Czarnecki *et al. J Cheminform (2015) 7:38*
DOI 10.1186/s13321-015-0088-0

Journal of Cheminformatics

RESEARCH ARTICLE

Open Access

Robust optimization of SVM hyperparameters in the classification of bioactive compounds

Wojciech M Czarnecki[1], Sabina Podlewska[2,3] and Andrzej J Bojarski[2*]

Abstract

Background: Support Vector Machine has become one of the most popular machine learning tools used in virtual screening campaigns aimed at finding new drug candidates. Although it can be extremely effective in finding new potentially active compounds, its application requires the optimization of the hyperparameters with which the assessment is being run, particularly the C and γ values. The optimization requirement in turn, establishes the need to develop fast and effective approaches to the optimization procedure, providing the best predictive power of the constructed model.

Results: In this study, we investigated the Bayesian and random search optimization of Support Vector Machine hyperparameters for classifying bioactive compounds. The effectiveness of these strategies was compared with the most popular optimization procedures—grid search and heuristic choice. We demonstrated that Bayesian optimization not only provides better, more efficient classification but is also much faster—the number of iterations it required for reaching optimal predictive performance was the lowest out of the all tested optimization methods. Moreover, for the Bayesian approach, the choice of parameters in subsequent iterations is directed and justified; therefore, the results obtained by using it are constantly improved and the range of hyperparameters tested provides the best overall performance of Support Vector Machine. Additionally, we showed that a random search optimization of hyperparameters leads to significantly better performance than grid search and heuristic-based approaches.

Conclusions: The Bayesian approach to the optimization of Support Vector Machine parameters was demonstrated to outperform other optimization methods for tasks concerned with the bioactivity assessment of chemical compounds. This strategy not only provides a higher accuracy of classification, but is also much faster and more directed than other approaches for optimization. It appears that, despite its simplicity, random search optimization strategy should be used as a second choice if Bayesian approach application is not feasible.

Keywords: Compounds classification, Virtual screening, Support Vector Machine, Parameters optimization, Bayesian optimization

Background

The application of computational methods at various stages of drug design and development has become a vital part of the process. As the methods developed become constantly more effective, despite the aims at optimizing their performance, the focus of the attention shifts away from performance optimization to the minimization of requirements for computational resources. The attainment of both effectiveness and the desired speed has been responsible for the recent extreme popularity of machine learning (ML) methods in computer-aided drug design (CADD) approaches. Machine learning methods are mostly used for virtual screening (VS) tasks, in which they are supposed to identify potentially active compounds in large databases of chemical structures. One of the most widely used ML methods in CADD is

*Correspondence: bojarski@if-pan.krakow.pl
[2] Department of Medicinal Chemistry, Institute of Pharmacology, Polish Academy of Sciences, 12 Smetna Street, 31-343 Krakow, Poland
Full list of author information is available at the end of the article

the Support Vector Machine (SVM). Although it has a potential of providing very high VS performance, its application requires the optimization of the parameters used during the training process, which was proved to be crucial for obtaining accurate predictions. To date, various approaches have been developed to make SVM faster and more effective. In cheminformatics applications, the most popular optimization strategies are grid search [1, 2] and heuristic choice [3, 4]. Depending on the problem, they are able to provide high classification accuracy—for example Wang et al. obtained 86% of accuracy in the classification of hERG potassium channel inhibitors for the heuristic choice of the SVM parameters [4]. On the other hand, Hamman et al. [1] were able to evaluate the cytochrome P450 activities with 66–83% of accuracy using grid search method of SVM parameters optimization. The need for optimizing SVM parameters is undeniable, as classification efficiency can change dramatically for various parameters values. A high computational cost of a systematic search over a predefined set of parameters' values is a trigger for development of new optimization algorithms. In recent years, Bayesian optimization [5, 6] (including gaussian processes [7]) and random search-based selection [8] have become more popular [9, 10]. As those approaches were not explored so far in the field of cheminformatics, we analyze their impact on classification accuracy and, more importantly, the speed and ease of use, that these approaches have lent to the optimization of SVM hyperparameters in the search for bioactive compounds.

Hyperparameters optimization

In the classical ML approach to a classification problem, we are given a training set $\{(x_i, y_i)\}_{i=1}^{N}$ (with x_i representing samples' features, in our case—fingerprint, and y_i being the class assignment) and we try to build a predictive model based on these data using a training algorithm that sets the parameters w (for example the weight of each fingerprint element) for fixed hyperparameters λ (for example a type of SVM kernel, the regularization strength C or the width of the RBF kernel γ). In other words, given an objective $\mathcal{S}$ that must be maximized, we are supposed to solve the following problem:

$$\underset{w}{\text{maximize}}\, \mathcal{S}\Big(w \big| \big\{(x_i, y_i)\big\}_{i=1}^{N}, \lambda\Big).$$

While this problem is often easily solvable (for example, in SVM, $\mathcal{S}$ is a concave function, and thus, we can find the maximum by using a simple steepest ascent algorithm), in general, it is very hard to find an optimal λ. This difficulty stems from the very complex shape of the $\mathcal{S}$ function once we treat λ as its arguments, which results in the joint optimization of the model parameters (w) and the set of hyperparameters (λ):

$$\underset{w,\lambda}{\text{maximize}}\, \mathcal{S}\Big(w, \lambda \big| \big\{(x_i, y_i)\big\}_{i=1}^{N}\Big).$$

A basic method for solving this problem is a grid search-based approach, which simply samples the set of possible λ values in a regular manner. For example, we choose the parameter C for a SVM in a geometrical progression, obtaining the values $\lambda_1, \ldots, \lambda_k$ and returning the best solution among each of the subproblems:

$$w_i = \arg\max_{w} \mathcal{S}\Big(w \big| \big\{(x_i, y_i)\big\}_{i=1}^{N}, \lambda_i\Big).$$

While such an approach guarantees finding the global optimum for $k \to \infty$, it might be extremely computationally expensive, as we need to train k classifiers, each of which can take hours. Instead, we can actually try to solve the optimization problem directly by performing an adaptive process that on one hand tries to maximize the objective function and on the other hand samples the possible λ space intelligently in order to minimize the number of classifier trainings. The main idea behind Bayesian optimization for such a problem is to use *all of the information* gathered in previous iterations for performing the next step. It is apparent that grid search-based methods violate this assumption as we do not use any knowledge coming out from the results of models trained with other λ values.

We can consider this problem as the process of finding the maximum for $f(\lambda)$, defined as

$$f(\lambda) = \max_{w} \mathcal{S}\Big(w \big| \big\{(x_i, y_i)\big\}_{i=1}^{N}, \lambda\Big).$$

Unfortunately, f is an unknown function and we cannot compute its gradient, Hessian, or any other characteristics that could guide the optimization process. The only action we can perform is to obtain a value for f at a given point. However, doing so is very expensive (because it requires training a classifier); thus, we need a fast (with respect to evaluating the function), derivative-free optimization technique to solve this problem.

For the task under consideration, $\mathcal{S}$ is the accuracy of the resulting SVM model with the RBF kernel, and $\lambda = \{C, \gamma\}$ is the set of two hyperparameters that we must fit to optimize the SVM performance to predict the bioactivity of compounds, which (loosely speaking) is measured by f.

Results and discussion

Six SVM optimization approaches were evaluated in the classification experiments of compounds possessing

Table 1 Details of the classification experiments performed

Targets	Fingerprints	Optimization method	C and γ range
			No of iterations
5-HT_{2A}	EstateFP	Bayes	$\log_{10}(C) \in [-2, 5]$
5-HT_{2C}	ExtFP	Random	$\log_{10}(\gamma) \in [-10, 3]$
5-HT_6	KlekFP	Grid search	20, 30, 50, 75, 100, 150
5-HT_7	MACCSFP	Small grid	
CDK2	PubchemFP	SVMlight	
M_1	SubFP	libSVM	
ERK2			
AChE			
A_1			
alpha2AR			
beta1AR			
beta3AR			
CB1			
DOR			
D_4			
H_1			
H_3			
HIVi			
IR			
ABL			
HLE			

activity towards 21 protein targets, represented by six different fingerprints (Table 1).

Classification effectiveness analysis

A global analysis of the classification efficiency revealed that Bayesian optimization definitely outperformed the other methods of SVM parameters' optimization (Fig. 1). For a particular target and fingerprint, Bayesian approach provided a higher classification accuracy in 80 experiments, a significantly greater number than the other strategies (22 for the runner-up: grid search). On the other hand, the SVMlight and libSVM were definitely the least effective methods of SVM usage; they did not provide the highest accuracy values for any of the target/fingerprint combinations. This result is an obvious consequence of the fact that SVMlight and libSVM are just basic heuristics and their results cannot be comparable with any hyperparameters optimization technique. Interestingly, libSVM achieved much better results than SVMlight even though its heuristic is much simpler.

The relationships between various methods tested were preserved when the results were analyzed with respect to the various fingerprints (Fig. 2)—the Bayesian optimization always provided the highest classification accuracy (for 13–14 targets for each of the fingerprints analyzed), whereas the 'global' second-place method—grid search—was outperformed by 'small grid'-cv for two fingerprints: MACCSFP and PubchemFP. The runners-up (grid search or 'small grid'-cv, depending on fingerprint) provided the best predictive power of the model for 3 proteins on average. The ineffectiveness of the SVMlight and libSVM strategies has been already indicated in the 'global' analysis, and with respect to various fingerprints, there was no protein for which those SVM optimization methods provided the highest classification accuracy.

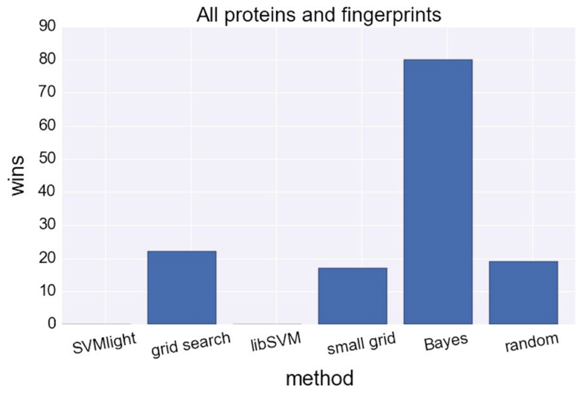

Fig. 1 Global analysis of classification accuracy obtained for different methods for SVM parameters optimization expressed as the number of experiments in which a particular strategy provided the highest accuracy values.

The situation becomes more complex when separate targets are taken into account (Fig. 3). The Bayesian optimization provided the best results for all considered representations for some proteins (CDK2, H_1, ABL); however, in few cases, other optimization approaches for tuning SVM parameters outperformed the Bayesian method (5-HT_6—random and grid search, beta1AR—'small grid'-cv and grid search, beta3AR—grid search, HIVi—grid search, 'small grid'-cv, random, MAP kinases ERK2—'small grid'-cv and random search). These results show that a more careful model accuracy approximation is required for some proteins. Because we are interested in maximizing the accuracy on a naive test set, we approximate this set by performing internal cross-validation for each method. This is a well-known technique in ML; however, it might be *not reliable* for small datasets. Beta1AR, beta3AR, and HIVi are very small datasets in our comparison; thus, it seems probable that the poor results of the Bayesian approach (a poor approximation of the $\mathcal{S}$ value) were caused by the high internal variance in the dataset rather than because the Bayesian approach was actually worse than the grid search method.

Because grid search was the second-place method in the majority of the analyses, both for global analysis, and fingerprint- and target-based comparisons, a

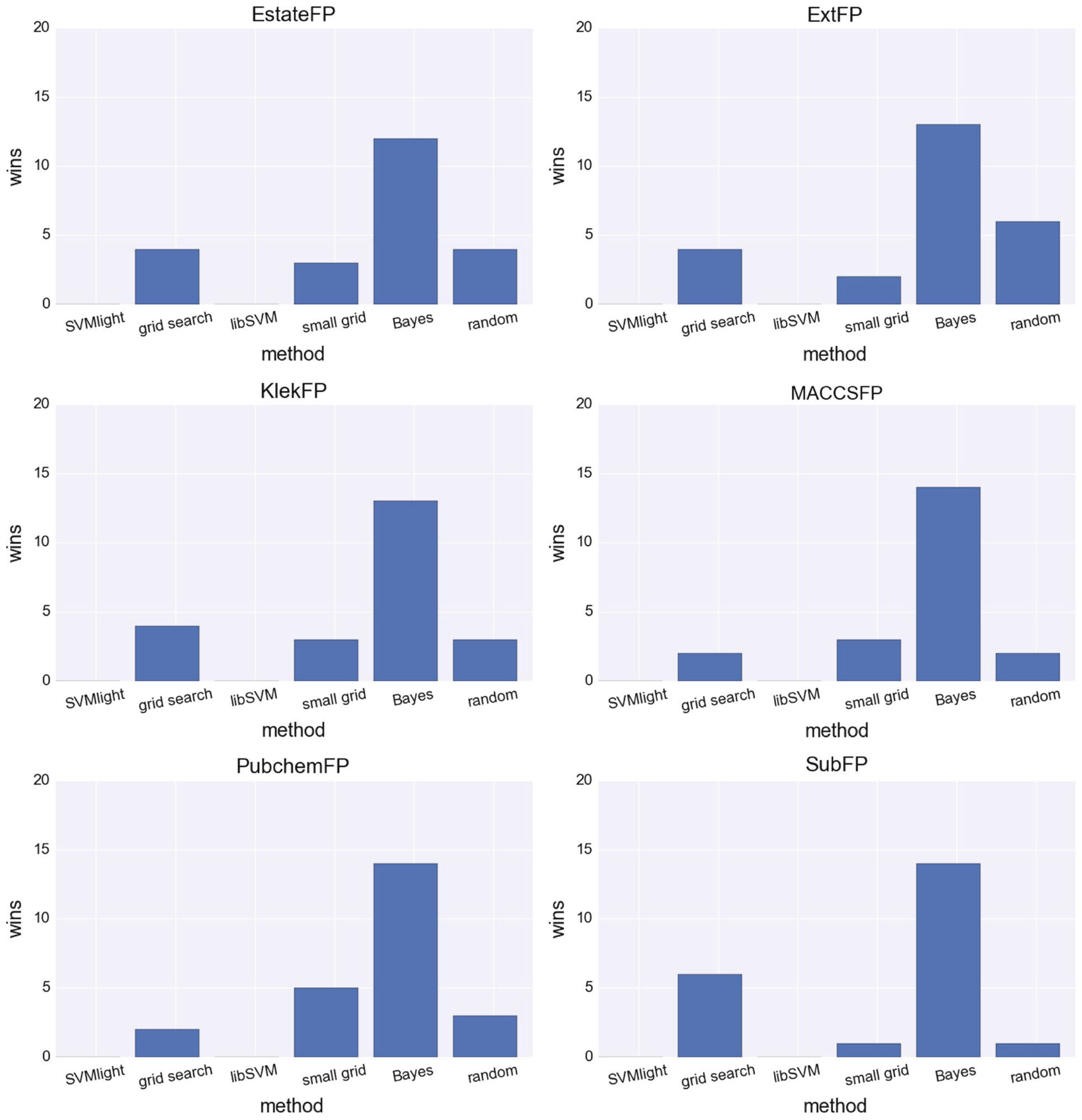

Fig. 2 Analysis of the effectiveness of different SVM optimization strategies with respect to various fingerprints expressed as the number of experiments in which a particular strategy provided the highest accuracy values for a given compounds representation.

direct comparison of the number of the highest accuracies obtained for Bayesian optimization and the grid search approach was performed (Table 2). The sum of the number of wins is not equal for the given fingerprint-based or target-based comparison as the draws were also considered.

The comparison of the number of ’wins’ for Bayesian optimization over the grid search indicates the superiority of the former approach. In the ‘global’ analysis, the Bayesian optimization strategy gave a higher accuracy for approximately a 3-fold higher number of experiments than the grid search. For the fingerprint-based analysis, the ratio of Bayesian/grid search wins was similar to the best ratio (in favor of Bayesian optimization) obtained for MACCSFP (18 : 4) and the worst (15 : 7 and 15 : 6) for EstateFP and SubFP, respectively.

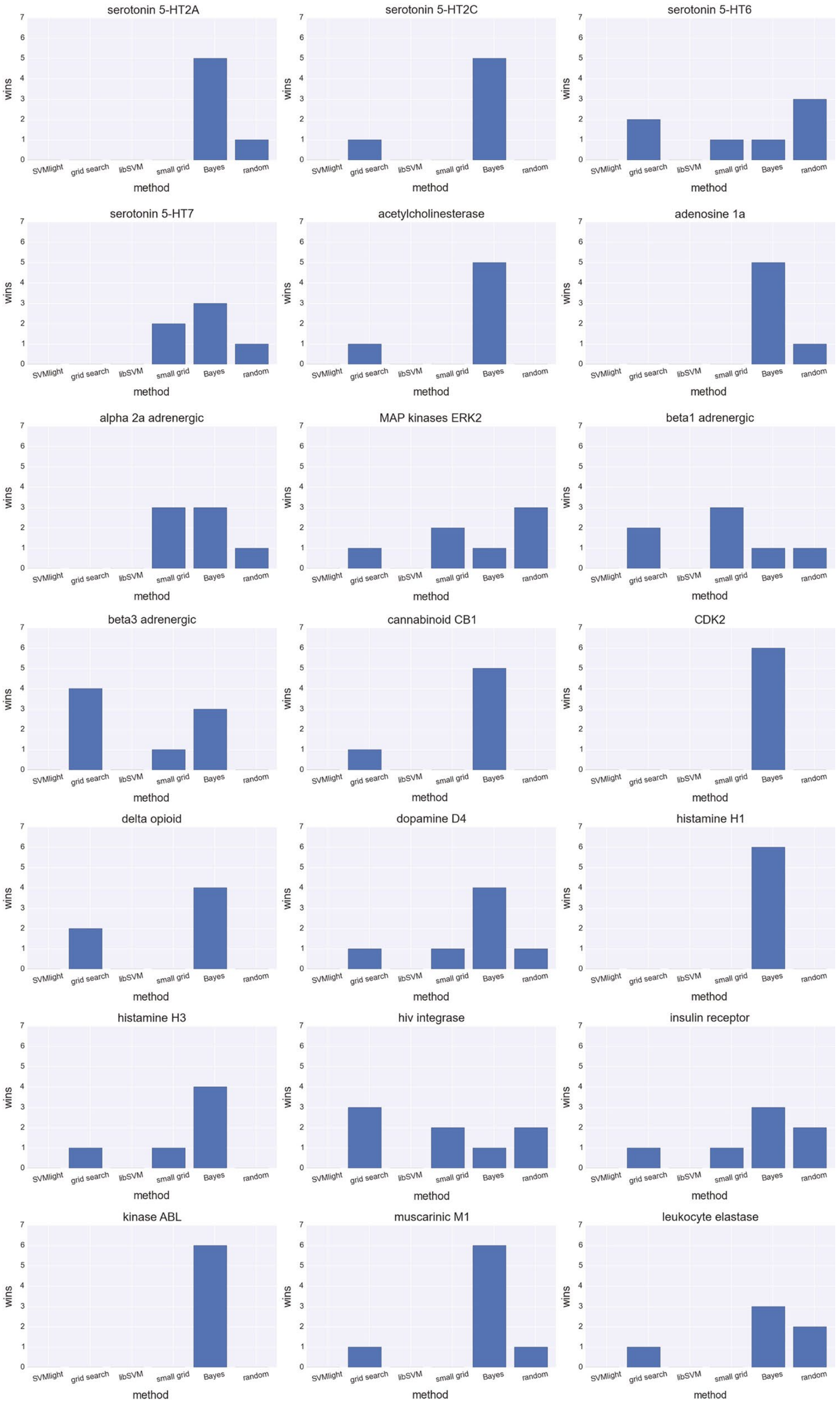

Fig. 3 Analysis of effectiveness of different SVM optimization strategies with respect to various targets expressed as the number of experiments in which a particular strategy provided the highest accuracy values for a given protein target.

Table 2 A comparison of the number of highest accuracies obtained with the Bayesian optimization and grid search

Comparison	Bayes	Grid search
Global	96	34
EstateFP	15	7
ExtFP	16	6
KlekFP	16	5
MACCSFP	18	4
PubchemFP	16	6
SubFP	15	6
5-HT_{2A}	5	1
5-HT_{2C}	5	1
5-HT_6	4	3
5-HT_7	3	3
CDK2	6	0
M_1	6	1
ERK2	5	1
AChE	5	1
A_1	5	1
alpha2AR	5	1
beta1AR	3	3
beta3AR	3	4
CB1	5	1
DOR	4	2
D_4	5	1
H_1	6	0
H_3	5	1
HIVi	1	5
IR	5	1
ABL	6	0
HLE	4	3

When target-based comparisons were considered, Bayesian optimization outperformed the grid search approach for some targets in all cases (i.e., CDK2, H_1, ABL); for others, there was only 1 case when the grid search strategy won (i.e., 5-HT_{2A}, 5-HT_{2C}, M_1, ERK2, AChE, A_1, alpha2AR, CB1, D_4, H_3, IR), still others were draws (i.e., 5-HT_7, beta1AR), and in two cases the grid search provided top accuracies (beta3AR, HIVi).

Examination of optimization steps in time

A time course study of the accuracy values was also conducted. Figure 4 shows analyses for 5-HT_{2A} as an example; the results for the remaining targets are in the Additional files section (Additional file 1). We demonstrated not only that Bayesian optimization required the smallest number of iterations to achieve optimal performance (for all representations the number of iterations was less than 20), but also that in the majority of cases, SVM optimized using a Bayesian approach achieved better performance than all of the other optimization methods. SVMlight and libSVM were not iteratively optimized; therefore, the accuracy/number of iterations function is constant for these approaches. In general, the Bayesian and random search approach were optimized very quickly (in less than 20 iterations), whereas the grid search method required many more iterations before the SVM reached optimal performance: 57 iterations (the lowest number) were required for EstateFP, and 138 (the highest number) for MACCSFP). Figure 4 also shows the rate of the improvement of the accuracy after the application of particular optimization approach, which depended on the representation of the compounds—for EstateFP, it was improvement from 0.8 (grid search) up to 0.82 (Bayesian), but for libSVM and SVMlight the improvement was significantly higher; for these two strategies, the classification accuracy was equal to 0.68. A similar result was obtained for ExtFP—the rate of improvement for the Bayesian optimization strategy compared with the random search approach was approximately 0.02 (from 0.88 to 0.90), but it was higher for the other optimization methods: 0.03 for grid search, 0.05 for libSVM and 0.22 for SVMlight. The pattern was similar for KlekFP and MACCSFP, with differences occurring only in the performance of libSVM. However, for PubchemFP and SubFP, grid search optimization provided the same predictive power for SVM as Bayesian optimization; for the SubFP, there was a selected range of iterations (117–142) when grid search provided slightly better SVM performance (by about 2%) in comparison to Bayesian approach.

In order to provide the comprehensive and global analysis of the changes in accuracy with an increasing number of iterations, the areas under curves (AUC) presented in Fig. 4 (and other curves that are placed in the Additional files section) were calculated. Example analysis for selected target/fingerprint pair (5-HT_{2A}, ExtFP) is presented in Table 3; the remaining analyses are in the Additional files section (Additional file 2). The global average AUC, the average AUC for particular fingerprints and targets are presented in Tables 4 and 5. The Tables also include final (for the selected target/fingerprint) and averaged (for the rest of the cases) final accuracy values obtained for a given strategy; the highest AUC/accuracy values for the particular case considered are marked with an asterisk sign. In general, the AUC of a curve indicates the strength of the trained model at any randomly chosen iteration. In other words, the AUC measures how quickly a given strategy converges to a strong model.

The analysis of the results obtained for the example target/fingerprint pair (5-HT_{2A}, ExtFP; Table 3) shows that both the highest AUC and final optimal accuracy values were obtained with the Bayesian strategy for SVM

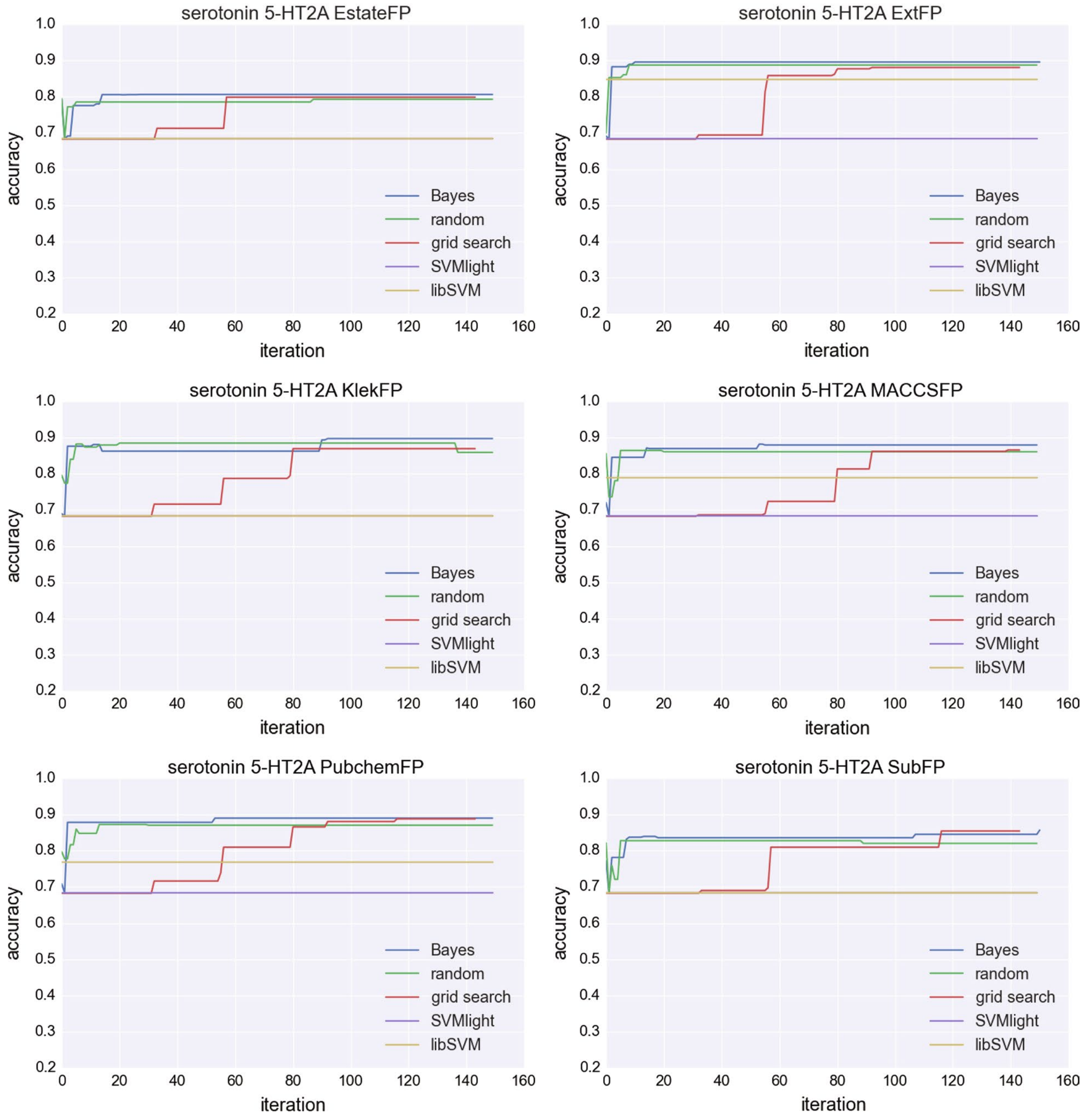

Fig. 4 Analysis of the changes in accuracy during the SVM optimization procedure for the subsequent optimization steps.

optimization. A similar observation was made for the global and fingerprint-based analysis; Bayesian optimization provided the best average AUC and average optimal accuracy for all fingerprints, as well as the global average value of this parameter. Interestingly, although grid search was the second-place method for optimal accuracy, it was actually the random search that outperformed this method in terms of AUC, which could be explained from an analysis of the respective curves. Although the grid search method provided higher final accuracy values, these occurred relatively 'late' (after a series of iterations), high accuracies were obtained almost immediately for random search (Figs. 4, 5). Similarly, the average AUC and optimal accuracy values calculated for various targets were highest for Bayesian optimization in the great majority of cases. HIVi and ERK2 were the only targets for which the averaged AUC obtained with the Bayesian optimization strategy was outperformed

Table 3 The AUC values obtained in 5-HT$_{2A}$, ExtFP for curves illustrating changes in the accuracy in time and final optimal accuracy values obtained

optimization method	AUC	Final accuracy
Bayes	0.892*	0.896*
Random	0.885	0.887
Grid search	0.802	0.881
SVMlight	0.683	0.683
libSVM	0.847	0.847

The highest values obtained among all strategies tested are marked with an asterisk sign

Table 4 The average AUC values–global, obtained for a particular fingerprint and particular target

Fingerprint/ target	Bayes	Random	Grid search	SVMlight	libSVM
global	0.883*	0.870	0.799	0.676	0.792
EstateFP	0.847*	0.829	0.774	0.690	0.763
ExtFP	0.902*	0.891	0.806	0.669	0.874
KlekFP	0.899*	0.889	0.812	0.669	0.730
MACCSFP	0.890*	0.876	0.798	0.683	0.828
PubchemFP	0.898*	0.885	0.816	0.669	0.808
SubFP	0.864*	0.854	0.787	0.677	0.749
5-HT$_{2A}$	0.860*	0.850	0.780	0.683	0.743
5-HT$_{2C}$	0.848*	0.821	0.702	0.568	0.717
5-HT$_6$	0.913*	0.910	0.886	0.814	0.862
5-HT$_7$	0.830*	0.816	0.748	0.675	0.714
CDK2	0.876*	0.875	0.796	0.664	0.768
M$_1$	0.850*	0.843	0.778	0.557	0.748
ERK2	0.958	0.961*	0.949	0.931	0.942
AChE	0.884*	0.854	0.788	0.611	0.764
A$_1$	0.843*	0.835	0.764	0.564	0.720
alpha2AR	0.875*	0.874	0.773	0.563	0.725
beta1AR	0.910*	0.864	0.798	0.710	0.828
beta3AR	0.874*	0.823	0.826	0.545	0.722
CB1	0.874*	0.854	0.782	0.622	0.793
DOR	0.888*	0.880	0.734	0.599	0.814
D$_4$	0.841*	0.837	0.759	0.698	0.745
H$_1$	0.898*	0.880	0.638	0.548	0.801
H$_3$	0.937*	0.926	0.906	0.897	0.905
HIVi	0.939	0.945*	0.934	0.901	0.911
IR	0.936*	0.936*	0.925	0.886	0.897
ABL	0.850*	0.831	0.748	0.587	0.733
HLE	0.867*	0.865	0.763	0.578	0.779

The highest values obtained among all strategies tested are marked with an asterisk sign

Table 5 The average final accuracy values—global, obtained for a particular fingerprint and particular target

fingerprint/ target	Bayes	Random	Grid search	SVMlight	libSVM
Global	0.889*	0.873	0.876	0.676	0.792
EstateFP	0.852*	0.832	0.833	0.690	0.763
ExtFP	0.907*	0.896	0.892	0.669	0.874
KlekFP	0.907*	0.890	0.891	0.669	0.730
MACCSFP	0.898*	0.878	0.880	0.683	0.828
PubchemFP	0.901*	0.886	0.894	0.669	0.808
SubFP	0.869*	0.856	0.864	0.677	0.749
5-HT$_{2A}$	0.871*	0.848	0.860	0.683	0.743
5-HT$_{2C}$	0.855*	0.825	0.772	0.568	0.717
5-HT$_6$	0.916*	0.915	0.933	0.814	0.862
5-HT$_7$	0.833*	0.819	0.819	0.675	0.714
CDK2	0.885*	0.881	0.870	0.664	0.768
M$_1$	0.858	0.846	0.897*	0.557	0.748
ERK2	0.959	0.961*	0.961*	0.931	0.942
AChE	0.889*	0.857	0.872	0.611	0.764
A$_1$	0.856	0.838	0.882*	0.564	0.720
alpha2AR	0.880*	0.873	0.872	0.563	0.725
beta1AR	0.914*	0.870	0.864	0.710	0.828
beta3AR	0.879	0.825	0.972*	0.545	0.722
CB1	0.881*	0.857	0.868	0.622	0.793
DOR	0.897*	0.884	0.872	0.599	0.814
D$_4$	0.849*	0.838	0.837	0.698	0.745
H$_1$	0.904*	0.879	0.691	0.548	0.801
H$_3$	0.938*	0.926	0.919	0.897	0.905
HIVi	0.938	0.946	0.967*	0.901	0.911
IR	0.939	0.937	0.956*	0.886	0.897
ABL	0.857*	0.836	0.840	0.587	0.733
HLE	0.867*	0.871	0.864	0.578	0.779

The highest values obtained among all strategies tested are marked with an asterisk sign

by other optimization methods. On the other hand, the group of targets for which the average optimal accuracy values were the highest for methods other than Bayesian optimization was a bit more extensive (i.e., M$_1$, ERK2, A$_1$, beta3AR, HIVi, IR). However, for most of these targets, the difference between the best average accuracy and that obtained with Bayesian optimization was approximately 3% (however, for example for beta3AR this difference approached to 10%, from 0.879 to 0.972). On the other hand, an improvement of several percentage points was also observed when the average AUC and optimal accuracy obtained with the Bayesian strategy were compared with the strategy that provided the 'second-best' accuracy value in the ranking.

The number of iterations required to achieve optimal SVM performance was also analyzed in detail (Fig. 5; Additional file 3). The most striking observation was that all curves corresponding to the Bayesian optimization results were both shifted towards higher accuracy values and were much 'shorter', meaning that a significantly lower

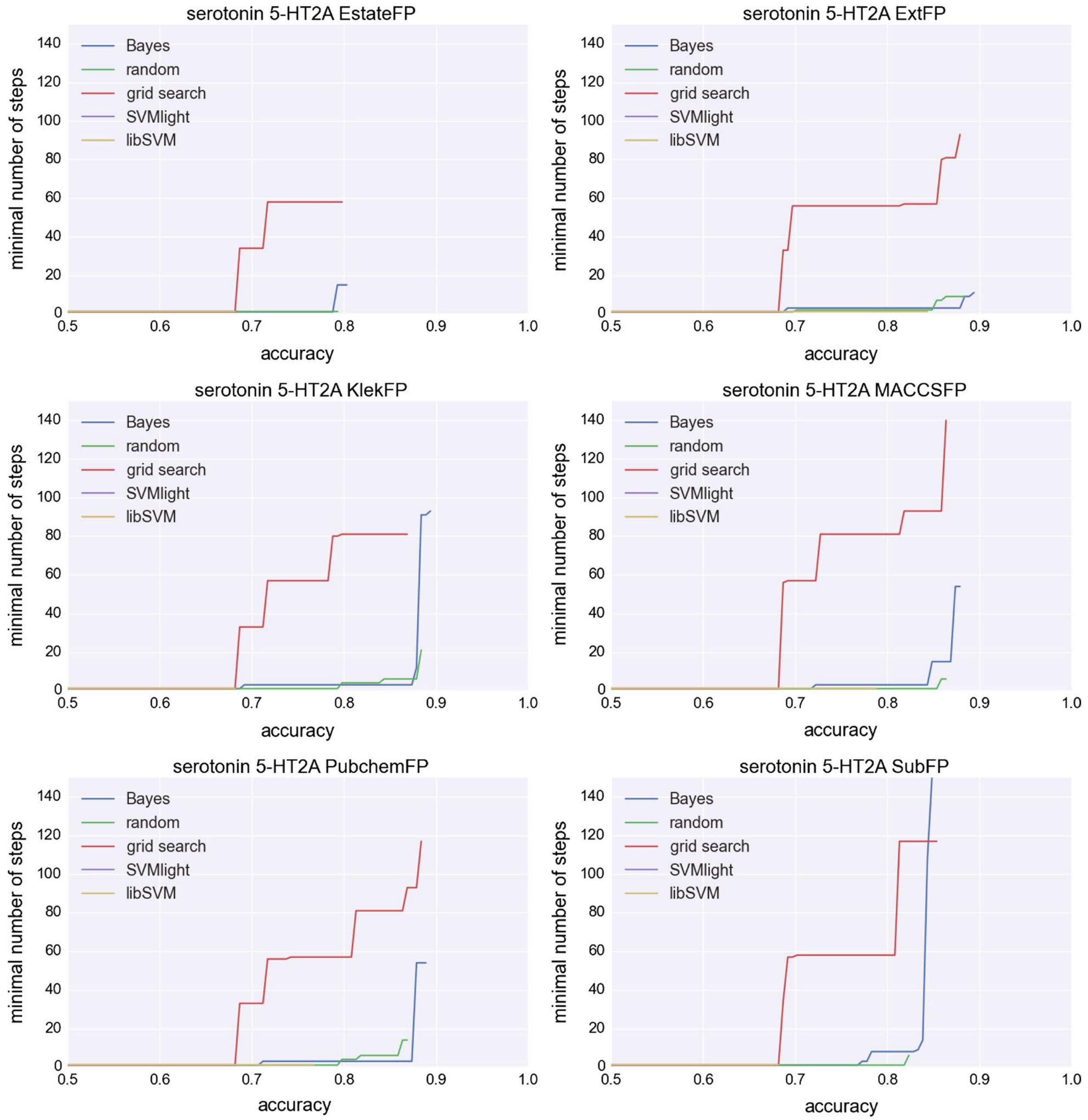

Fig. 5 Analysis of the number of iterations of the optimization procedure required to achieve the highest accuracy. The figure presents the number of iterations required for a particular optimization strategy to achieve optimal performance for the predictive model.

number of iterations was necessary in total to reach optimal SVM performance. Two relevant points arise from a comparison of Bayesian optimization with the grid search method (which sometimes outperformed Bayesian optimization): obtaining optimal accuracy with the grid search method required many more calculations, and even when grid search yielded higher accuracy values than Bayesian optimization, the difference between the two was approximately 1–2%. This result indicates that even when Bayesian optimization 'lost', the results provided by this strategy were still very good and taking into account the calculation speed, it can be successfully applied also in experiments for which it was not indicated to be the best approach. A very interesting observation arising from Fig. 5 is that random search reached the optimal classification effectiveness (as measured by accuracy) in the least number of iterations, below 10 in the majority of cases. EstateFP, ExtFP, MACCSFP and PubchemFP, showed similar tendency with respect to the comparison of Bayesian optimization and the grid search strategy; for an initial number of iterations (40), the accuracy values obtained with the grid search were approximately 20% lower than those obtained with the Bayesian approach. However, as the number of iterations for grid search increased, the accuracy values were also higher, and when the number of iterations reached approximately 100, the grid search results were similar to those obtained with Bayesian optimization. On the other hand, for both KlekFP and SubFP, the initial observations were the same; for a lower number of iterations, Bayesian optimization led to significantly higher accuracy values than the grid search approach, and for a higher number of iterations (over 80 for KlekFP and over 115 for SubFP), grid search provided accuracy values at a similar level to the values obtained with the Bayesian strategy. However, increasing the number of iterations for Bayesian optimization from approximately 10 to 90 for KlekFP and 150 for SubFP did not lead to a significant increase in the accuracy (an almost vertical line corresponding to these numbers of iterations), which was already very high (over 0.85 for KlekFP and over 0.8 for SubFP). Further optimization led to further improvement in accuracy of approximately 2–3%.

The results were also analyzed regarding the changes in the accuracy when additional steps were applied. A panel of example results is shown in Fig. 6 for the cannabinoid CB1/SubFP combination (the remaining targets are in Additional file 4). The black dots show the set of parameters tested in the particular approach, and the black squares represent the set of parameters selected as optimal. This chart shows the advantage of Bayesian optimization in terms of the way of work, and the sequence of selected parameters. The set of tested parameters is fixed for grid search optimization, whereas in case of random search, it is based on the random selection. On the other hand, the selection of parameters for Bayesian optimization is more directed, which also affects the effectiveness of the classification. For grid search, only a small fraction of the parameters tested provided satisfactory predictive power of the model (only approximately 35% of the predictions resulted in an accuracy exceeding 0.7). Surprisingly, a relatively high classification efficiency was obtained with the use of the random search approach—60% of the sets of parameters tested provided predictions with an accuracy over 0.7. However, investigation of the Bayesian optimization approach to parameter selection revealed that the choice of parameters tested was *justified*, and hence, the results obtained with their use were significantly better than those obtained with the other approaches—75% predictions with accuracy over 0.7.

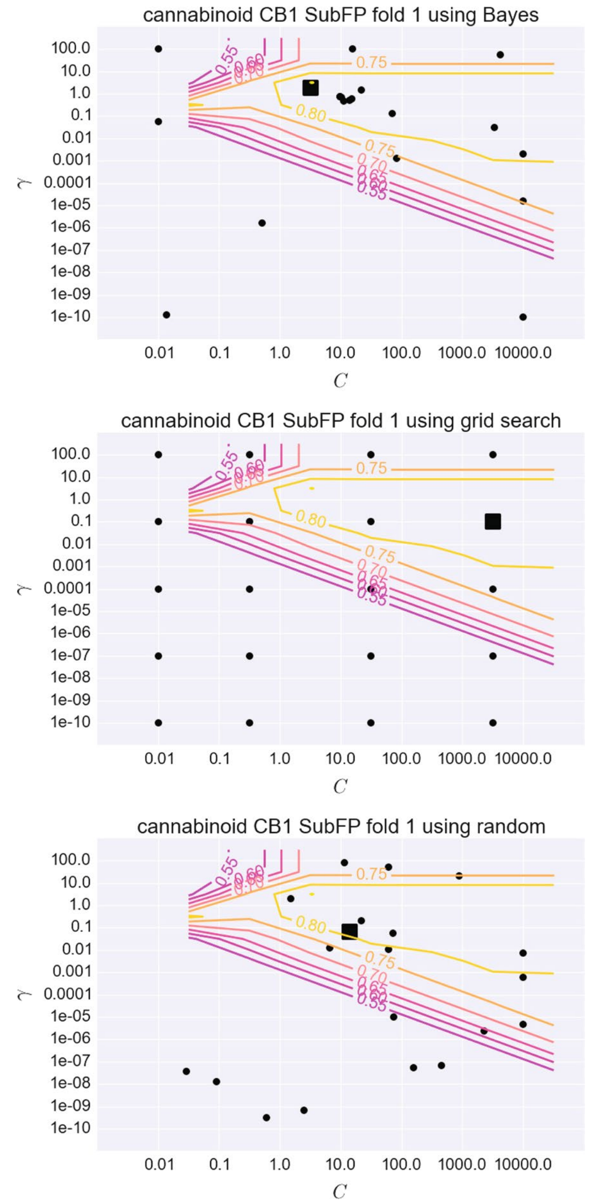

Fig. 6 Analysis of the changes in accuracy for different steps during the SVM optimization procedure.

We conclude that there are three SVM hyperparameters selection approaches worth using for activity prediction for compounds:

- libSVM heuristic (when only one set of hyperparameters is needed),
- random search (when we need a strong model quickly, using less than a few dozen iterations),
- a Bayesian approach (when we want the strongest model and can wait a bit longer).

The SVMlight heuristic as well as the traditional grid search approach have definitely been shown to be significantly worse in terms of the resulting model accuracy as well as time needed to construct such model.

Experimental

Several compounds datasets were prepared and their proper description using various fingerprints was provided for the planned experiments. The ChEMBL database constituted a source of active and inactive compounds with experimentally verified activity towards selected targets. The following proteins were considered in the study: serotonin receptors 5-HT_{2A} [11], 5-HT_{2C} [12], 5-HT_6 [13], and 5-HT_7 [14], cyclin dependent kinase 2 (CDK2) [15], muscarinic receptor M_1 [16], MAP kinase ERK2 [17], acetylcholinesterase (AChE) [18], adenosine receptor A_1 [19], alpha-2A adrenergic receptor [20], beta-1 adrenergic receptor (beta1AR) [21], beta-3 adrenergic receptor (beta3AR) [21], cannabinoid CB1 receptor [22], delta opioid receptor (DOR) [23], dopamine receptor D_4 [24], histamine receptor H_1 [25], histamine receptor H_3 [26], HIV integrase (HIVi) [27], insulin receptor (IR) [28], tyrosine kinase ABL [29], and human leukocyte elastase (HLE) [30]. Only molecules whose activities were quantified in K_i or IC_{50} and that were tested in assays on human, rat-cloned or native receptors were taken into account. The compounds were considered active when the median value of all K_i values provided for a particular instance was lower than 100 nM, and inactive when the median K_i value was greater than 1000 nM. The number of compounds from each group for the selected targets is shown in Table 6. The following fingerprints were used for compounds representation: E-state Fingerprint (EstateFP) [31], Extended Fingerprint (ExtFP) [32], Klekota and Roth Fingerprint (KlekFP) [33], MACCS Fingerprints (MACCSFP) [34], Pubchem Fingerprint (PubchemFP), and Substructure Fingerprint (SubFP), generated with the use of the PaDEL-Descriptor [35]. A brief characterization of the fingerprints is provided in Table 7).

The following SVM strategies were used:

- default SVM parameters used in the WEKA package ($C = 1, \gamma = \frac{1}{d}$)—libSVM.
- default SVM parameters from the SVMlight library ($C = \frac{1}{\text{mean}\|x_i\|^2}$, $\gamma = \frac{1}{d}$).
- grid search optimization of SVM parameters—$\log_{10}(C) \in [-2, 5]$, $\log_{10}(\gamma) \in [-10, 3]$.
- SVM parameters optimization in the truncated cross-validation mode ('small grid'-cv).
- SVM parameters optimization in the random cross-validation mode—number of iterations: up to 150.
- Bayesian optimization using BayesOpt [36]—number of iterations: up to 150.

Table 6 The number of active and inactive compounds in the dataset

Protein	Actives	Inactives
5-HT_{2A}	1836	852
5-HT_{2C}	1211	927
5-HT_6	1491	342
5-HT_7	705	340
CDK2	741	1462
M_1	760	939
ERK2	72	958
AChE	1147	1804
A_1	1789	2286
alpha2AR	364	283
beta1AR	195	477
beta3AR	111	133
CB1	1964	1714
DOR	2535	1992
D_4	1034	449
H_1	636	546
H_3	2706	313
HIVi	102	915
IR	147	1139
ABL	409	582
HLE	820	610

The range of C and γ values tested was as follows: $\log_{10}(C) \in [-2, 5]$, $\log_{10}(\gamma) \in [-10, 3]$ (the result of preliminary grid search experiments). The number of iterations in which random search, 'small grid'-cv and Bayesian optimization experiments were performed fell within the following set: 20, 30, 50, 75, 100, 150.

The predictive power of SVM for different optimization strategies applied was measured by the accuracy:

$$\text{Accuracy}(TP, FP, TN, FN) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

with TP being the number of true positives (correctly classified actives), TN—the number of true negatives (correctly classified inactives), FP—the number of false positives (inactives wrongly classified as active), and FN—the number of false negatives (actives wrongly classified as inactive).

Conclusions

The paper presents strengths of Bayesian optimization applied for fitting SVM hyperparameters in cheminformatics tasks. Because the importance and necessity of the SVM optimization procedure is undeniable, various approaches to this task have neen developed so far. However, the most popular approaches to SVM optimization are not always very effective, in terms of both the

Table 7 Fingerprints used for compounds representation

Fingerprint	Abbreviation	Length	Short description
E-State fingerprint	EStateFP	79	Computes electrotopological state (E-state) index for each atom, describing its electronic state with consideration of the influence of other atoms in particular structure
Extended fingerprint	ExtFP	1024	A hashed fingerprint with each atom in the given structure being a starting point of a string of a length not exceeding six atoms. A hash code is produced for every path of such type and in turn it constitutes the basis of a bit string representing the whole structure
Klekota and Roth fingerprint	KlekFP	4860	Fingerprint analyzing the occurrence of particular chemical substructures in the given compound. Developed by Klekota and Roth
MACCS fingerprint	MACCSFP	166	Fingerprint using the MACCS keys in its bits definition
Pubchem fingerprint	PubchemFP	881	Substructure fingerprint with bits divided into several sections: hierarchic element counts, rings, simple atom pairs, simple atom nearest neighbours, detailed atom neighbourhoods, simple SMART patterns, complex SMART patterns
Substructure fingerprint	SubFP	308	Substructure fingerprint based on the SMART patterns developed by Christian Laggner

predictive power of the models obtained and the computational requirements. This study demonstrated that Bayesian optimization not only provides better classification accuracy than the other optimization approaches tested but is also much faster and directed—in the majority of cases, the number of iterations required to achieve optimal performance was the lowest out of the all methods tested, and the set of parameters tested provided the best predictions on average. Interestingly, if good classification results are desired to be obtained quickly (using a low number of iterations and without complex algorithms), the random search method in which hyperparameters are randomly selected from a predefined range) leads to very good performance of the SVM for predicting the activity of compounds and can thus be used when Bayesian optimization approach is not feasible.

Consequently, we can formulate the following rule of thumb for tuning SVM's hyperparameters for the classification of bioactive compounds:

1. If you have no resources for performing hyperparameters optimization, use $C = 1, \gamma = \frac{1}{d}$ (as defined in libSVM).
2. If you have limited resources (up to 20 learning procedures) or limited access to complex optimization software, use a random search for C and γ with distribution defined in the "Methods" section.
3. If you have resources for 20 or more training runs and access to Bayesian optimization software[a], use a Bayesian optimization of C, γ.

In general, there is no scenario in which one should use a grid search approach (it is always preferable to use random search or a Bayesian method) or SVMlight heuristics (it is always better to use libSVM) in the tasks connected with the assessment of compounds bioactivity.

Methods

The objective of the iterative global optimization of a function $f : \mathcal{L} \to \mathbb{R}$ is to find the sequence of points

$$\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_n \in \mathcal{L},$$

that converges to the optimal $\hat{\boldsymbol{\lambda}}$, $f(\hat{\boldsymbol{\lambda}}) = \sup_{\boldsymbol{\lambda} \in \mathcal{L}} f(\boldsymbol{\lambda})$. A good algorithm should find a solution at least over some family of functions $\mathcal{F}$, not necessarily containing f.

The above-mentioned issue can be viewed as a sequential decision making problem [37] in which at time step i a decision based on all previous points $\alpha_i(\boldsymbol{\lambda}_{1:i-1}, \bar{f}_{1:i-1})$, where $\bar{f}_i = f(x_i) + \varepsilon_i$ is made. In other words, we have access to approximations of f values from previous steps. For simplicity, assume that $\varepsilon_i = 0$ (f is deterministic); however, in general, all methods considered can be used in a stochastic scenario (for example, when randomized cross-validation is used as underlying method for f evaluation).

The goal is to find α which minimizes $\delta_n(\alpha) = f(\hat{\boldsymbol{\lambda}}) - f(\boldsymbol{\lambda}_n)$, meaning that we are interested in

$$\arg\min_{\alpha} \delta_n(\alpha) = \arg\min_{\alpha} f(\hat{\boldsymbol{\lambda}}) - f(\boldsymbol{\lambda}_n), \tag{1}$$

which could be efficiently solved if f is known.

Approximation of generalization capabilities

In general, we are interested in how well our predictive model behaves on a naive test set. In other words, we are assuming that our data are a finite iid (independent and identically distributed) sample from some underlying joint distribution over samples (compounds) and their binary labels (biological activity) μ:

$$\mathrm{T} = \{(x_i, y_i)_{i=1}^N\} \sim \mu(\mathcal{X} \times \{-1, +1\}),$$

where $\mathcal{X}$ represents a feature space of compounds under investigation. We want to maximize the *expected accuracy* over all possible compounds from μ, in other words

$$\begin{aligned} f(\boldsymbol{\lambda}) &= \mathbb{E}_{(x_i,y_i)\sim\mu}[\mathrm{svm}(x_i|\mathrm{T},\boldsymbol{\lambda}) = y_i] \\ &= \int [\mathrm{svm}(x_i|\mathrm{T},\boldsymbol{\lambda}) = y_i] d\mu, \end{aligned}$$

where [p] is a characteristic function returning 1 if and only if p is true, and $\mathrm{svm}(x_i|\mathrm{T},\boldsymbol{\lambda})$ is a prediction of x_i's label by SVM trained with hyperparameters $\boldsymbol{\lambda}$ on training set T.

Clearly, we cannot integrate over an unknown probability distribution, but we can approximate this value using internal cross-validation. In other words, we are using a stochastic approximation

$$\bar{f}(\boldsymbol{\lambda}) = \mathrm{CV}_{\mathrm{T}}(\mathrm{svm}(X_{\mathrm{test}}|\mathrm{T}_{\mathrm{train}},\boldsymbol{\lambda}), Y_{\mathrm{test}}),$$

where $\mathrm{CV}_{\mathrm{T}}(p,y)$ is the mean accuracy of the model of predictions p as compared to the true labels y over splits of set T into $\mathrm{T}_{\mathrm{train}}$ and $\mathrm{T}_{\mathrm{test}}$ (composed of X_{test} data and corresponding labels Y_{test}). Thus we can assume [38] that

$$\bar{f}(\boldsymbol{\lambda}) = f(\boldsymbol{\lambda}) + \varepsilon,$$

where ε is a random noise variable (resulting from the approximating error and stochastic nature of cross validation).

Random optimization

First, let us define a random optimization technique as a strategy $\alpha^{\mathcal{R}}(\boldsymbol{\lambda}_{1:i-1}, \bar{f}_{1:i-1}) = \alpha_i^{\mathcal{R}} = \boldsymbol{\lambda}_i \sim \boldsymbol{P}(\mathcal{L})$, for some probability distribution over the hyperparameters $\boldsymbol{P}(\mathcal{L})$. In other words, in each iteration, we sample from $\boldsymbol{P}(\mathcal{L})$, ignoring all previous samples and their results. Finally, we return the maximum of the values obtained.

It is easily seen that a random search, under the assumption that $\forall_{\lambda\in\mathcal{L}} \boldsymbol{P}(\alpha_i^{\mathcal{R}} = \boldsymbol{\lambda}) > 0$, has a property described in (1). A random search will converge to the optimum [39], if only each set of parameters is possible to generate when taking new sample from our decision making process. In practise, it is only necessary that $\boldsymbol{P}(f(\alpha_i^{\mathcal{R}}) = f(\hat{\boldsymbol{\lambda}})) > 0$. Similarly, if one uses a grid search approach that discretizes $\mathcal{L}$, then given enough iterations and the assumption that f is continuous, one will converge to the optimal solution. It is important to note that the speed of such a convergence can be extremely low.

The only thing missing is the selection of $\boldsymbol{P}(\mathcal{L})$. According to many empirical studies showing that meaningful changes in the SVM results as the function of its hyperparameters can be expressed in log-scale of these parameters we use

$$\boldsymbol{P}(\boldsymbol{\lambda} = (C,\gamma)) = \frac{\log_{10} C - \log_{10} C_{\min}}{\log_{10} C_{\max} - \log_{10} C_{\min}} \cdot \frac{\log_{10} \gamma - \log_{10} \gamma_{\min}}{\log_{10} \gamma_{\max} - \log_{10} \gamma_{\min}}$$

where we are interested in $\mathcal{L} = [C_{\min}, C_{\max}] \times [\gamma_{\min}, \gamma_{\max}]$. In other words, we are using a log-uniform distributions independently over C and γ.

Grid search

For grid search optimization we select $\boldsymbol{\lambda}$ in a similar manner to the random search approach, uniformly in the log-scale of the parameters, and given M_p choices of parameter p

$$\begin{aligned} \boldsymbol{\lambda}_{ij} &= (C_i, \gamma_j) \\ &= \left(10^{\log_{10} C_{\min} + (i-1)\frac{\log_{10} C_{\max} - \log_{10} C_{\min}}{M_C - 1}}, \right. \\ &\quad \left. 10^{\log_{10} \gamma_{\min} + (j-1)\frac{\log_{10} \gamma_{\max} - \log_{10} \gamma_{\min}}{M_\gamma - 1}} \right). \end{aligned}$$

We put the linear order of $\boldsymbol{\lambda}_{ij}$ by raveling the resulting matrix by column, which is the most common practice in most ML libraries. It is worth noting that one could achieve better scores by alternating this ordering to any random permutation; however, in practice, such alternation is rarely performed.

Bayesian optimization

If the exact form of f is known (for example, if f is convex and its derivative is known), then the optimization procedure would be much simpler. Unfortunately, f is a black-box function wih a very complex structure, expensive even to evaluate. However, some simplifying assumptions for f might make a problem solvable. Assume that f can be represented as a sample from a probability distribution over a family of functions $f \sim \boldsymbol{P}(f), f \in \mathcal{F}$.

We can now express the expectation over the loss function δ_n:

$$\arg\min_{\alpha} \mathbb{E}_{\boldsymbol{P}(f)}[\delta_n(\alpha)] = \arg\min_{\alpha} \int_{\mathcal{F}} \delta_n(\alpha) d\boldsymbol{P}(f).$$

Given that in step n the values of $\boldsymbol{\lambda}_i, \bar{f}_i$ for $i = 1, \ldots, n-1$ are already known and using the Bayes rule, we can write:

$$\boldsymbol{P}(f|x_{1:i}, \bar{f}_{1:i}) = \frac{\boldsymbol{P}(x_i, \bar{f}_i|f)\boldsymbol{P}(f|x_{1:i-1}, \bar{f}_{1:i-1})}{\boldsymbol{P}(x_i, \bar{f}_i)}, \quad \forall i = 1, \ldots, n-1,$$

thus

$$\begin{aligned} &\arg\min_{\alpha} \mathbb{E}_{\boldsymbol{P}(f|x_{1:n-1}, \bar{f}_{1:n-1})}[\delta_n(\alpha)] \\ &\quad = \arg\min_{\alpha} \int_{\mathcal{F}} \delta_n(\alpha) d\boldsymbol{P}(f|x_{1:n-1}, \bar{f}_{1:n-1}). \end{aligned}$$

This is a very basic equation for general Bayesian optimization techniques. Given additional assumptions about the prior distribution of $\boldsymbol{P}(f)$, very efficient solutions for the entire process can be provided. In the case considered here, a very common approach exploiting features of the Gaussian processes is employed; thus, we assume that our target function (the generalization capabilities of the SVM with RBF kernel applied to the prediction of the activity of chemical compounds) f is a sample of the stochastic process. A crucial advantage of such a simplification is that we can easily manipulate the distribution over such functions: in particular, the posterior distribution is also a Gaussian process. Consequently, in each iteration a calculated posterior can be used as an informative prior for the next iteration, creating a relatively simple iterative procedure.

The only thing missing is the selection of the loss function, because $\delta_n(\alpha)$ defined above requires knowledge of the optimal solution. There are many surrogate functions (also called proxies) that might be of use. In our investigations we used one of the most well-known [7] and studied *expected improvement* [40, 41], which has the following closed form solution under the above assumptions:

$$\alpha_{\mathrm{EI}}(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{1:n-1},\bar{f}_{1:n-1}) = \sigma(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{1:n-1},\bar{f}_{1:n-1})(\gamma(\boldsymbol{\lambda})\phi(\gamma(\boldsymbol{\lambda})) + \mathcal{N}(\gamma(\boldsymbol{\lambda}))),$$

$$\gamma(\boldsymbol{\lambda}) = \frac{\mu(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{1:n-1},\bar{f}_{1:n-1}) - f(\boldsymbol{\lambda}_{\mathrm{best}})}{\sigma(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{1:n-1},\bar{f}_{1:n-1})}$$

where μ, σ^2 are Gaussian process mean and variance predictions, $\boldsymbol{\lambda}_{\mathrm{best}}$ is a current best solution $f(\boldsymbol{\lambda}_{\mathrm{best}}) = \max_{i=1,\ldots,n-1} f(\boldsymbol{\lambda}_i)$, ϕ is a cumulative density function of the standard normal distribution and $\mathcal{N}$ is a probability density function of the standard normal distribution. Thus, in each iteration we simply select the point that maximizes the above equation

$$\alpha_n(\boldsymbol{\lambda}_{1:n-1},\bar{f}_{1:n-1}) = \arg\max_{\boldsymbol{\lambda}} \alpha_{\mathrm{EI}}(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{1:n-1},\bar{f}_{1:n-1}).$$

Endnotes

[a] For example BayesOpt http://rmcantin.bitbucket.org/html/

Additional files

Additional file 1: Analysis of the time course of accuracy values during execution of the SVM optimization procedure. The file contains the analysis of the changes in accuracy values with different time for SVM optimizations strategies for all targets tested.

Additional file 2: AUC values obtained for all target/fingerprint pairs for curves illustrating changes in the accuracy with time and the optimal accuracy values obtained. The file contains the AUC and optimal accuracy values obtained in the experiments. The AUC values were calculated on the basis of curves illustrating changes in the accuracy values in time for different SVM optimization strategies.

Additional file 3: Analysis of the number of iterations of the optimization procedure required for reaching the highest accuracy for all tested targets. The file presents the number of iterations after which the optimal accuracy values were reached for all targets tested.

Additional file 4: Analysis of the changes of accuracy for different steps for all tested targets. The file presents the set of parameters tested in the subsequent iterations of the optimization procedure with the analysis of classification accuracy values that they were providing.

Author's contributions

WCz and SP performed the experiments. All the authors analyzed and discussed the results and wrote the manuscript. All authors read and approved the final version of the manuscript.

Author details

[1] Faculty of Mathematics and Computer Science, Jagiellonian University, 6 S. Lojasiewicza Street, 30-348 Krakow, Poland. [2] Department of Medicinal Chemistry, Institute of Pharmacology, Polish Academy of Sciences, 12 Smetna Street, 31-343 Krakow, Poland. [3] Faculty of Chemistry, Jagiellonian University, 3 Ingardena Street, 30-060 Krakow, Poland.

Acknowledgements

The study was partially supported by a Grant OPUS 2014/13/B/ST6/01792 financed by the Polish National Science Centre (http://www.ncn.gov.pl) and by the Statutory Funds of the Institute of Pharmacology Polish Academy of Sciences. SP and AJB participate in the European Cooperation in Science and Technology (COST) Action CM1207: GPCR-Ligand Interactions, Structures, and Transmembrane Signalling: an European Research Network (GLISTEN).

Compliance with ethical guidelines

Competing interests

The authors declare that they have no competing interests.

Received: 25 April 2015 Accepted: 6 July 2015
Published online: 14 August 2015

References

1. Hammann F, Gutmann H, Baumann U, Helma C, Drewe J (2009) Classification of cytochrome P 450 activities using machine learning methods. Mol Pharm 33(1):796–801
2. Smusz S, Czarnecki WM, Warszycki D, Bojarski AJ (2015) Exploiting uncertainty measures in compounds activity prediction using support vector machines. Bioorganic Med Chem Lett 25(1):100–105
3. Lee JH, Lee S, Choi S (2010) In silico classification of adenosine receptor antagonists using Laplacian-modified naïve Bayesian, support vector machine, and recursive partitioning. J Mol Graph Model 28(8):883–890
4. Wang M, Yang X-G, Xue Y (2008) Identifying hERG potassium channel inhibitors by machine learning methods. QSAR Comb Sci 27(8):1028–1035
5. Swersky K, Snoek J, Adams RP (2013) Multi-task bayesian optimization. In: Advances in neural information processing systems, vol 26. Lake Tahoe, pp 2004–2012
6. Snoek J, Rippel O, Swersky K, Kiros R, Satish N, Sundaram N et al (2015) Scalable bayesian optimization using deep neural networks. arXiv preprint aXiv:1502.05700
7. Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, vol 25. Lake Tahoe, pp 2951–2959
8. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(1):281–305
9. Eggensperger K, Feurer M, Hutter F, Bergstra J, Snoek J, Hoos H et al (2013) Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In: NIPS Workshop on Bayesian Optimization in Theory and Practice. Lake Tahoe, pp 1–5
10. Thornton C, Hutter F, Hoos HH, Leyton-Brown K (2013) Auto-weka: combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 847–855
11. Sencanski M, Sukalovic V, Shakib K, Soskic V, Dosen-Micovic L, Kostic-Rajacic S (2014) Molecular modelling of $5HT_{2A}$ receptor—arylpiperazine ligands interactions. Chem Biol Drug Design 83(4):462–471
12. Millan MJ (2005) Serotonin 5-HT2C receptors as a target for the treatment of depressive and anxious states: focus on novel therapeutic strategies. Therapie 60(5):441–460
13. Upton N, Chuang TT, Hunter AJ, Virley DJ (2008) 5-HT6 receptor antagonists as novel cognitive enhancing agents for Alzheimer's disease. Neurotherapeutics 5(3):458–469

14. Roberts AJ, Hedlund PB (2012) The 5-HT(7) receptor in learning and memory. Hippocampus 22(4):762–771
15. Chen H, Van Duyne R, Zhang N, Kashanchi F, Zeng C (2009) A novel binding pocket of cyclin-dependent kinase 2. Proteins 74(1):122–132
16. Leach K, Simms J, Sexton PM, Christopoulos A (2012) Structure-function studies of muscarinic acetylcholine receptors. In: Fryer AD, Christopoulos A, Nathanson NM (eds) Handbook of experimental pharmacology, vol 208. Springer, UK, pp 29–48
17. Zhang F, Strand A, Robbins D, Cobb MH, Goldsmith EJ (1994) Atomic structure of the MAP kinase ERK2 at 2.3 A resolution. Nature 367(6465):704–711
18. Soreq H, Seidman S (2001) Acetylcholinesterase–new roles for an old actor. Nat Rev Neurosci 2(4):294–302
19. Hocher B (2010) Adenosine A1 receptor antagonists in clinical research and development. Kidney Int 78(5):438–445
20. Hein L (2001) The alpha 2-adrenergic receptors: molecular structure and in vivo function. Z Kardiol 90(9):607–612
21. Wallukat G (2002) The beta-adrenergic receptors. Herz 27(7):683–690
22. Pertwee RG (1997) Pharmacology of cannabinoid CB1 and CB2 receptors. Pharmacol Ther 4(2):129–180
23. Quock RM (1999) The delta-opioid receptor: molecular pharmacology, signal transduction, and the determination of drug efficacy. Pharmacol Rev 51(3):503–532
24. Rondou P, Haegeman G, Van Craenenbroeck K (2010) The dopamine D4 receptor: biochemical and signalling properties. Cell Mol Life Sci 67(12):1971–1986
25. Thurmond RL, Gelfand EW, Dunford PJ (2008) The role of histamine h1 and h4 receptors in allergic inflammation: the search for new antihistamines. Nat Rev Drug Discov 7(1):41–53
26. Passani MB, Lin J-S, Hancock A, Crochet S, Blandina P (2004) The histamine H3 receptor as a novel therapeutic target for cognitive and sleep disorders. Trend Pharmacol Sci 25(12):618–625
27. Craigie R (2001) Hiv integrase, a brief overview from chemistry to therapeutics. J Biol Chem 276(26):23213–23216
28. Whitehead JP, Clark SF, Ursø B, James DE (2000) Signalling through the insulin receptor. Curr Opin Cell Biol 12(2):222–228
29. Lanier LM, Gertler FB (2000) From Abl to actin: Abl tyrosine kinase and associated proteins in growth cone motility. Curr Opin Neurobiol 10(1):80–87
30. Lee WL, Downey GP (2001) Leukocyte elastase: physiological functions and role in acute lung injury. Am J Respir Crit Care Med164(5):896–904
31. Hall LH, Kier LB (1995) Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information. J Chem Inform Model 35(6):1039–1045
32. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E (2003) The chemistry development kit (CDK): an open-source Java library for chemo- and bioinformatics. J Chem Inform Comp Sci 43(2):493–500
33. Klekota J, Roth FP (2008) Chemical substructures that enrich for biological activity. Bioinform Oxf Engl 24(21):2518–2525
34. Ewing T, Baber JC, Feher M (2006) Novel 2D fingerprints for ligand-based virtual screening. J Chem Inform Model 46(6):2423–2431
35. Yap CW (2011) Padel-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32(7):1466–1474
36. Martinez-Cantin R (2014) Bayesopt: a bayesian optimization library for nonlinear optimization, experimental design and bandits. arXiv preprint arXiv:1405.7430
37. Mockus J (1994) Application of bayesian approach to numerical methods of global and stochastic optimization. J Glob Optim 4(4):347–365
38. Bishop CM (2006) Pattern recognition and machine learning. Springer, NJ
39. Auger A, Doerr B (2011) Theory of randomized search heuristics: foundations and recent developments, vol 1. World Scientific, Series on Theoretical Computer Science
40. Mockus J, Tiesis V, Zilinskas A (1978) The application of bayesian methods for seeking the extremum. Towards Glob Optim 2(117–129):2
41. Schonlau M, Welch WJ, Jones DR (1998) Global versus local search in constrained optimization of computer models. Lect Notes Monogr Ser 34:11–25