J Comput Aided Mol Des (2015) 29:305–314
DOI 10.1007/s10822-015-9832-9

Introducing the 'active search' method for iterative virtual screening

Roman Garnett · Thomas Gärtner · Martin Vogt · Jürgen Bajorath

Received: 4 December 2014 / Accepted: 22 January 2015 / Published online: 1 February 2015

Abstract A method is introduced for sequential similarity searching for active compounds. Given a set of known actives and a screening database, a strategy is devised to optimally rank test compounds by observing the outcome of each iteration before selecting the next compound. This 'active search' approach is based upon Bayesian decision theory. A typical ranking procedure used in virtual compound screening corresponds to a myopic approximation to the optimal strategy. Exploratory active search represents a less-myopic approach and is shown to accurately identify a variety of active compounds in iterative virtual screening trials on 120 compound classes. Source code and data for the active search approach presented herein is made freely available.

Keywords Active search · Iterative virtual screening · Bayesian decision theory

R. Garnett · T. Gärtner
Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität Bonn, Römerstr. 164, 53117 Bonn, Germany

T. Gärtner
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany

M. Vogt · J. Bajorath (✉)
Program Unit Chemical Biology and Medicinal Chemistry, Department of Life Science Informatics, B-IT, LIMES, Rheinische Friedrich-Wilhelms-Universität Bonn, Dahlmannstr. 2, 53113 Bonn, Germany
e-mail: bajorath@bit.uni-bonn.de

Introduction

The development of methods for ligand-basedvirtual screening (LBVS) is a central task in chemoinformatics [1]. Regardless of the specifics of LBVS methods, they are generally based on computational assessment of molecular similarity used as an indicator of activity similarity [2]. Therefore, known active reference molecules are compared to test compounds and their similarity is quantified with respect to chosen molecular representations (such as fingerprints or numerical descriptors) [1, 2]. One of the most widely used LBVS approaches is similarity searching [2, 3], which produces rankings of database compounds in the order of decreasing similarity to reference molecule(s). For similarity searching, molecular fingerprints are the most widely used descriptors [3]. In analogy to high-throughput screening (HTS), LBVS can also be carried out in an iterative manner [4, 5]. Following iterative or sequential screening schemes, a small subset of a source database is screened per iteration and newly identified active compounds provide additional input information for the subsequent round [4, 5]. Hence, the principal idea underlying iterative screening is to limit the number of compounds that need to be evaluated to identify a significant proportion of available hits [4]. For LBVS this means that small database selection sets, however computed iteratively, must be enriched with compounds having a high probability of activity [5]. However, this represents a non-trivial task because only a small number of database compounds can be expected to be specifically active against a given target. Herein, we introduce a new approach for iterative LBVS by adapting the principle of an 'active search' from computer science that was considered in an abstract context [6]. The active search approach originally addressed the problem of selecting preferred data points for a given analysis from

large amounts of available data [6] from the point of view of Bayesian decision theory [7]. Based on the choice of a utility function for a given data selection task, a hierarchy of computationally increasingly complex selection strategies was derived to approximate the optimal strategy (which typically has too high computational complexity to be feasible) [6].

In the following, the active search approach is adapted for LBVS and compared to standard k-nearest neighbor (k-NN) similarity searching. In active search, given an initially available small set of compounds with known activity, a single compound is selected for evaluation. After observing the outcome (active vs. inactive), the next compound is selected, and this 'lookahead' procedure is continued to identify new hits. One- or two-step lookahead iterations are carried out and compared. The goal of the proposed method is to identify active compounds as early as possible by taking information from previous iterations into consideration. It is shown that one-step-lookahead search corresponds to a myopic approximation to the optimal selection strategy, similar to 1-NN similarity searching, but that two-step-lookahead represents a less-myopic approximation. In many LBVS trials, two-step-lookahead active search identified more active compounds in fewer iterations than iterative 1-NN similarity search or the one-step-lookahead procedure

Methodology

Active search

To adapt the active search for iterative virtual screening, we closely follow the theory originally introduced by Garnett et al. [6]. Suppose we have a finite set of elements (i.e., compounds) $\mathscr{X} \stackrel{\text{def}}{=} \{x_i\}$ and an identified subset $\mathscr{R} \subset \mathscr{X}$, the members of which we will call *active*. Furthermore, suppose it is unknown which members of $\mathscr{X}$ belong to $\mathscr{R}$ a priori, but it is possible to successively request binary observations $y \stackrel{\text{def}}{=} \mathbf{1}_{\mathscr{R}}(x)$, for an unlabeled element $x \in \mathscr{X}$, where $\mathbf{1}_{\mathscr{R}}$ is the indicator function for set $\mathscr{R}$. The goal is to select a sequence of queries to maximize a given utility function. For the active search problem, we define the utility of a set of observations $\mathscr{D} \stackrel{\text{def}}{=} \{(x_i, y_i)\}$ to be the number of actives found:

$$u(\mathscr{D}) \stackrel{\text{def}}{=} \sum y_i.$$

This simple expression captures the essence of the search problem as defined above.

The search methodology is based upon Bayesian decision theory. This will require selecting a classification model that provides the posterior probability of a point x belonging to $\mathscr{R}$ conditioned on previously observed data $\mathscr{D}$,

$$\Pr(y = 1 \mid x, \mathscr{D}).$$

It is assumed that this model is specified a priori; the decision-theoretic analysis does not depend on the nature of the model.

Without loss of generality, it is assumed that at the onset only a fixed number of queries t will be allowed. The policy for deciding the locations of the queries requires successive calculation of the expected utility of each of the remaining unlabeled objects, then the label for the object that maximizes expected utility is determined. At time $i \leq t$, the label for the object

$$x_i^* \stackrel{\text{def}}{=} \arg\max_{x_i \in \mathscr{X} \setminus \mathscr{D}_{i-1}} \mathbb{E}\left[u(\mathscr{D}_t) \mid x_i, \mathscr{D}_{i-1}\right]$$

is observed. We begin by considering the simplified case when exactly one further query is permitted (one-step lookahead) and will then address the general case. Suppose that $t-1$ observations $\mathscr{D}_{t-1}$ have already been made. To select the final observation, we calculate the expected utility of a candidate object x_t, integrating out the unknown value of y_t.

For active search, the expected utility is

$$\begin{aligned}\mathbb{E}\left[u(\mathscr{D}_t) \mid x_t, \mathscr{D}_{t-1}\right] &= \textstyle\sum_y u(\mathscr{D}_t) \Pr(y_t = y \mid x_t, \mathscr{D}_{t-1}) \\ &= u(\mathscr{D}_{t-1}) + \Pr(y_t = 1 \mid x_t, \mathscr{D}_{t-1}).\end{aligned}$$

Because $u(\mathscr{D}_{t-1})$ does not depend on x_t, the optimal decision x_t^* is therefore the object with the largest posterior probability having the desired class label. This decision is intuitive: with only one evaluation remaining, there is no possible benefit to explore, hence suggesting to make a greedy final attempt. Typically, virtual screening sorts compounds for testing by their similarity to known actives (which is interpreted as a ranking by probability of activity); from the above, we can see that this corresponds to the optimal active search policy in the extremely myopic case of being permitted only one observation. Thus, 1-NN similarity search is similar to one-step lookahead for a particular choice of model $\Pr(y = 1 \mid x, \mathscr{D})$.

Given the optimal strategy for selecting x_t, we now consider the problem of choosing the location of the second-to-last point x_{t-1}. When making the decision in this case (as well as with any other x_i with $i < t$), the problem becomes more difficult because one must now consider the possible consequences of the choices and the way they might impact future decisions. Hence, during the calculation of the expected utility for the two-step lookahead case, we must integrate out the unknown location of the final observation x_t and its label:

$$\mathbb{E}\left[u(\mathscr{D}_t) \mid x_{t-1}, \mathscr{D}_{t-2}\right] = \iiint u(\mathscr{D}_t)\Pr(y_{t-1} \mid x_{t-1}, \mathscr{D}_{t-2}) \cdot p(x_t \mid \mathscr{D}_{t-1})\Pr(y_t \mid x_t, \mathscr{D}_{t-1})\,\mathrm{dy_{t-1}}\,\mathrm{dx_t}\,\mathrm{dy_t}. \quad (1)$$

Note, however, that the integral over x_t can be evaluated trivially because $p(x_t \mid \mathscr{D}_{t-1})$ is simply $\delta(x_t - x_t^*)$, where δ is the Dirac delta function—that is, given the value of y_{t-1}, the location of the last choice x_t is deterministic and known, as discussed above. Notice that the value of x_t^* depends on the data $\mathscr{D}_{t-1}$, including the unknown value of y_{t-1}; it might differ as a function of that conditioning.

Therefore, to evaluate the two-step expected utility at a point x_{t-1}, we sample over the unknown value $y_{t-1} \in \{0, 1\}$, and for each possible value of y_{t-1}, find the optimal last observation x_t^*, given the fictitious observation as described above. Note that sampling over y_t^* is not required in the search case, because the expected one-step utility has already been evaluated.

The procedure described above can be repeated recursively to calculate the expected ℓ-step lookahead utility of choosing a point for any $\ell \leq t$ thus increasing the potential of the method.

We note that, in general, the computation of expected utility for each unlabeled point can be prohibitive for long lookahead horizons. However, Garnett et al. [6] showed how the search space can be effectively pruned if a model permits certain bounds. Namely, the model must satisfy two conditions:

- Observing an inactive point cannot raise activity probabilities of remaining unlabeled points. That is, for any set of observations $\mathscr{D}$, if we augment $\mathscr{D}$ with a single negative observation at x' to build $\mathscr{D}' = \mathscr{D} \cup \{(x', 0)\}$, we have

  $$\Pr(y = 1 \mid x, \mathscr{D}') \leq \Pr(y = 1 \mid x, \mathscr{D}'),$$

  for all unlabeled x.
- There must be a nontrivial bound on the maximum probability among unlabeled points after a given number of additional active observations. That is, there exists a function $p^*(n, \mathscr{D})$ such that

  $$p^*(n, \mathscr{D}) \leq \max \Pr(y = 1 \mid x, \mathscr{D} \cup \mathscr{D}', \textstyle\sum_{y' \in \mathscr{D}'} \leq n), \quad (2)$$

  which bounds the maximum probability among unlabeled points after adding at most n additional positive observations to $\mathscr{D}$.

The pruning operates by identifying data points that cannot possibly be an optimal solution by bounding their expected utility top-down with little computational overhead. In practice, this pruning procedure, which is also applied herein (see below), enables multiple-step-lookahead active search on large databases.

Generic MATLAB code implementing active search with arbitrary models and lookahead horizons is freely available under a permissive license [8].

Calculations

Active search is based upon a probabilistic model yielding the probability that a test compound will be active, given a current set of labeled examples $\mathscr{D}$. Here, we used an NN classifier based on the Tanimoto coefficient (Tc) [9] of binary fingerprint representations. Let $x, x' \in \{0, 1\}^n$ be two n-dimensional binary strings. The Tanimoto coefficient $t(x, x')$ is defined by:

$$t(x, x') = \frac{\sum_{i=1}^{n} \min(x_i, x_i')}{\sum_{i=1}^{n} \max(x_i, x_i')}.$$

Let $\mathrm{NN}(x)$ represent the k-NNs of a compound x in $\mathscr{X}$ and $\mathrm{L\text{-}NN}(x)$ the subset of $\mathrm{NN}(x)$ for which class labels are currently known. The probability of activity is defined as

$$\Pr(y = 1 \mid x, \mathscr{D}) \stackrel{\text{def}}{=} \frac{\gamma + \sum_{x' \in \mathrm{L-NN(x)}} t(x, x')y'}{1 + \sum_{x' \in \mathrm{L-NN(x)}} t(x, x')}.$$

Here the constant $\gamma \in [0, 1]$ serves as a "pseudocount" to define the probabilities for objects having few labeled neighbors. In our calculation, search parameters $k = 100$ and $\gamma = 10^{-3}$ were applied, and this model satisfies the properties required for pruning the active search space [6]. The active search procedure uses pre-selected molecular representations and does not involve optimization of descriptors (or latent variables).

For each search task (defined by an activity class and fingerprint), 20 independent calculations were carried out. In each case, a single active compound was randomly selected to serve as the initial set $\mathscr{D}$. Then one-step- or two-step-lookahead protocols were applied for active search to sequentially select 500 additional compounds for class label assignment.

As a control, to put the performance of active search into perspective, iterative 1-NN similarity searching [10] was carried out on the same reference and test sets and also for 500 iterations in each case. In 1-NN similarity searching, each test compound is compared to the set of known active compounds and the largest Tc value (resulting from the comparison with the most similar reference compound) is assigned to the test compound as the final similarity score [10]. Compared to other k-NN protocols, iterative 1-NN similarity searching most closely mimics the one-step-lookahead procedure.

Data sets and fingerprints

For practical evaluation, 120 activity classes from Binding DB [11] previously described by Heikamp and Bajorath [12] were selected. The sets contained compounds with confirmed activity of at least 1μM potency (K_i or IC_{50}) against different human targets. In each case, all active compounds represented positive objects. In addition, 100,000 compounds were randomly chosen from ZINC [13] to represent negative objects.

For each compound, three binary molecular fingerprints were computed from the two-dimensional chemical structure to serve as representations: ECFP4 [14], GpiDAPH3 [15], and MACCS [16]. ECFP4 is a so-called atom environment fingerprint that encodes individual atoms of a molecule and environments of increasing radius of these atoms as an integer via a hashing function [14]. ECFP4 encodes environments up to a bond distance of two from the central atom and is able to distinguish $\sim 4 \times 10^9$ distinct environments. MACCS is a dictionary-based substructure fingerprint that encodes 166 predefined substructural features of 1–10 atoms. GpiDAPH3 is a 2D pharmacophore fingerprint that assigns one of eight types to each atom and encodes triplets of these atoms and their bond distances, giving rise to 30,240 distinct features.

The use of these three representations resulted in 360 separate search trials (three for each activity class) for each search strategy (one-step, two-step active search, and 1-NN similarity search).

Results

In order to assess the utility of the active search approach for the identification of active compounds, the performance of the one-/two-step-lookahead procedure was evaluated on a large scale and compared to iterative 1-NN similarity searching. The alternative search approaches were systematically compared on 120 different activity classes. The performance was evaluated by determining the number of active compounds identified after a fixed number of iterations. We note that conventional receiver-operating characteristic (ROC) and area under ROC (AUC) calculations are not applicable to assess the performance of active search. ROC and AUC are based upon a complete (fixed) ranking. By contrast, the key aspect of active search ist that beliefs are updated after each iteration, after which only the top compound is investigated. The ranking is never considered beyond this.

Table 1 reports the average number of active compounds found by 1-NN similarity searching and the one-/two-step-lookahead strategy after all 500 iterations over all search trials (including all activity classes). In 1-NN similarity searching, for the overall best performing fingerprint (ECFP4), on average, ~ 49.1 % of all selected compounds were active. In active search, for the best-performing fingerprint (ECFP4), on average ~ 58.3 % of all compounds selected by one-/two-step lookahead were active. Hence, compared to iterative similarity searching, active search yielded superior predictions. Furthermore, there was a consistent increase in the number of correctly identified active compounds from 1-NN over one-step- to two-step-lookahead search (Table 1). In k-NN similarity searching, only active compounds are taken into consideration. By contrast, active search considers both the probability of activity and inactivity, as detailed in the methods section, which sets it apart from similarity searching and might rationalize its higher performance. One-step-lookahead selects the compound with highest probability of activity at each iteration, while two-step-lookahead combines probabilities of activity and inactivity for the first compound to identify a second compound with highest probability of activity. This renders two-step-lookahead less myopic and more exploratory than 1-NN and one-step-lookahead search.

In light of the findings reported in Table 1 the one-step- and two-step-lookahead strategies are compared in detail. For all three fingerprints, the two-step-lookahead procedure detected a larger number of active compounds at termination compared to one-step selection. On the basis of a paired t-test, the difference in the number of identified active compounds was statistically significant for all fingerprints at the $\alpha = 10^{-5}$ level.

Figure 1 monitors the improvements in search performance at termination across all 360 search trials per selection procedure by moving from one-step- to two-step-lookahead on the basis of percentiles. In 66 % of all search trials, there was an increase in the number of active compounds found (and in 79 % of all trials, there was no reduction). The median increase was 1.15 active compounds. In 10 % of the search trials, an increase of more than 13 active compounds was observed for the two-step- relative to the one-step-lookahead procedure. The percentile-based results favor the two-step-lookahead strategy.

Figure 2 reports the average differences in the number of active compounds detected by the two-step- and one-step-lookahead together with the 5th and 95th percentiles of these differences. For all fingerprints, a positive impact on search performance using the two-step strategy is evident. For the ECFP4 and GpiDAPH3 fingerprints, the 5th percentile does not reveal an advantage to either strategy, whereas the MACCS fingerprint displays a tendency towards one-step selection in this low percentile. By contrast, the 95th percentile reveals considerable improvements in search performance using the two-step-lookahead strategy for all three fingerprints.

Table 1 Global search results

Fingerprint	1-NN	1-Step	2-Step	Difference	p Value	95 % CI	
						Low	High
ECFP4	245.6	288.6	291.5	2.88	1.94×10^{-7}	1.85	3.91
GpiDAPH3	230.0	255.4	260.6	5.19	4.20×10^{-12}	3.85	6.52
MACCS	167.9	221.2	223.6	2.46	8.46×10^{-6}	1.42	3.51

The table reports the average number of actives found at termination by 1-NN fingerprint similarity searching and the one-step- and two-step-lookahead active search procedures across all 120 activity classes and 20 independent search trials for each fingerprint. The result of a two-sided paired t-test is also given, with the null hypothesis that the difference in performance between two-step-lookahead and one-step-lookahead is zero, and the corresponding 95 % confidence intervals on the difference are provided

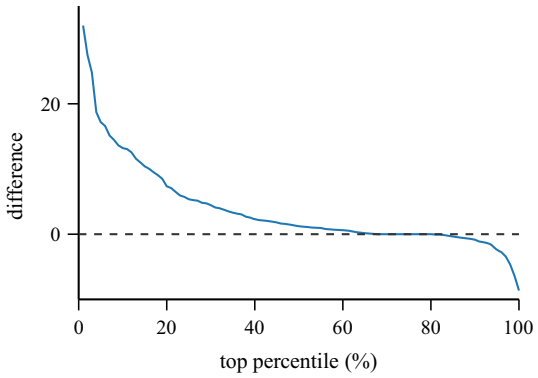

Fig. 1 Global differences in the number of identified active compounds. Percentiles of differences in number of active compounds found by the two-step- versus one-step-lookahead strategy at termination across all 360 search trials per strategy

Table 2 reports the average search performance for each activity class. For 62 of the 120 activity classes, a paired t-test confirms that that the increase in number of actives detected by two-step-lookahead was statistically significant at the $\alpha = 0.05$ level. For 44 and 30 of these activity classes, statistical significance was also confirmed at the $\alpha = 0.01$ level and $\alpha = 0.001$ level, respectively. In only three instances there was a significant decrease in the number of actives found by two-step- compared to one-step-lookahead at the $\alpha = 0.05$ level. Among these, the maximum mean decrease was 0.73 compounds. Hence, analysis of search performance at the level of individual activity classes also confirmed superior performance of the two-step-lookahead procedure, consistent with the conclusions drawn from global performance analysis.

Discussion

In light of our findings that the active search procedure improved the performance of fingerprint similarity searching on a global scale and that two-step-lookahead was superior to one-step-lookahaed, we focus here on active search and discuss the principal behavior of the two-step-lookahead strategy versus the simple one-step approach. We emphasize that the one-step approach is intrinsically myopic, whereas the two-step strategy is more exploratory. The two-step-lookahead strategy is able to make decisions that do not maximize the posterior probability of observing a given activity at the current step. Instead, it is designed to potentially explore a region in search space where the probability of immediate activity detection might be lower, but where there is an increased probability of discovering more active compounds during the next two iterations. The following simple example, taken from [6], demonstrates the effect of this tradeoff in the case of active search. Figure 3 depicts a three-point space $\mathscr{X}$. The two connected compounds have the same label; they are either both active or both inactive, with ε being the marginal probability of activity. The compound on the right is independent of the others and has probability of activity of $\delta > \varepsilon$. If two class label queries are permitted in order to detect as many active compounds as possible, the expected performance of the one- and two-step strategies can be directly determined. The one-step strategy will always select the isolated compound first and will then choose either of the two connected compounds, with expected final utility of $\varepsilon + \delta$. Following the two-step strategy, the expected final utility when choosing one of the connected compounds first is $2\varepsilon + (1 - \varepsilon)\delta$, whereas the expected final utility of selecting the isolated compound is again $\varepsilon + \delta$. For the two-step strategy, the difference in expected utility between either connected and the isolated compound is $\varepsilon - \varepsilon\delta > 0$. It follows that one of the connected compounds will always be selected first, and the two-step strategy will outperform the one-step greedy strategy on average for any positive value of δ.

This example illustrates non-trivial decisions that an optimal strategy can facilitate. Exploring regions where class labels are correlated might ultimately lead to the identification of more active compounds than single-step

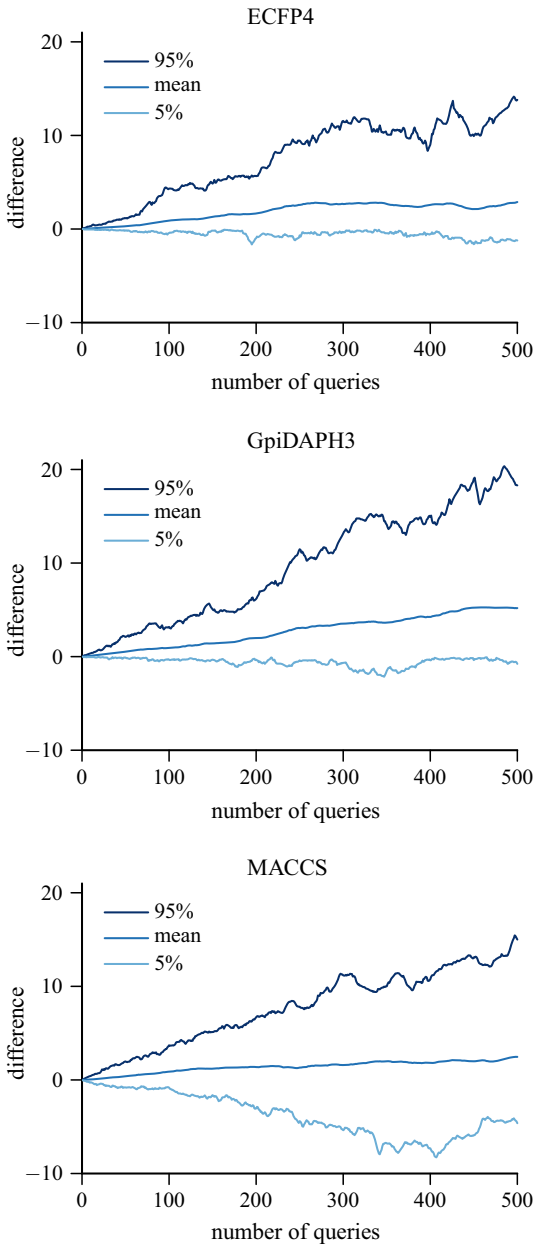

Fig. 2 Average improvement in search performance. Reported is the average increase in the number of active compounds found as a function of number of iterations (*queries*) for the two-step- versus one-step-lookahead strategy across all 120 activity classes and all search trials and fingerprints. The 5th and 95th percentiles of the differences are also shown

selection, even if the probability that the first compound selected is active is not maximal. A welcome side effect of an exploratory search is that it enables learning class labels of compounds populated in the chosen region, even if no active compound is found there. Exploratory evaluations can therefore improve the overall quality of the probabilistic model, although this goal is not explicitly defined at any step during the derivation of an active search strategy.

In our experiments, we only considered the impact of moving from greedy one-step-lookahead search to the slightly less-myopic two-step-lookahead procedure. A natural question is whether we would obtain even better results from looking three or more steps ahead. The Bayesian optimal decision can only be obtained when looking ahead to the end of the entire search horizon, and it has been shown that further lookahead can always improve performance by any arbitrary amount, in specially constructed problems [6]. As usual, however, one must weigh the benefits of theoretical optimality against increased computational expense. Without pruning, computing the ℓ-step-lookahead active search utility for each of n points takes time $\mathcal{O}\left((2n)^{\ell-1}\right)$, which grows exponentially as a function of ℓ. The pruning approach can reduce running time significantly, especially for smaller lookahead horizons; in particular, two-step lookahead was feasible on all of our problems here. As the lookahead increases, the required bound on the maximum probability, $p^*(n, \mathscr{D})$, (2) must inevitably weaken; the larger this bound, the less we may prune from the exponentially growing search space. Therefore we are effectively limited to smaller horizons in practice. Two-step lookahead was previously compared against three-step lookahead on a search problem from a non-biological domain, and although it showed marginal gains, the largest increase in performance was obtained by moving from one- to two-step lookahead [6]. Importantly, two-step lookahead is the first point where the expected utility contains terms simultaneously encouraging both exploration and exploitation. We consider two-step lookahead to represent the best general-purpose algorithm: increased search performance with relatively low additional computational overhead due to still-effective pruning.

Finally, we note that we have limited our discussion here to decision-theoretical questions: given a model, how may we use it to select points for investigation? The computation and maximization of expected utility does not rely on a particular form for the classification model $p(y = 1 \mid x, \mathscr{D})$. Of course, we can expect that having a more-accurate model can lead to better performance, so in practice we would recommend careful model construction and selection. We regard the active search framework as a meta-procedure to potentially use the derived model more effectively when searching for active compounds. We do note that satisfaction of the pruning requirements might be a property to consider during the model-construction phase of analysis; in some cases, we expect that a simpler model with two-step lookahead could outperform a more-complicated and expensive model used greedily. The development of alternative, more-general search-space pruning mechanisms could also represent an interesting direction for future research.

Table 2 Activity-class specific active search performance

#	Name	Symbol	Mean	Significance	95 % CI	
					Low	High
1	11β-Hydroxysteroid dehydrogenase	HSD11B1	8.28	****	4.80	11.77
2	5-Lipoxygenase	ALOX5	6.13	**	1.79	10.48
3	Acetylcholinesterase	ACHE	4.23		−1.16	9.63
4	Adenosine A_3 receptor	ADORA3	3.97		−1.10	9.03
5	Adenosine kinase	ADK	5.60		−0.28	11.48
6	Adenosine A_1 receptor	ADORA1	19.38	****	13.64	25.12
7	Adenosine A_{2A} receptor	ADORA2A	11.27	*	1.50	21.03
8	Adenosine A_{2B} receptor	ADORA2B	15.87	****	11.81	19.92
9	Aldose reductase	AKR1B1	6.37	*	0.68	12.05
10	α_{1A} adrenergic receptor	ADRA1A	−0.40		−1.11	0.31
11	α_{1B} adrenergic receptor	ADRA1B	−0.02		−0.33	0.30
12	α_{2A} adrenergic receptor	ADRA2A	0.08		−0.50	0.67
13	Androgen receptor	AR	11.03	****	7.97	14.09
14	Angiotensin-converting enzyme	ACE	−0.35	*	−0.62	−0.08
15	Aromatase	CYP19A1	5.15	****	3.22	7.08
16	Butyrylcholinesterase	BCHE	1.18	**	0.42	1.95
17	C–C chemokine receptor type 3	CCR3	0.48		−0.52	1.49
18	Calpain–1	CAPN1	5.37	****	3.28	7.45
19	Cannabinoid receptor type 1	CNR1	2.15		−0.26	4.56
20	Cannabinoid receptor type 2	CNR2	−2.27		−6.47	1.94
21	Carbonic anhydrase I	CA1	5.48	****	3.12	7.85
22	Carbonic anhydrase II	CA2	1.87		−0.62	4.36
23	Carbonic anhydrase IX	CA9	3.43	**	1.28	5.59
24	Caspase-3	CASP3	0.30		−0.45	1.05
25	Cathepsin K	CTSK	3.37		−4.03	10.76
26	Cathepsin L	CTSL1	0.22		−1.43	1.86
27	Cathepsin S	CTS	0.58		−2.85	4.02
28	Cyclin-dependent kinase 2	CDK2	1.62	*	0.03	3.21
29	Checkpoint kinase 1	CHEK1	4.93		−0.61	10.48
30	Corticotropin-releasing hormone receptor 1	CRHR1	1.20	*	0.25	2.15
31	Cyclin-dependent kinase 1	CDK1	13.77	****	10.02	17.51
32	Cyclin-dependent kinase 2	CDK2	1.42		−3.75	6.58
33	Cyclin-dependent kinase 4	CDK4	2.58	*	0.18	4.98
34	Cyclooxygenase-2	PTGS2	7.48	*	0.97	13.99
35	δ-Opioid receptor	OPRD1	2.38	**	0.81	3.95
36	Dihydrofolate reductase	DHFR	1.12	*	0.17	2.06
37	Dipeptidyl peptidase-4	DPP4	5.82	*	0.93	10.70
38	Dopamine receptor D_1	DRD1	−0.25		−1.83	1.33
39	Dopamine receptor D_3	DRD3	12.53	***	6.41	18.66
40	Dopamine receptor D_4	DRD4	6.85	**	1.77	11.93
41	Dopamine receptor D_2	DRD2	7.25	****	3.86	10.64
42	Dopamine transporter	SLC6A3	5.40	**	2.16	8.64
43	Epidermal growth factor receptor	EGFR	2.88	*	0.29	5.48
44	Endothelin receptor type A ET_A	EDNRA	2.50	*	0.39	4.61
45	Epoxide hydratase 1	EPHX1	−0.27		−1.18	0.65
46	Estrogen receptor α	ESR1	−0.03		−0.48	0.42
47	Estrogen receptor β	ESR2	1.05		−0.50	2.60

Table 2 continued

#	Name	Symbol	Mean	Significance	95 % CI	
					Low	High
48	Factor Xa	F10	2.30	*	0.02	4.58
49	Fibroblast growth factor receptor 1	FGFR1	0.40		−0.20	1.00
50	Glucocorticoid receptor	HR3C1	1.98	**	0.53	3.44
51	Glycogen phosphorylase	PYGL	−0.08		−0.17	0.00
52	Glycogen synthase kinase 3 β	GSK3B	3.48		−1.24	8.20
53	Human epidermal growth factor receptor 2	ERBB2	−0.42		−1.25	0.42
54	hERG	KCNH2	6.18	****	3.26	9.11
55	Histamine H_3 receptor	HRH3	8.63	***	4.30	12.96
56	Histone deacetylase 1	HDAC1	−2.47		−9.99	5.05
57	IκB kinase 2	IKBKB	1.57		−0.33	3.46
58	Inosine monophosphate dehydrogenase 2	IMPDH2	9.73	***	4.51	14.96
59	Integrin $\alpha_{IIb}\beta_3$	ITGA2B/ITGB3	1.87	**	0.66	3.07
60	Integrin $\alpha_V\beta_3$	ITGAV/ITGB3	0.17		−0.28	0.62
61	Interleukin 8 receptor β	CXCR2	0.00		0.00	0.00
62	κ-Opioid receptor	OPRK1	−0.28		−3.13	2.56
63	Leukocyte elastase	ELANE	0.32		−0.22	0.85
64	Leukotriene A_4 hydrolase	LTA4H	0.80		−0.40	2.00
65	Mitogen-activated protein kinase p38-α	MAPK14	8.08	*	1.38	14.78
66	Matrix metalloproteinase 1	MMP1	0.93	**	0.29	1.58
67	Matrix metalloproteinase 13	MMP13	0.20		−0.57	0.97
68	Matrix metalloproteinase 2	MMP2	3.25	****	2.05	4.45
69	Matrix metalloproteinase 8	MMP8	17.53	****	13.11	21.96
70	Matrix metalloproteinase 9	MMP9	1.08		−0.31	2.48
71	Melanin-concentrating hormone receptor 1	MCH1R	8.50	***	4.29	12.71
72	Melatonin receptor 1A	MTNR1A	0.63	*	0.02	1.24
73	Melatonin receptor 1B	MTNR1B	0.95		−0.12	2.02
74	Metabotropic glutamate receptor 5	GRM5	0.90	*	0.06	1.74
75	Methionine aminopeptidase 2	METAP2	4.30	***	1.87	6.73
76	Mitogen-activated protein kinase 8	MAPK8	0.52		−1.17	2.21
77	μ-Opioid receptor	OPRM1	4.35	****	2.36	6.34
78	Muscarinic acetylcholine receptor M_1	CHRM1	1.38		−0.78	3.55
79	Muscarinic acetylcholine receptor M_2	CHRM2	10.25	****	7.40	13.10
80	Muscarinic acetylcholine receptor M_3	CHRM3	5.82	****	3.36	8.28
81	Neprilysin	MME	−0.63	**	−1.03	−0.24
82	Neuropeptide Y receptor 5	NPY5R	6.65	**	1.98	11.32
83	Nitric oxide synthase	NOS1	−0.73	****	−0.93	−0.54
84	Tachykinin receptor	NK1	1.53	****	1.06	2.01
85	Nociceptin receptor	OPRL1	4.78		−0.11	9.68
86	Norepinephrine transporter	SLC6A2	5.75	***	2.73	8.77
87	PDGFR-β	PDGFRB	−0.88		−2.07	0.30
88	Phosphodiesterase 4A/4B/4C/4D	PDE4A/B/C/D	8.12	*	1.64	14.60
89	Phosphodiesterase 3A/3B	PDE3A/B	0.33	**	0.10	0.56
90	Phosphodiesterase 5	PDE5	3.30		−2.24	8.84
91	Platelet-activating factor receptor	PTAFR	8.83	****	5.02	12.65
92	Poly [ADP-ribose] polymerase-1	PARP1	3.00		−1.43	7.43
93	Progesterone receptor	PGR	−0.07		−1.71	1.58
94	Protein farnesyltransferase subunit β	FNTB	10.55	***	5.02	16.08

Table 2 continued

#	Name	Symbol	Mean	Significance	95 % CI	
					Low	High
95	Protein kinase B	ATK1	1.45		−0.89	3.79
96	Protein kinase C θ	PRKCQ	1.58	****	0.83	2.34
97	5-HT_{1A}	HTR1A	8.07	**	3.17	12.97
98	5-HT_{1B}	HTR1B	3.72	*	0.51	6.92
99	5-HT_{1D}	HTR1D	0.23		−7.02	7.49
100	5-HT_{2A}	HTR2A	8.35	****	5.02	11.68
101	5-HT_{2B}	HTR2B	1.23		−0.15	2.61
102	5-HT_{2C}	HTR2C	3.52	*	0.76	6.27
103	5-HT_6	HTR6	0.72		−0.05	1.48
104	5-HT_7	HTR7	2.85		−0.00	5.70
105	Serotonin transporter	SLC6A4	6.70	***	3.41	9.99
106	σ-Opioid receptor	SIGMAR1	4.93	**	1.78	8.08
107	c-Src	SRC	0.28		−0.07	0.64
108	Thrombin	F2	3.60		−0.58	7.78
109	Thromboxane A synthase	TBXAS1	0.65	*	0.16	1.14
110	Thromboxane A2 receptor	TBXA2R	2.00	****	1.39	2.61
111	Trypsin	PRSS1	0.75	***	0.36	1.14
112	Tumor necrosis factor-α-converting enzyme	ADAM17	0.42		−0.10	0.93
113	Tyrosine-protein kinase Lck	LCK	2.38		−1.42	6.19
114	Tyrosine-protein kinase Src	SRC	5.08	***	2.35	7.82
115	Tyrosine-protein kinase c-Kit	KIT	0.83		−0.84	2.51
116	Tyrosine-protein kinase CSF1R	CSF1R	4.72	**	1.76	7.67
117	Urokinase-type plasminogen activator	PLAU	0.00		0.00	0.00
118	Vanilloid receptor 1	TRPV1	3.35		−0.27	6.97
119	VEGF receptor 1	FLT1	−3.03		−6.58	0.52
120	VEGF receptor 2	KDR	4.42		−1.82	10.65

The table reports the average difference in number of actives found by the one-step- and two-step-lookahead strategy for each activity class over 60 search trials (20 independent calculations for each of three fingerprints). A 95 % confidence interval on the difference from a paired *t* test is also provided. Significance markers */**/*** /**** indicate that the difference is significant at the $\alpha = 0.05/0.01/0.001/0.0001$ level

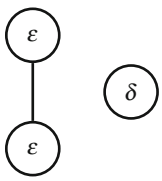

Fig. 3 Probability model of active search. A simple probability model comparing the typical behavior of the two-step versus the (greedy) one-step strategy is shown. The points on the *left* are known to have the same class label, and the marginal probability of the label is $\Pr(y = 1) \stackrel{\text{def}}{=} \varepsilon$. The label of the solitary point on the *right* is independent of the others having a probability of $\Pr(y = 1) \stackrel{\text{def}}{=} \delta > \varepsilon$

Conclusions

Herein we have introduced a probabilistic approach to iterative ligand-based virtual screening that further extends the potential of standard fingerprint similarity searching. The concept of active search was adapted from data pre-selection for specific analysis tasks and applied to compound screening via one-/two-step-lookahead strategies. The underlying idea is that active search iteratively increases the horizon of the search and its search space coverage and learns from class label distributions in chosen

regions of search space. The active search strategies have been systematically evaluated on a large number of compound activity classes and shown to yield superior search performance compared to iterative (standard) similarity searching. Furthermore, two-step active search outperformed the one-step strategy. On the basis of our findings, the active search concept complements existing virtual screening approaches and should be of considerable interest for iterative screening applications.

In addition to the already available MATLAB code implementing generic active search, code and data to replicate the herein described experiments will be made freely available at the following URL: https://github.com/rmgarnett/active_virtual_screening

Acknowledgments Part of this work was supported by the German Science Foundation (DFG) under the reference number GA 1615/1-1.

References

1. Geppert H, Vogt M, Bajorath J (2010) Current trends in ligand-based virtual screening: molecular representations, data mining methods, new application areas, and performance evaluation. J Chem Inf Model 50:205–216
2. Maggiora G, Vogt M, Stumpfe D, Bajorath J (2014) Molecular similarity in medicinal chemistry. J Med Chem 57:3186–3204
3. Stumpfe D, Bajorath J (2011) Similarity searching. Wiley Interdiscip Rev Comput Mol Sci 1:260–282
4. Bajorath J (2002) Integration of virtual and high-throughput screening. Nat Rev Drug Discov 1:882–894
5. Stahura F, Bajorath J (2004) Virtual screening methods that complement HTS. CCHTS 7:259–269
6. Garnett R, Krishnamurthy Y, Xiong X, Schneider J, Mann RP (2012) Bayesian optimal active search and surveying. In: Langford J (ed) Proceedings of the 29th international conference on machine learning (ICML 2012). Pineau J, pp 1239–1246
7. Robert C (2007) The Bayesian choice. Springer, New York
8. Garnett R Active Search Toolbox for MATLAB. https://github.com/rmgarnett/active_search
9. Willett P (1998) Chemical similarity searching. J Chem Inf Comput Sci 38:983–996
10. Hert J, Willett P, Wilton DJ, Acklin P, Azzaoui K, Jacoby E, Schuffenhauer A (2004) Comparison of fingerprint-based methods for virtual screening using multiple bioactive reference structures. J Chem Inf Comput Sci 44:1177–1185
11. Liu T, Lin Y, Wen X, Jorissen RN, Gilson MK (2007) Bindingdb: a web-accessible database of experimentally determined protein-ligand binding affinities. Nucleic Acids Res 35:D198–D201
12. Heikamp K, Bajorath J (2011) How do 2D fingerprints detect structurally diverse active compounds? Revealing compound subset-specific fingerprint features through systematic selection. J Chem Inf Model 51:2254–2265
13. Irwin JJ, Sterling T, Mysinger MM, Bolstad ES, Coleman RG (2012) ZINC: a free tool to discover chemistry for biology. J Chem Inf Model 52:1757–1768
14. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50:742–754
15. Molecular Operating Environment (MOE) (2012) Chemical Computing Group, Montreal, Canada
16. MACCS Structural Keys (2011) Accelrys, San Diego, CA