is Jaccard coefficient and corresponding Jaccard (or Tanimoto) kernel J, defined for two sets A and B as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|},$$

which in an obvious way can be translated to the operation over binary vectors $\bar{A}$ and $\bar{B}$

$$J(\bar{A},\bar{B}) = \frac{\sum_{i=1}^{d} \min\{\bar{A}_i, \bar{B}_i\}}{\sum_{i=1}^{d} \max\{\bar{A}_i, \bar{B}_i\}}.$$

However, there are more useful measures, which also denote valid kernels, one of which is Sørensen coefficient S

$$S(A,B) = \frac{|A \cap B|}{|A| + |B|},$$

and analogously

$$S(\bar{A},\bar{B}) = \frac{\sum_{i=1}^{d} \min\{\bar{A}_i, \bar{B}_i\}}{\sum_{i=1}^{d} \bar{A}_i + \sum_{i=1}^{d} \bar{B}_i}.$$

These two measures have been shown to perform very well in various tasks [5, 16] and both of them will be used in this paper in two ways: as measures of compounds similarity and as SVM kernel.

4 Proposed experimental setting

Active learning has been proposed for the exact same problem in the past [26]. Proposed approach is mathematically valid and is an important first step in applying active learning to the problem of drug discovery. However, in authors opinion, previous work did not capture the true nature of the virtual screening process. First of all, existing approach deals with single-query active

learning, which is completely unrealistic assumption. The compounds are never bought/synthesized and tested one by one - chemists buy or synthesize whole groups of compounds. The k-batch setting is crucial in order to truly simulate the procedure. Secondly, previous works assume the iid of the samples and so - that one can use whole set of known active/inactive compounds to model the true distribution of compounds. This is also false, as described in previous sections, due to high bias in the way compounds are tested. In particular, such experiments do not answer the fundamental question:

> *Does given active learning strategy leads to the discovery of new, uknown drug candidates?*

We propose to model the problem using two important modifications to previous works:

1. one should use k-batch active learning scenario,
2. one has to identify a specific group of compounds which can be used to estimate the ability to find new drug candidates.

Group mentioned above should consist of compounds which:

- form a group including active compounds (favourably a chemical group),
- are not present in the training set,
- are common enough to ensure the reliable estimation of generalization capabilities.

Let us first describe how one can find such cluster. We have performed a hierarchical clustering of data $\mathcal{U}$ using Agglomerative Clustering algorithm with maximum (complete) linkage criterion. Jaccard similarity measure was used as metric. Pair of clusters $\mathcal{S}, \aleph$ was selected as two disjoint subtrees meeting two criteria: clusters $\mathcal{S}$ and $\aleph$ constitute respectively in at least 40% and 10% of original data and the ratio of average inter–cluster distance to average distance between samples is the biggest.

This heuristic yields in most cases sensible clustering, which was further confirmed by visualization as can be seen in Fig. 2. However, it should be noted that for more robust generalization power estimation, manual clustering should be performed. In our case, it often happens that clusters are noisy, as for instance $\mathcal{S}$ might contain few samples close to $\aleph$, while manual clustering done by chemist wouldn't include such a situation. Noisy clustering is battled in our case by performing exhaustive number of experiments with multiple proteins and fingerprints.

Simulation starts from a random sample from $\mathcal{S}$, which simulates (represents) the current chemical knowledge about compounds activity. During active learning process one should monitor efficiency on $\mathcal{S}$, denoting model ability to correctly classify compounds similar to the known ones (local search of new drug candidates) as well as efficiency on $\aleph$, denoting model ability to actually discover new drugs. One can further split each of these two parts into train and test parts, one (train) available in a samples pool (their labels can be obtained during active learning process) and other (test) are only used to estimate the generalization capabilities of the model.

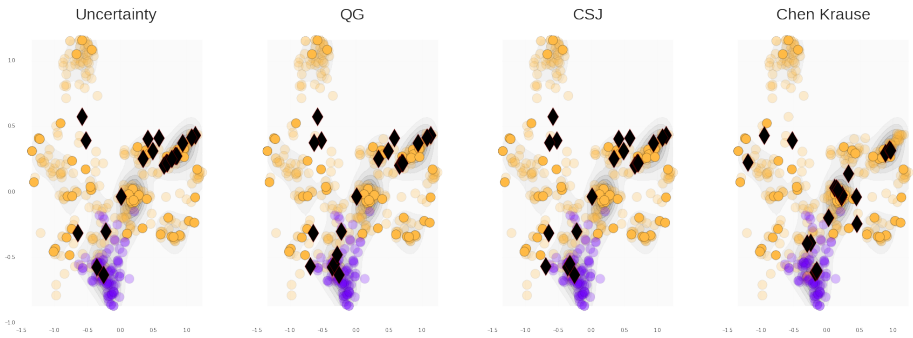


Fig. 2. Visualization of the clustering, ℵ cluster is denoted by purple dots, yellow ones show remaining part of $\mathcal{U}$. Semi-transparent objects denote unlabelled examples and finally black diamonds are samples selected by each strategy.

5 Proposed active learning strategy

There are dozens of very efficient, successful strategies for active learning where one selects a simple instance in each iteration. However, in batch scenario, where one selects k points in each iteration even the simplest approaches are computationally expensive [8] or even NP-hard [2]. For this reason, it is a common choice to use a simple, single instance-based strategy, to rank points and select k most promising ones. Unfortunately, such an approach leads to the selection of highly correlated data, which can work even worse than passive learning [20]. This problem is somehow similar to many others in ML, in particular the construction of ensemble of learners [13]. In both of the above-mentioned cases, one needs to select a set of objects which provide some knowledge, but at the same time, the diversification should be ensured. In the context of active learning, it is a common practice [2] to look for a set of samples maximizing[1]

$$u_C(\mathcal{A}) = (1-C)\frac{1}{|A|}\sum_{a\in\mathcal{A}} u(a) + C\frac{2}{|A|(|A|-1)}\sum_{a,b\in\mathcal{A}\times\mathcal{A}} d(a,b),$$

where C is a parameter denoting balance between maximizing utility $u(\cdot)$ and inner batch distances $d(\cdot,\cdot)$. Unfortunately, finding solution of such a problem is known to be NP-hard. Thus researchers often use heuristic simplifications, with a very popular quasi-greedy solution [12, 2]. In such an approach one builds query set $\mathcal{A}$ iteratively by first selecting sample maximizing $u(\cdot)$ and then in ith iteration (thus $|\mathcal{A}| = i-1$) one selects $a = \arg\max_{a\in\mathcal{U}} u_C(\mathcal{A}\cup\{a\})$. It is easy to notice that such an approach requires $\mathcal{O}(k^2|\mathcal{U}|)$ time and leads to very rough estimation of the true solution. One can improve the above method through introduction of randomized starts at the cost of additional computations. The idea is to select first sample with probability proportional to $u(\cdot)$ value and then

[1] In the original work min distance was used instead of a mean.

use quasi-greedy approach. After multiple such starts one selects the one yielding maximum $u_C(\cdot)$ value.

We propose to follow a different generalization path instead. In order to enforce internal diversification of the batch, we merge the quasi-greedy strategy with non-Euclidean clustering. The idea is to first split dataset into M clusters so selecting mini-batches from each of them should yield distant samples and then run quasi-greedy approach in each of them so also internal distances inside each mini-batch are big. Following Alg. 1 shows the exact procedure.

Algorithm 1 Cluster-based Sørensen-Jaccard sampling

procedure $\mathrm{CSJ}_M(\mathcal{U}, k)$
  $\mathcal{A} \leftarrow \{\}$
  $U_1, \ldots, U_M \leftarrow$ **find** M **clusters using Sørensen**$(\mathcal{U})$
  for $i = 1$ **to** M **do**
    $\mathcal{Q} \leftarrow$ **select** k/M **samples by Quasi-greedy using Jaccard**$(\mathcal{U}_i)$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{Q}$
  end for
  return $\mathcal{A}$
end procedure

There are many ways of performing clustering based on Sørensen coefficient. One of them is to build a Sørensen kernel [16] and run a kernelized k-means algorithm [7]. Another approach [5], yielding similar results in much shorter time, is to randomly select a subset of compounds $\{C_i\}_{i=1}^h$, span a new space through projection

$$\varphi(x) = [S(x, C_1), \cdots, S(x, C_h)]^T,$$

and use a simple k-means (or any other clustering technique) in the projected space. In this paper, we follow the second path due to the simplicity and efficiency of such an approach.

6 Experiments

Let us briefly outline the experimental setting. We use datasets consisting of chemical compounds of experimentally confirmed activity/inactivity towards six different proteins, leading to six, binary, classification problems. We use ExtFP, MACCSFP and PubchemFP [27] fingerprints to embed compounds in the $\{0, 1\}^d$ space. As a main model we use SVM with Jaccard kernel, due to its known applicability in the domain. We analyze three different sizes of batches (number of compounds selected in each iteration), namely $k = 20, 50, 100$. SVM is retrained at each iteration and its hyperparameter C is fitted using internal 5-fold cross validation. All experiments are performed with repeated, randomized, stratified train/test splits in order to minimize the variance of the results. We investigate five selection strategies:

- passive learner, simply selecting samples at random,
- greedy uncertainty sampling, as a baseline method [19],
- rand greedy, described in previous sections, as a stronger version of quasi-greedy strategy [2],
- proposed, CSJ sampling, with $M = 2$ (just two clusters using Sørensen coefficient),
- probabilistic method of Chen and Krause [3], generalized to the nonlinear scenario through performing Jaccard based non-linear projection [4], and fitting their linear approximator on the top [3].

All of them are implemented using Python with help of scikit-learn [15]. One can find source code of all the above approaches at github[2].

As outlined in the previous sections, we investigate behavior of the proposed methods on the test set of $\mathcal{U}$, $\aleph$ cluster and on unlabeled part of samples from $\aleph$. We will now briefly discuss results and emerging conclusions.

Let us first investigate how the proposed methods deal with building a concept of the activity in the whole compounds space. Table 1 summarizes the average ranking (position, obtained after performing the whole experiment and ordering strategies according to the given criterion) for results measured on the test part of the whole $\mathcal{U}$ set. Two different results are analyzed, first – final WAC[3] of the model after the experiment and area under the WAC curve (which is equivalent to the mean WAC over the experiment – measuring how fast is given strategy leading to good results). These results show how good is each strategy

batch size	20	50	100	avg
CSJ_2 sampling	**2.33**	**2.17**	**2.17**	**2.22**
Rand Greedy	**2.33**	3.33	**2.17**	2.61
Chen Krause	2.50	2.33	3.50	2.78
Uncertainty	3.17	3.67	3.17	3.33
Passive	4.67	3.50	4.00	4.06

batch size	20	50	100	avg
CSJ_2 sampling	2.17	**2.17**	**2.00**	2.11
Rand Greedy	**1.33**	**2.17**	**2.00**	**1.83**
Chen Krause	4.00	3.00	3.00	3.33
Uncertainty	3.33	3.33	4.17	3.61
Passive	4.17	4.33	3.83	4.11

Table 1. Average ranking of final WAC score (on the left) and AUC score (on the right) for each strategy over all considered experiments on the test part of $\mathcal{U}$ for given batch size.

in building a general concept of activity. Here one can notice that rand greedy strategy obtains better AUC scores, meaning that it is able to faster converge to good model. On the other hand CSJ is a close second place, and outperforms all methods when it comes to final WAC score. One should note also that CSJ behaves much better once batch size is big enough. Proposed strategy is much better in diversifying samples in a batch, so with bigger batches its strength is better captured. It is quite interesting that strategy proposed by Chen and

[2] `http://github.com/gmum/mlls2015/`

[3] WAC=$\frac{1}{2}\frac{\mathrm{TP}}{\mathrm{TP+FN}} + \frac{1}{2}\frac{\mathrm{TN}}{\mathrm{TN+FP}}$

Krause behaves worse than rand greedy. There might be multiple reasons for such behavior. First, this method requires fitting of many hyperparameters, which might be performed suboptimally as during active learning scenario it is hard to fit multiple hyperparameters of the strategy. Second, proposed delinearization is not fully consistent with the kernelized SVM, one should probably change whole strategy to the kernel space, but it would drastically increase the computational complexity. Finally, their strategy does not include much diversification in the batches, which we argue is a crucial element for the considered problem.

Let us now focus on the main element of the proposed scenario – evaluation of the ℵ cluster, measuring how good is a particular strategy in finding actual new drugs. It is worth stressing, that using Sørensen clustering in CSJ is supposed to simulate the fact that we do not know the true measure of "diversity" of compounds. We use Jaccard coefficient to build ℵ cluster, so if we use Jaccard also for clustering, obtained results would be less reliable (we did also perform such experiments, and obtained results were actually very similar to the ones reported here). At the same time, Sørensen coefficient is quite similar to Jaccard's, which is supposed to model real life situation, where we do have a measure which well captures a compounds similarity [1], but is not the exact same one that described the actual diversity. Table 2 shows analogous results to the previous Table, but measured on the ℵ cluster. One can notice significant

batch size	20	50	100	avg
CSJ_2 sampling	**2.00**	**2.17**	**2.17**	**2.11**
Rand Greedy	2.33	2.50	2.83	2.56
Chen Krause	3.33	3.67	4.50	3.83
Uncertainty	3.83	4.17	2.83	3.61
Passive	3.50	2.50	2.67	2.89

batch size	20	50	100	avg
CSJ_2 sampling	**1.17**	**1.50**	**2.00**	**1.56**
Rand Greedy	2.00	2.17	2.17	2.11
Chen Krause	4.33	4.00	2.83	3.72
Uncertainty	3.33	3.50	3.67	3.50
Passive	4.17	3.83	4.33	4.11

Table 2. Average ranking of final WAC score (on the left) and AUC score (on the right) for each strategy over all considered experiments on the test part of ℵ cluster for given batch size.

difference between results obtained by CSJ and all competing approaches. It strongly suggests that proposed approach is much better in exploration of the input space. It is worth noting, that when it comes to final WAC score, passive learning is better than greedy uncertainty as well as Chen and Krause method. Difference between rand greedy and passive is also barely significant, showing that their exploration is very limited. On the other hand when it comes to the speed of converge (measured as AUC) passive learning loses with all the competing methods, as can be seen in Figure 3. So it seems that the exploration issues of most of the considered strategies appear in the "later" part of the experiment (they seem to discover the cluster and focus on it more than passive, but they leave it to too early; only CSJ consistently analyzes its samples).

Finally, we briefly analyze the strategies ability to eliminate unlabeled samples from ℵ. High scores in such an experiment are important if we assume that there is a finite amount of interesting drug candidates, and they are all available in the pool $\mathcal{U}$. Then, "buying" labels of such samples is equivalent to actually discovering all interesting drugs. Results in Table 3 are final confirmation of

batch size	20	50	100	avg
CSJ_2 sampling	**2.00**	**2.00**	**1.50**	**1.83**
Rand Greedy	2.83	3.50	2.33	2.33
Chen Krause	3.17	3.50	3.50	3.39
Uncertainty	3.33	3.50	4.17	3.67
Passive	3.67	2.50	3.50	3.22

batch size	20	50	100	avg
CSJ_2 sampling	1.67	**1.50**	**1.50**	**1.56**
Rand Greedy	**1.50**	2.50	2.17	2.06
Chen Krause	4.33	4.00	3.33	3.89
Uncertainty	2.83	2.50	3.17	2.83
Passive	4.67	4.50	4.83	4.67

Table 3. Average ranking of final WAC score (on the left) and AUC score (on the right) for each strategy over all considered experiments on unlabeled elements of ℵ cluster for given batch size.

CSJ ability to fast exploration of the input space and consequently identifying drugs from the ℵ cluster. Once again most of the strategies led to worse (or comparable) final WAC results to the passive learning in this subtask.

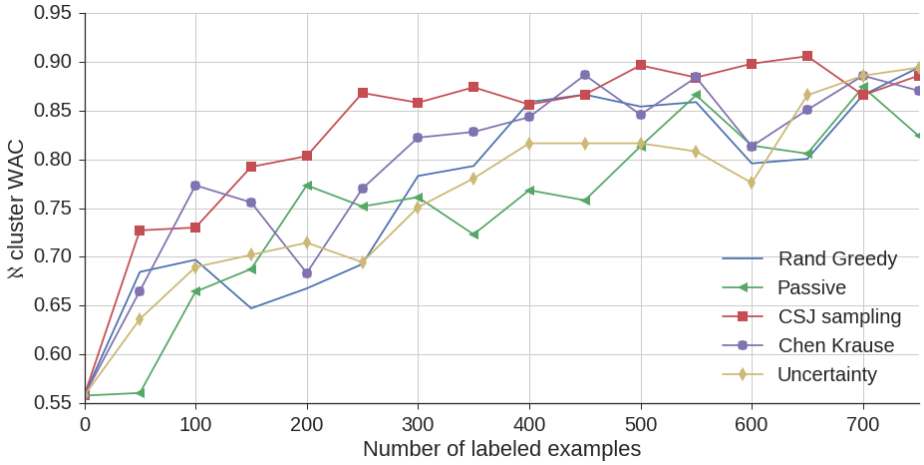


Fig. 3. Results of model prediction on ℵ cluster for 5 tested quering strategies on a single protein with batch size set to 50. While eventually all strategies achieve similar result CSJ stays strong throughout the AL process.

7 Conclusions

There are two main contributions of this paper. First, we introduced and described an experimental setting for active learning based drug candidates identification procedure. The proposed method is the first that does not make unrealistic assumptions of previous research in the area and shows a proof of concept of the solution. However, in order to obtain fully scientifically sound setting, one should replace automatic clustering with expert based identification of compounds group (which might be very hard due to the very limited knowledge of active compounds in the whole input space).

Second contribution is introducing simple active learning k-batch strategy, exploiting both Sørensen and Jaccard coefficients, that achieves significantly better scores than competing approaches in conducted experiments. It would be valuable to further investigate other methods of diversifying samples inside the batch and efficiently estimate their informativeness.

Acknowledgments. Work of first two authors was partially supported by National Science Center Poland Found grant no. 2013/09/N/ST6/03015.

References

1. Bajusz, D., Racz, A., Heberger, K.: Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? Journal of Cheminformatics 7(1), 20 (2015), http://www.jcheminf.com/content/7/1/20
2. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: ICML. vol. 3, pp. 59–66 (2003)
3. Chen, Y., Krause, A.: Near-optimal batch mode active learning and adaptive submodular optimization. In: ICML. pp. 160–168 (2013)
4. Czarnecki, W.M.: Weighted tanimoto extreme learning machine with case study of drug discovery. IEEE Computational Intelligence Magazine (2015)
5. Czarnecki, W.M., Rataj, K.: Compounds Activity Prediction in Large Imbalanced Datasets with Substructural Relations Fingerprint and EEM (2015)
6. Ewing, T., Baber, J.C., Feher, M.: Novel 2D fingerprints for ligand-based virtual screening. Journal of chemical information and modeling 46(6), 2423–2431 (2006)
7. García, M.L.L., García-Ródenas, R., Gómez, A.G.: K-means algorithms for functional data. Neurocomputing 151, 231–245 (2015)
8. Guo, Y., Schuurmans, D.: Discriminative batch mode active learning. In: Advances in neural information processing systems. pp. 593–600 (2008)
9. Hall, L.H., Kier, L.B.: Electrotopological State Indices for Atom Types: A Novel Combination of Electronic, Topological, and Valence State Information. Journal of Chemical Information and Modeling 35(6), 1039–1045 (1995)
10. Jastrzebski, S., Czarnecki, W.M.: Analysis of compounds activity concept learned by SVM using robust Jaccard based low-dimensional embedding. Schedae Informaticae (2015)
11. Klekota, J., Roth, F.P.: Chemical substructures that enrich for biological activity. Bioinformatics (Oxford, England) 24(21), 2518–2525 (Nov 2008)

12. Kremer, J., Steenstrup Pedersen, K., Igel, C.: Active learning with support vector machines. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 4(4), 313–326 (2014)
13. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine learning 51(2), 181–207 (2003)
14. McCallumzy, A.K., Nigamy, K.: Employing em and pool-based active learning for text classification. In: ICML. pp. 359–367. Citeseer (1998)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research 12, 2825–2830 (2011)
16. Ralaivola, L., Swamidass, S.J., Saigo, H., Baldi, P.: Graph kernels for chemical informatics. Neural Networks 18(8), 1093–1110 (2005)
17. Roy, N., McCallum, A.: Toward optimal active learning through monte carlo estimation of error reduction. ICML, Williamstown pp. 441–448 (2001)
18. Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: ICML. pp. 839–846. Citeseer (2000)
19. Settles, B.: Active learning literature survey. University of Wisconsin, Madison 52(55-66), 11 (2010)
20. Settles, B.: Active learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 6(1), 1–114 (2012)
21. Smusz, S., Kurczab, R., Bojarski, A.J.: The influence of the inactives subset generation on the performance of machine learning methods. Journal of Cheminformatics 5, 17 (2013)
22. Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., Willighagen, E.: The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics. Journal of chemical information and computer sciences 43(2), 493–500 (2003)
23. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: Proceedings of the ninth ACM international conference on Multimedia. pp. 107–118. ACM (2001)
24. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. The Journal of Machine Learning Research 2, 45–66 (2002)
25. Wang, M., Yang, X.G., Xue, Y.: Identifying hERG Potassium Channel Inhibitors by Machine Learning Methods. QSAR & Combinatorial Science 27(8), 1028–1035 (Aug 2008)
26. Warmuth, M.K., Liao, J., Rätsch, G., Mathieson, M., Putta, S., Lemmen, C.: Active learning with support vector machines in the drug discovery process. Journal of Chemical Information and Computer Sciences 43(2), 667–673 (2003)
27. Yap, C.W.: Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. Journal of computational chemistry 32(7), 1466–1474 (2011)