Teaser This paper focuses on machine-learning approaches in the context of ligand-based virtual screening for addressing complex compound classification problems and predicting new active molecules.

Machine-learning approaches in drug discovery: methods and applications

Q1 **Antonio Lavecchia**

Department of Pharmacy, Drug Discovery Laboratory, University of Napoli 'Federico II', via D. Montesano 49, I-80131 Napoli, Italy

During the past decade, virtual screening (VS) has evolved from traditional similarity searching, which utilizes single reference compounds, into an advanced application domain for data mining and machine-learning approaches, which require large and representative training-set compounds to learn robust decision rules. The explosive growth in the amount of public domain-available chemical and biological data has generated huge effort to design, analyze, and apply novel learning methodologies. Here, I focus on machine-learning techniques within the context of ligand-based VS (LBVS). In addition, I analyze several relevant VS studies from recent publications, providing a detailed view of the current state-of-the-art in this field and highlighting not only the problematic issues, but also the successes and opportunities for further advances.

Antonio Lavecchia received his PhD in pharmaceutical sciences in 1999 from the University of Catania, Italy. During his doctoral studies, he conducted research in the College of Pharmacy at the University of Minnesota (USA) as well as at the IMIM Hospital del Mar Research Institute in Barcelona (Spain). At present, he is professor of medicinal chemistry and head of the Drug Discovery Laboratory of the Department of Pharmacy, University of Napoli 'Federico II'. In 2006, he was awarded the Farmindustria Prize for Pharmaceutical Research. His research focuses on application of computational tools to the design of bioactive molecules and to the study of targeting biological systems of pharmacological interest.

Introduction

Data mining is defined as the automatic extraction of useful, often previously unknown information from large databases or data sets using advanced search techniques and algorithms to discover patterns and correlations in large pre-existing databases. Through data mining, one derives a model that relates a set of molecular descriptors to biological key attributes, such as efficacy or absorption, distribution, metabolism, and excretion (ADMET) properties. The resulting model can be used to predict key property values of new compounds, to prioritize them for follow-up screening, and to gain insight into their structure–activity relations (SARs). Data-mining models range from simple, parametric equations derived from linear techniques to complex, nonlinear models derived from nonlinear techniques [1–5]. For data-mining approaches, a major target area within the chemoinformatics spectrum is VS; that is, the application of computational tools to search large databases for new leads with higher probability of strong binding affinity to the target protein. This is also possible without knowing the molecular target or when the reference molecule(s) binds to more than one receptor (e.g., in bioprofile similarity searching). Successful studies have led to the identification of molecules either resembling the native ligands of a particular target or novel compounds [6,7]. VS methods can be classified into structure-based (SBVS) and ligand-based (LBVS) approaches based on the

E-mail addresses: antonio.lavecchia@unina.it, lavecchi@unina.it.

1359-6446/06/$ - see front matter © 2014 Published by Elsevier Ltd. http://dx.doi.org/10.1016/j.drudis.2014.10.012

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

amount of structural and bioactivity data available. If the 3D structure of the receptor is known, a SBVS method that can be used is high-throughput docking [8], but where the information on the receptor is scant, LBVS methods [4] are commonly used. Docking involves a complex optimization task of finding the most favorable 3D binding conformation of the ligand to the receptor molecule. Being computationally intensive, docking is not suitable for very large VS experiments. By contrast, LBVS methods, whose goal is to search chemical databases to find compounds that best match a given query, are popular because they are computationally inexpensive and easy to use. Furthermore, the assumption that structurally similar molecules exhibit similar biological activity compared with dissimilar or less similar molecules is generally valid. However, it is well known that small modifications of active compounds can either improve or decrease their potency and that active and inactive compounds might be similar and distinguishable only by small chemical differences [9]. This situation corresponds to the presence of 'activity cliffs' [10] in the activity landscape that produce an area of a rugged canyon-like surface [11,12], falling outside the applicability domain of global similarity approaches in LBVS [13]. Thus, LBVS methods have an increasingly important role at the beginning of the drug discovery projects, especially where little 3D information is available for the receptor.

LBVS approaches are divided broadly into similarity search and compound classification techniques [14]. Similarity search utilizes molecular fingerprints derived from molecular graphs (2D) or conformations (3D) [15,16], 3D pharmacophore models [17], simplified molecular graph representations [18], or molecular shape queries [19–22], compares them in a pair-wise manner with database compounds using a similarity metric, and produces a compound ranking in the order of decreasing molecular similarity to reference molecules. From this ranking, candidate compounds are selected. As a measure of similarity, fingerprint or feature set overlap is quantified using similarity coefficients, most frequently the Tanimoto coefficient, defined as $N_{ab}/(N_a + N_b - N_{ab})$, where N_a and N_b are the number of features/bits set in the fingerprint of compounds *a* and *b*, respectively, and N_{ab} is the number of features/bits set in both fingerprints of *a* and *b*.

Compound classification techniques are divided further into basic classification methods, such as clustering and partitioning (for which many different algorithms exist), and machine-learning approaches [23,24], such as support vector machines (SVM), decision trees (DT), *k*-nearest neighbors (*k*-NN), naïve Bayesian methods and artificial neural networks (ANN), which are becoming increasingly popular in LBVS. The goal of all these techniques is to predict compound class labels (e.g., active versus inactive) on the basis of models derived from training sets, as well as to provide a ranking of database compounds according to the probability of activity [25]. In addition, selection of compounds for the assembly of target-focused compound libraries is also possible [26]. The first application of machine learning in drug discovery was substructural analysis (SSA), which was described by Cramer *et al.* as a tool for the automated analysis of biological screening data [27]. Machine learning is now an active area of research in computer science, with the increasing availability of big data collections of all sorts prompting interest in the development of novel tools for data mining [28,29]. Thus, taken together, there is a broad spectrum of applications for machine-learning methods in computer-aided drug discovery that makes it attractive to review selected approaches and to highlight their applications.

Here, I survey the most popular machine-learning approaches in the context of LBVS, paying particular attention to novel algorithms and methods that have evolved to largely dominate the field at present. I concentrate on the developments of these methodologies over the past few years and highlight the benefits, bottlenecks, and successes of each.

Support vector machines

SVMs, developed by Vapnik and coworkers [30,31], are supervised machine-learning algorithms for facilitating compound classification, ranking and regression-based property value prediction. Typically, SVMs are used for binary property or activity predictions, for example, to distinguish between drugs and nondrugs [32,33] or between compounds that have or do not have specific activity [33–35], synthetic accessibility [36], or aqueous solubility [37]. First, the compound libraries are projected into a high-dimensional feature space where molecules, represented as descriptor vectors, hopefully become linearly separable, as visualized in Fig. 1. This projection is achieved via the use of a kernel function, such as one of the following four families of functions: linear, polynomial, sigmoid, and radial basis (RBF). The first three functions are global kernels, and only RBF is a local kernel. Extensive work has shown that RBF-based SVM outperforms SVM based on the other three kernels and, thus, is used widely [38]. The Gaussian or other polynomial kernel functions are often used in LBVS in combination with numerical property descriptors or 2D fingerprints, but simple linear kernels have also been used successfully [39]. The choice of SVM kernels and setup of the kernel parameters are largely dependent on empirical and experimental analysis, because no well-established methods are currently available for this. Foody and Mathur showed that the kernel parameters and the error penalty factor, *C*, defined by the user, rather than the class of the chosen kernels, are the decisive factors in the performance of SVM [40].

Once linearly separable, the two classes of compound can be separated in this feature space by a hyperplane. In fact, there is an infinite number of such hyperplanes and SVM chooses the hyperplane that maximizes the margin between the two classes on the assumption that the larger the margin, the lower the error of the classifier when dealing with unknown data. The hyperplanes that define such margins are called 'support hyperplanes', and the data points that lie on these hyperplanes are the 'support vectors' (Fig. 1). In the case of nonseparable classes, which are common, the soft-margin hyperplane is applicable, which maximizes the margin while keeping the number of misclassified samples minimal.

In LBVS, the scores derived by a SVM classification have also been successfully used to rank database compounds according to their decreasing probability of activity [35,41]. The signed distance between a candidate compound and the hyperplane can be used for such a ranking. To further improve the SVM ranking, which is generally undervalued compared with the tendency by SVMs to optimize the classification performance, two studies have introduced specialized ranking functions for VS that utilize optimization functions to minimize the ranking error [42,43].

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

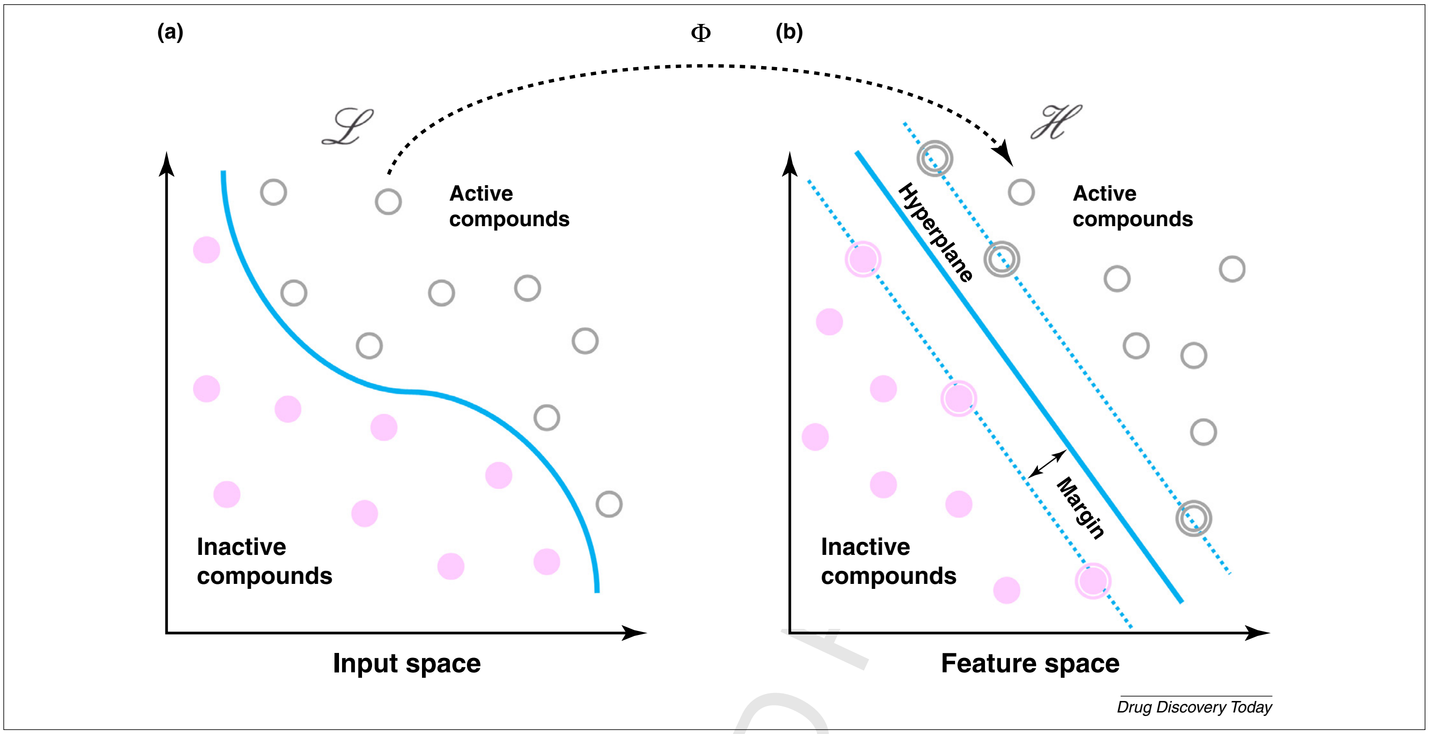

FIGURE 1

Projection into high-dimensional feature space. Using a mapping function Φ, active (empty gray points) and inactive (filled pink points) compounds that are not Q7 linearly separable in low-dimensional input space $\mathcal{L}$ **(a)** are projected into high-dimensional feature space $\mathcal{H}$ (b) and separated by the maximum-margin hyperplane. Points intercepted by the dotted line are called 'support vectors' (circled points).

A variety of new kernel functions have been introduced for SVMs, including both ligand and target kernels that capture rather different information for similarity assessment [44–46], such as graph or descriptor similarity (compounds) and sequence or binding site similarity (target proteins). For example, graph kernels [47,48], that enable one to compute the overall similarity between labeled graphs, have the advantage of enabling similarity measurement without the need to compute or store a vector representation of the compounds. However, they are computationally expensive and necessitate parameter determination. The Tanimoto kernel [49], defined in accordance with the popular Tanimoto coefficient, is widely applied to molecular fingerprints. Using different fingerprint representations or descriptor vectors of molecules, the comparison of different compound properties is possible. In addition, the Tanimoto kernel is parameter free. Furthermore, kernel functions that consider the 3D structure of compounds have been developed. For example, the pharmacophore kernel [50], which focuses on three-point pharmacophores in 3D space, outperforms fingerprint representations of pharmacophores in SVM calculations [50]. Different kernel functions [51] have been introduced for molecular representations at different levels, ranging from 1D SMILES strings and 2D bond graphs to 3D atom coordinates. However, Azencott *et al.* overall showed that the 2D kernel functions for feature vectors outperform kernel functions designed for higher-dimensional compound representations [51]. Ligand and target kernels have also been combined in the so-called 'target–ligand kernel' [52,45], where the similarity in target–ligand space is expressed as the tensor product of pairwise ligand and target similarities. Interestingly, the most precise protein kernel (based on sequence similarity, structural similarity, or ontology information) was not necessarily the most reliable. Bajorath and colleagues [53] suggested that simplified strategies for designing target–ligand SVM kernels should be used because varying the complexity of the target kernel does not influence the identification of ligands much for virtually deorphanized targets. Hence, predicting protein–ligand association is dominated by the ligand neighborhood [53]. Meslamani *et al.* [54] proposed that the use of kernel functions, taking into account true 3D cavity descriptors rather than simple sequence-based target, slightly enhances the accuracy of the models to discriminate true target–ligand complexes from false pairs. Recently, newly designed kernel functions have been introduced that compare compound pairs based on the 'matched-molecular pairs' [55]. These kernel functions capture chemical transformation and core structure information for pairs of compounds to predict 'activity cliffs', that is, structurally similar compounds having large potency differences, from which SAR determinants can be deduced.

A new development of SVM research is the introduction of hybrid techniques, according to which multiple machine-learning methods are combined to improve the quality of predictions. For example, Plewczynski [56] proposed the 'brainstorming approach', which effectively combines different powerful supervised machine-learning methods (i.e., SVM, random forest, neural networks, and DT), trained on an initial set of active compounds, into a single metapredictor that can be used to search for unknown inhibitors. This metapredictor approach achieved higher performance than any single method used in consensus. Similarly, Cheng *et al.* [57] introduced a new method to classify cytochrome P450 inhibitors and non-inhibitors by combining different single machine-learning classifiers algorithms, including SVM, DT, *k*-NN,

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

and naïve Bayesian classification, fused in a back-propagation artificial neural network (BP-ANN) algorithm. The overall performance of the newly developed combined classifier was found superior to that of three classic fusion techniques (mean, maximum, and multiply), and led to improvements in predictive accuracy.

Xie *et al.* [58] focused on the application of two-stage SVM and docking calculations for searching novel c-Met tyrosine kinase inhibitors from 18 million compounds. The combined approach considerably increased hit rates and enrichment factors of active compounds compared with the individual methods. The authors identified 1000 top-ranked virtual hits, with eight of the 75 selected hits tested active (hit rate 10.7% after additional selection). In a recent paper, Meslamani *et al.* [59] presented an automated workflow, PROFILER, using several methods to predict active compounds for different targets. The protocol used four ligand-based (SVM classification, SVR affinity prediction, nearest neighbors interpolation, and shape similarity) and two structure-based approaches (docking and protein–ligand pharmacophore match) in series, according to well-defined ligand and target property checks. The workflow successfully recovered the main targets of 189 clinical candidates in 72% of the cases and enabled the deciphering of previously unknown cross-reactivities of some drug candidates to unrelated targets.

New techniques similar to SVMs have also appeared recently in the field of chemoinformatics. For example, Tipping [60] introduced the relevance vector machines (RVMs), a Bayesian inference-based machine-learning method, which has an identical functional form to SVM, but provides probabilistic classification. This methodology has also been successfully applied to LBVS [61].

Decision tree

DT comprises a set of ‘rules’ that provide the means to associate specific molecular features and/or descriptor values with the activity or property of interest. The DT approach has been applied to problems such as designing combinatorial libraries, predicting ‘drug-likeness’, predicting specific biological activities, and generating some specific compound profiling data. This method is used not only for the identification of substructures that discriminate activity from nonactivity within a given compound database [62], but also for the classification of chemical compounds into drug and nondrug [63]. DTs are also used to predict ADME/Tox properties, such as absorption [64,65], distribution [66], solubility or permeability of drugs [67], P-glycoprotein [68] or blood–brain barrier (BBB) penetration [69], and metabolic stability [70].

A DT is commonly depicted as a tree, with the root at the top and the leaves at the bottom, as displayed in Fig. 2a. Starting from the root, the tree splits from the single trunk into two or more branches. Each branch itself might further split into two or more branches. This continues until a leaf is reached, which is a node that is not further split. The split of a branch is referred as an internal node of the tree. The root and leaves are also referred to as nodes. Each leaf node is assigned with a target property, whereas a nonleaf node (root or internal node) is assigned with a molecular descriptor that becomes a test condition with branches out into groups of differing characteristics. An unknown compound is classified based on the leaf node that it reaches after going through a series of questions (nodes) and answers (deciding which branches to take), starting with the first question from the root node. In the example in Fig. 2a, an unknown compound will be classified with target property Y_A, if it fulfills a certain condition for molecular descriptor X_1. Otherwise, molecular descriptor X_2 of the unknown compound is checked at the next step. If the value is less than 1, the unknown compound will be marked with target property Y_A. If not, the unknown will be given the label of target property Y_B.

DTs are generally formed in a top-down manner and the tree construction process focuses on selecting the best test conditions to expand the extremities of the tree. The quality of the test conditions (i.e., the conditions used to split the data at the node) is usually determined by the ‘purity’ of a split, which is often computed as the weighted average of the purity values of each branch, where the weights are determined by the fraction of examples that follow that branch. The metrics (e.g., information gain) used to select the best test generally prefer test conditions that result in a balanced tree, where purity is increased for most of the examples, over test conditions that yield high purity for a relatively small subset of the data but low purity for the rest [71]. Entropy, information-gain ratio [72], or Gini diversity index [73] can be used as measure for the best classification. Thus, rare cases, which correspond to high purity branches covering few examples, will often not be included in the decision tree.

DT models are simple to understand, interpret, and validate. However, their predictions are known to suffer from high variance. Often a small change in the data can result in a different series of splits, complicating the interpretation. This instability is the result of the hierarchical nature of the process: the effect of an error in the top split is disseminated down to all the splits below. In addition, the structure of the decision tree is sensitive to small changes in the training data. The DT learning process is significantly affected if the training data set size is small. By contrast, a huge training set might introduce overfitting of the tree. Therefore, it is recommended to maintain a moderate training data size, a height balance tree structure with a moderate number of levels, and provision for heuristically improving the classification accuracy by adding or removing subtrees at the lowest level. The performance of a DT also depends on the proper selection of a sequence of splitting attributes of the training set for different levels in the hierarchy. The splitting attributes need to be sorted according to decreasing order of merit or importance. It is essential that the most important attribute is used for splitting at the root node, and the next in the rank for the immediate descendants of the root, and so on.

Ensemble methods

A common process to limit high variance is pruning of the tree using either model complexity parameters or cross-validation. Generally, a single DT does not provide a high-performance model. Ho proposed the use of an ensemble of DTs, each created using a subset of the total descriptor set to increase the variance of the predictions, which he called the ‘random decision forest’ [74]. Ensemble techniques, such as bagging [75], boosting [76], and stacking [77], are better predictors than an individual constituent learner and benefit from variability in the ensemble members, and so they take advantage of the variance of DTs. The modern adaption of the random decision forest, the random forest (RF) algorithm developed by Breiman [78], introduced bagging and

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

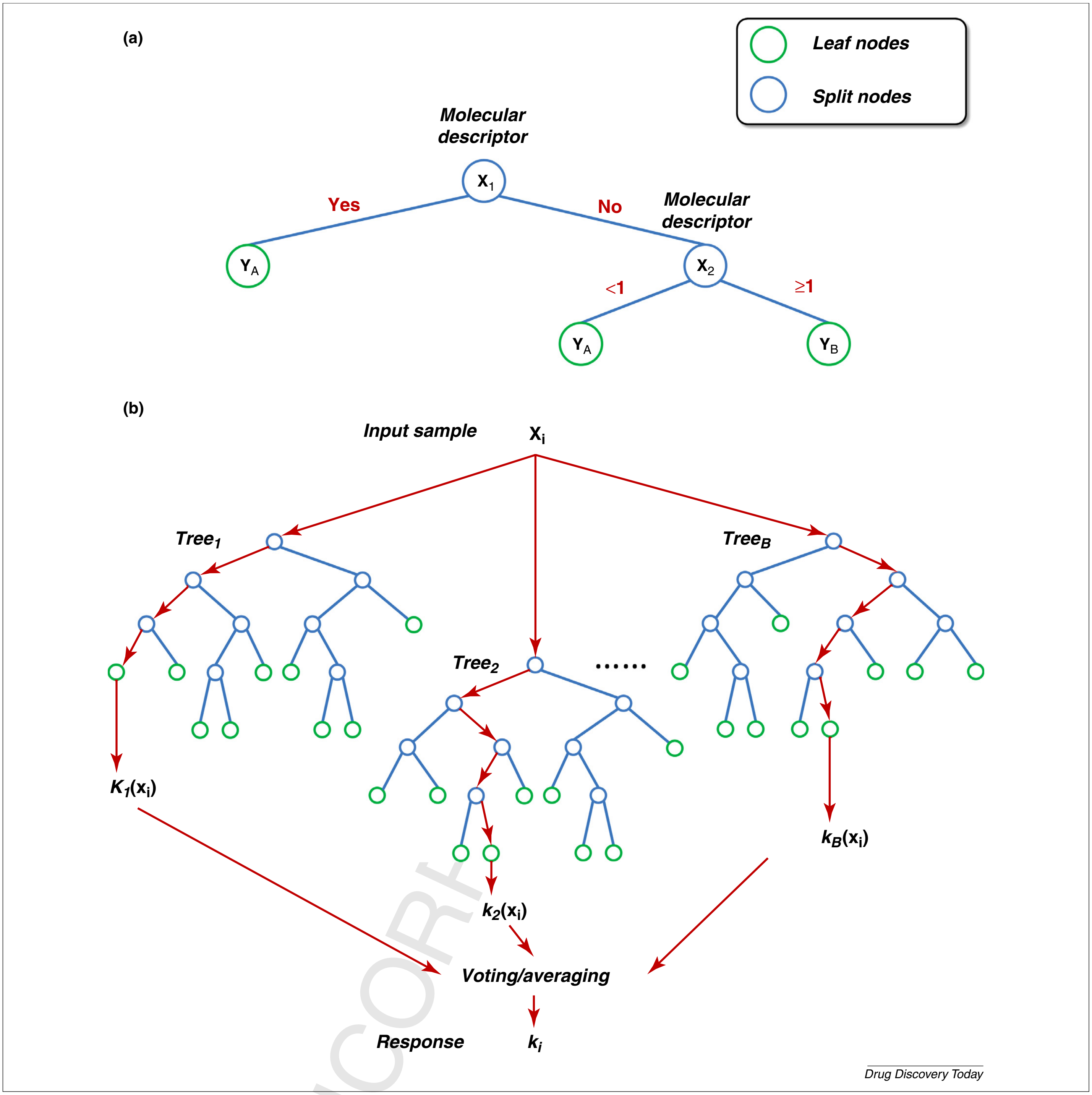

FIGURE 2

Decision tree. **(a)** The diagram displays a 2D structure with a single root node, followed by a set of yes/no decisions (binary splits) that finally result in a set of leaf nodes. For classification, a test event is passed from the root node down the tree and will end up in a certain leaf node depending on how it responded to the various split criteria. The event is then classified according to the class label of this leaf node. In the example, molecules with target properties Y_A and Y_B are classified based on two descriptors, X_1 and X_2. **(b)** A general architecture of a random forest (RF). Tree structures indicate yes/no rules at each branching, with the associated subspace partitioning of a hypothetical 2D space shown. Individual predictions from all trees are collected and combined as a single ensemble prediction by voting (for classification) or averaging (for regression).

subset selection at each node of the DT to increase further prediction variance. RF is an ensemble classifier comprising many DTs (Fig. 2b). Many classification trees are grown during training. A training set is created for each tree by random sampling with replacement from the original data set. During the construction of each tree, approximately one third of the cases are left out of the selection and this becomes the out-of-bag cases that are used as a test set. The classification performance of the test set is evaluated based on the out-of-bag error rates. Features will not be deleted based on one decision or one tree, but many trees will decide

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

and confirm elimination of features. Another positive characteristic of RF is that it is applicable to high-dimensional data with a low number of observations, a large amount of noise, and high correlated variables. Moreover, RF is less prone to overfitting and can handle the problem of imbalanced classes. RF algorithms can also be used for regression, but their advantages are less clear.

RF models have been proved to further increase the LBVS performance of individual DTs. Moreover, RF has attractive properties that have previously been found to improve the prediction of quantitative SAR (QSAR) data [79]. These properties include relatively high accuracy of prediction, built-in descriptor selection, and a method for assessing the importance of each descriptor to the model. Tong and coworkers [80] published a similar method, called Decision Forest, which uses a different set of descriptors to build diverse accurate decision tree models. This method was applied to mining estrogen receptor binders from a data set of 57 000 molecules.

RF has also found applications in the area of post dock-scoring functions and predicting protein–ligand binding affinity. For example, in a recent study by Ballester and Mitchell [81], the scoring function (RF-Score) derived from the machine-learning method yielded a high correlation ($R^2 = 0.953$) for a large training set of 1105 protein–ligand complexes. Teramoto and Fukunishi used a RF classifier to predict the root mean square deviation (rmsd) of a docked conformation from the bioactive conformation [82]. The authors used 100 protein–ligand crystal structures, and produced 100 decoys for each ligand using AutoDock. Descriptors for the RF classification were generated using 11 scoring functions. The RF classifiers predicted which poses were within 2.0 Å rmsd of the X-ray coordinates for 90% of cases, whereas the performance of the individual scoring function varied from 26% to 76%.

In 2005, Springer and coworkers [83] presented PostDOCK, a post-processing filter to distinguish true binding protein–ligand complexes from docking artifacts generated by the popular docking program DOCK 4.0.1 [84]. PostDOCK uses biochemical descriptors to characterize the protein–ligand interaction, including vdW and electrostatic terms from the DOCK scoring function, solvent accessible surface area (SASA) terms, and hydrogen bonding, metal binding, lipophilic, and rotatable bond terms from the ChemScore scoring function. The authors used a RF classifier to separate the binding and the nonbinding ligands from a test set of 44 structurally diverse protein targets, and showed that PostDOCK was able to outperform both the DOCK and ChemScore scoring functions.

Furthermore, Sato and coworkers [85] found that SVM, ANN, and RF models could outperform GlideScore when at least five crystal structures [protein kinase A (PKA), Src, cathepsin K, carbonic anhydrase II, and HIV-1 protease] were used for model building. SVM produced peak performance models using 20 crystal structures, whereas ANN models depended on the choice of complexes. The authors also looked at screening efficiencies of machine learning-derived models for targets where only a few crystal structures are currently available and enriched these training sets with docked poses of active compounds. In this scenario, the SVM models did not show significant learning effects, whereas RF models performance was improved dramatically, because RF is known as a statistical method robust against data with noise (e.g., incorrect docked poses). Thus, SVM should be a method of choice for training sets with reliable structures, and RF, where noise is expected.

Naïve Bayesian classifier

Naïve Bayesian classifiers are frequently used in chemoinformatics both alongside or compared against other classifiers, generally for predicting biological rather than physicochemical properties. Practical applications of these methods have been carried out not only in the VS field, but also in other areas, such as the prediction of the toxicity of the compound [86], phospholipidosis mechanism [87], and protein target and bioactivity classification for drug-like molecules [88,89]. It is in principle possible to use naïve Bayesian classifiers for regression [90], but this is rarely seen in chemoinformatics.

Bayesian methods are based on Bayes' theorem, which gives a mathematical framework for describing the probability of an event that might have been the result of any of two or more causes [91] (Eqn. (1)):

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \tag{1}$$

This equation describes the probability P for state A existing for a given state B. To calculate the probability, Bayes used the probability of B existing given that A exists, multiplied by the probability that A exists, and normalized by the probability that B exists. This admittedly complicated explanation can be interpreted as follows: for an existing state B, what is the probability that state B is caused by state A? The importance of this theorem is that probabilities can be derived without specified knowledge about $P(A/B)$, if information about $P(B/A)$, $P(A)$, and $P(B)$ is available. The essence of the Bayesian approach is to provide a mathematical rule explaining how a hypothesis changes in light of new evidence [92]. In a Bayesian analysis, a set of observations should be seen as something that changes opinion. In other words, Bayesian theory allows scientists to combine new data with their existing knowledge or expertise. A Bayesian method can be used to model the dependencies between variables that directly influence each other, which are usually few. The rest of the variables are assumed conditionally independent.

Although the Bayesian idea has been used for many years, its popularity as a tool within drug discovery and structure–activity analysis is only recent. Bayesian classifiers [93] are increasingly being used given their versatility, robustness, and ease of use. In LBVS, Bayesian modeling methods are applied to predict the probability that a compound represented by a descriptor vector is active [i.e., the probability of activity $P(A/B)$ given descriptor representation B]. From known active (A) and inactive (Z) training compounds, the conditional probability distributions $P(B/A)$ and $P(B/Z)$ given representation B are estimated, respectively. Therefore, Bayesian classifiers are also well suited for ranking of compound databases with respect to probability of activity. The biggest weakness is that the naïve Bayesian model is inappropriate in cases where there are strong conditional dependencies between variables. However, there are a surprisingly large number of cases in which it does well, partly because the classifications made can still be optimal, even if the probability estimations are inaccurate because of feature dependence.

Another Bayesian approach adapted for LBVS is the binary kernel discrimination [94], which makes use of binary fingerprint

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

representations and, unlike Bayesian classification, utilizes a Parzen-window technique to evaluate the joint distributions, thus making no assumption of independence of individual features. However, there has been a clear trend to utilize not individual Bayesian classifiers, but combined multitarget classifiers or Bayesian networks [95], which significantly improve the classification accuracy. Bayesian networks are directed acyclic graphs, in which each node is annotated with quantitative probability information. They are constructed by selecting a set of variables that define the nodes of the network. The nodes are connected via directed links that indicate their inheritance, and each node has a conditional probability distribution that quantifies the effect of the parents on the node. The graphics in Fig. 3a shows a naïve Bayesian classifier, in which the arrows point from the label *Y* to the sample space *X*, indicating assumption of knowledge of the sample distribution under the label. Furthermore, absence of arcs among all random variables indicates that all random variables are mutually independent given the class (conditional independence). The dependencies between attributes, which are missing in naïve Bayesian classifiers, are added in the Bayesian network shown in Fig. 3a.

A recent development of Bayes' methods applied to LBVS is the Bayesian model averaging, a technique recently introduced in the machine-learning world [96] and that has now also found application in VS in a paper published by Angelopoulos *et al.* [97]. In this study, the authors compared Bayesian model averaging to SVM and ANN for the prediction of protein pyruvate kinase activity using DRAGON descriptors. Bayesian models were averaged over an ensemble of compound classification trees, and it was found that the resulting models were interpretable, in addition to showing the performance was at least as good if not better than SVM and ANN. Furthermore, Abdo *et al.* [98] introduced a novel similarity-based VS approach based on a Bayesian inference network (BIN), where the features carry different statistical weights, with features that are statistically less relevant being deprioritized. In this study, retrieval of active compounds from each of 12 Q2 activity data sets derived from the MDDR database was increased by 2–4% in absolute terms (or approximately 8–10% in relative terms) using a variety of circular count fingerprint-based methods, compared with the benchmark Tanimoto coefficient. This result still suggests the importance of considering mutual dependencies of features in the VS task. Furthermore, molecular similarity-based clustering has been integrated with Bayesian models of bioactivity to include activity information in the structural organization of biological screening data [99].

k-Nearest neighbors

The *k*-NN algorithm is a simple and intuitive method to predict the class [100], property [101], or rank [102] of a molecule based on nearest training examples in the feature space. *k*-NN is a kind of instance-based learning or lazy learning, where the function is only approximated locally and all calculations are deferred until classification (Fig. 4). *k*-NN can also be used for regression. It is one of the simplest machine-learning algorithms. A molecule is classified by a majority vote of its neighbors, with the molecule being assigned to the class most common among its *k* nearest neighbors. *k* is a positive integer, typically small. If $k = 1$, then the molecule is simply assigned to the class of its nearest neighbor. In binary classification problems, it is helpful to choose *k* to be an odd number to avoid tied votes. The same method can be used for regression by simply assigning the property value of the object to be the average of the values of its *k* nearest neighbors. However, it can be useful to weigh the contributions of the neighbors, such that the nearer neighbors contribute more to the average than the more distant ones; a procedure for doing this was published by Nigsch *et al.* [103]. The neighbors are taken from a set of molecules for which the correct classification (or, in case of regression, the value of the property) is known. This can be regarded as the training set for the algorithm, although no explicit training phase is required. To identify neighbors, the objects are represented by position vectors in the multidimensional feature space. Usually Euclidean distance is adopted, although other distance measures, such as the Manhattan or Mahalanobis distance, could in principle be used instead. The Euclidean distance is the square root of the sum of squares differences between descriptor values, whereas the Manhattan distance, city-block, or Hamming, represents the distance between points in a city road grid, and examines the

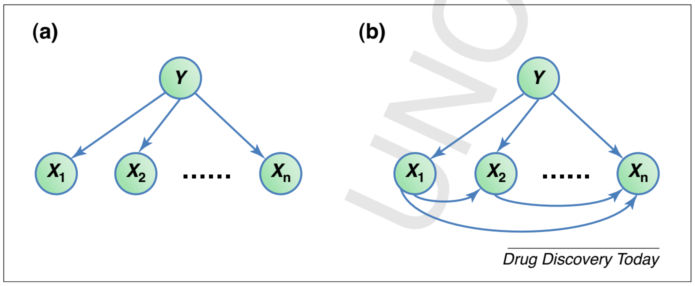

FIGURE 3

Bayesian networks. **(a)** A naïve Bayesian classifier. The arrows point from the label *Y* to the sample space *X*, indicating assumption or knowledge of the sample distribution given the label. Furthermore, absence of arcs between all pairs of random variables indicates that all random variables are mutually independent given the class (conditional independence). **(b)** Bayesian network that captures interattribute dependencies.

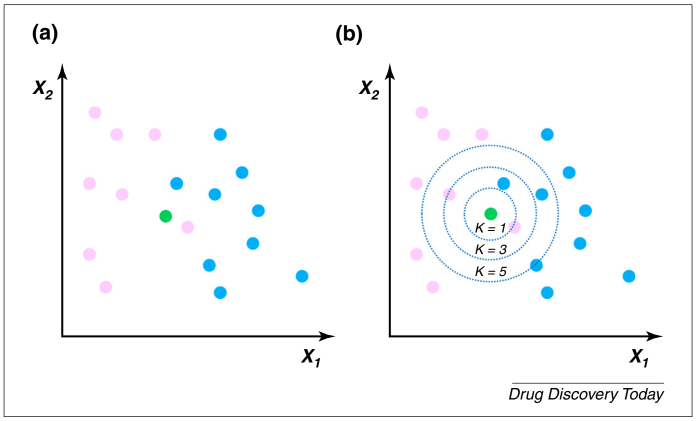

FIGURE 4

k-Nearest neighbors. **(a)** 2D data set showing points belonging to two classes (class 1: pink points; class 2: blue points). The green point is a new data point to be classified. **(b)** The simple nearest-neighbor technique ($k = 1$) classifies the green point as class 1 because it is closest to a pink point (innermost dashed circle). If $k = 3$, it will again be assigned to the pink class because there are two pink points and one blue point inside the inner circle. If $k = i$, it is assigned to the blue class because there are three blue points and two pink points in the outer circle.

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

absolute differences between the coordinates of a pair of feature vectors. Mahalanobis distance takes the distribution of the points (correlations) into account, and is a useful way of determining the similarity of a set of values from an unknown sample to a set of values measured from a collection of known samples. The Mahalanobis distance is the same as the Euclidean distance if the covariance matrix is the identity matrix. A major difficulty is in the construction of a distance measure that reflects a useful metric of similarity. A poor choice of distance metrics might result in meaningless classifications. However, no rationale, except empirical analysis, seems to exist in choosing distance metrics.

The *k*-NN algorithm is sensitive to the local structure of the data. Thus, it is ideal for calculating properties with strong locality, as is the case with protein function prediction [104]. Although intuitive, the *k*-NN approach does have limitations. First, because only *k* neighbors are used to predict a new compound, this method is sensitive to noisy data. A single misclassified training datum could cause a new molecule to be predicted incorrectly. By extension, irrelevant descriptors will likewise lead to spurious predictions. In addition, the predicted value can never be lower or greater than the minimum and maximum activity in the training set.*k*-NN has been used for predicting the activity of anticonvulsants and dopamine D1 antagonists [105], the inhibition of protein kinases [106], the psychoactivity of cannabinoid compounds [107], the activity of steroid, anti-inflammatory and anticancer drugs [108], and of estrogen receptor agonists [109].

Artificial neural networks

ANNs are the most popular and deeply studied techniques in soft computing. In medicinal chemistry, ANNs have been applied in compound classification, QSAR studies, primary VS of compounds, identification of potential drug targets, and localization of structural and functional features of biopolymers [110–113]. ANN techniques have been also used in the fields of robotics, pattern identification, psychology, physics, computer science, biology, and others [113–116].

ANNs arose as an attempt to model brain structure and functioning. However, in addition to any neurological interpretation, they can be considered as a class of general, flexible, nonlinear regression models [117]. The network comprises several simple units, called neurons, arranged in a certain topology, and connected to each other. Neurons are organized into layers. Depending upon their position, layers are called input layers, hidden layers or output layers. An ANN can contain several hidden layers. If, in an ANN, neurons are connected only to those in the following layers, it is called a feed-forward network (Fig. 5a). In this group are included multiplayer perceptrons (MLP), radial basis function (RBF) networks, and Kohonen's self-organizing maps (Kohonen's SOM). By contrast, if recursive or feed-back connections exist Q3 between neurons in different layers, the network is called recurrent (Fig. 5b). Elman and Hopfield networks are classic examples of recurrent topologies. A typical neuron comprises a linear activator followed by a nonlinear inhibiting function (Fig. 5c). The linear activation function yields the sums of weighted inputs plus an independent term, so-called 'bias', *b*. The nonlinear inhibiting function attempts to arrest the signal level of the sum. Step, sigmoid, and hyperbolic tangent functions are the most common functions used as inhibitors (Fig. 5d). Sometimes, purely linear functions are also used for this purpose, especially in output layers. The process of adjusting weights and biases, from supplied data, is called 'training' and the used data, the 'training set'. The process of training an ANN can be broadly classified into two categories: (i) supervised learning, which requires using both the input and the target values for each sample in the training set [111,112]. Tasks that fall within this paradigm are pattern recognition, classification (clustering), function approximation, and prediction. The most common algorithm in this group is the back-propagation, used in the MLP, but it also includes most of the training methods for recurrent networks, time delay networks, and RBF networks; and (ii) unsupervised learning, which is used when the target pattern is not completely known. It includes the methods based on the adaptive resonance theory (ART) and SOM. Tasks that fall within this paradigm are general estimation of problems such as pattern recognition, clustering, estimation of distributions, compression, and filtering.

Back-propagation, which is applied to MLPs, is the most popular and well-studied training algorithm [117]. It is a gradient-descendent method that minimizes the mean-square error of the difference between the network outputs and the targets in the training set.

Nonlinear function approximation is the most important application of multilayer neural networks. It has been proved that a two-layer neural network can approximate any continuous function, within any arbitrary pre-established error, provided that it has a sufficient number of neurons in the hidden layer. This is the so-called 'universal approximation property'. A general and sometimes problematic feature of ANN simulations is that the resulting classification models can usually not be interpreted or explained in physical or chemical terms (a situation often referred to as the 'black box' character of ANNs). By contrast, a major advantage of ANNs is their ability to capture and model nonlinear relations.

Kohonen's SOMs and counterpropagation ANNs

Kohonen's SOMs comprise connected nodes that have an associated vector that corresponds to the input data (i.e., the molecular descriptors) in the map. A simple Kohonen's SOM network is presented in Fig. 6a. As can be seen, the map is an ordered array of neurons. In the example, the map is a rectangle. However, the map can be a line, a circle, a hexagonal structure, or a multidimensional structure of any desired shape. Once a map is constructed, it must be trained to group similar items together, a process called 'clustering'. The SOM is trained using a combination of neighborhood size, neighborhood up-date parameters, and a weight-change parameter. In molecular clustering, descriptor vectors are calculated for test molecules and SOM nodes are assigned corresponding vectors, initially with random values. Then, each test molecule is mapped to the node having the smallest distance to its descriptor vector in chemical space. The neuron closest in distance to the input is declared the winner (Fig. 6b). During the learning phase, vectors of machining nodes and connected neighboring nodes are changed and made more similar to the one of the test molecule. This creates groups of similar nodes that match test molecules having similar descriptor vectors. The learning process continues by gradually reducing the connection weights and value adjustments of neighboring nodes. These calculations generate larger numbers of groups of similar nodes but reduce group size,

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

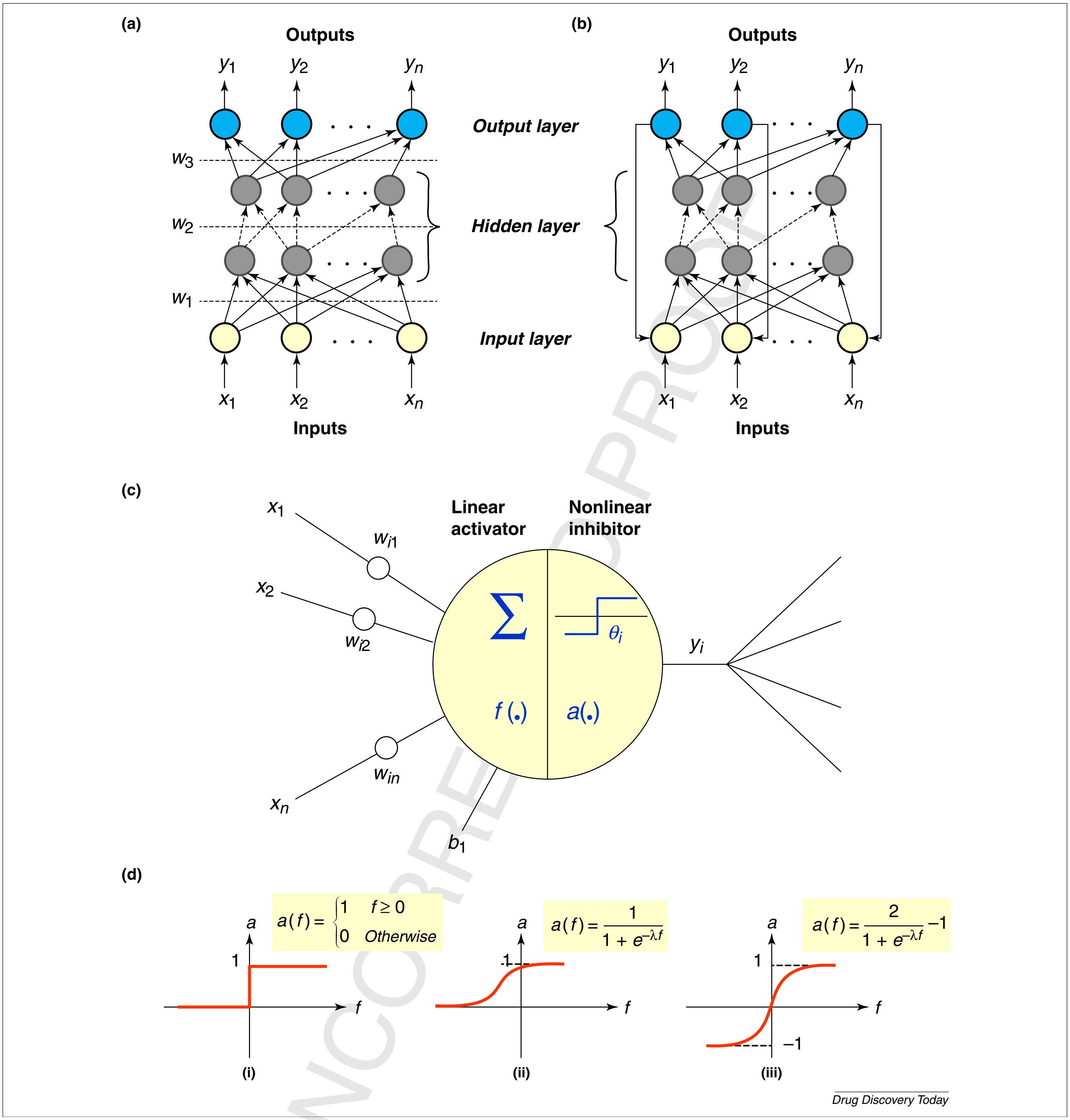

FIGURE 5

Q8 Artificial neural networks. (a) Example of a multilayer feed-forward neural network. For the *i*th layer of links, the symbols $w_{(i)}$, $x_{(i)}$, and $y_{(i)}$ represent a vector of weights between the layers, inputs of nodes at one layer, and output at the output layer, respectively. **(b)** Example of a recurrent neural network, which contains feedback from the outputs to the inputs and its outputs are determined by the current inputs and by the preceding outputs. When organized in layers, there are interconnections between neurons in the same layer and between nonconsecutive layers. **(c)** Logical scheme of a neuron as a perceptron. w_{ij} are the efficacies of synapses coming into neuron *i*, represented by the large circle. x_j are 1–0 variables representing the arrival or non-arrival of a spike along the presynaptic axon connecting neuron *w* to *i*. Integration function *f*(.) is the postsynaptic potential and activation function *a*(.) is the decision function of the neuron. If the neuron will (will not) fire, y_i will take the value 1 (0). The neuronal model also includes an externally applied bias, denoted b_1, which has the effect of increasing or decreasing the net input of the activation function, depending on whether it is positive or negative, respectively. **(d)** Typical inhibiting functions: (i) step, (ii) sigmoid, and (iii) hyperbolic tangents.

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

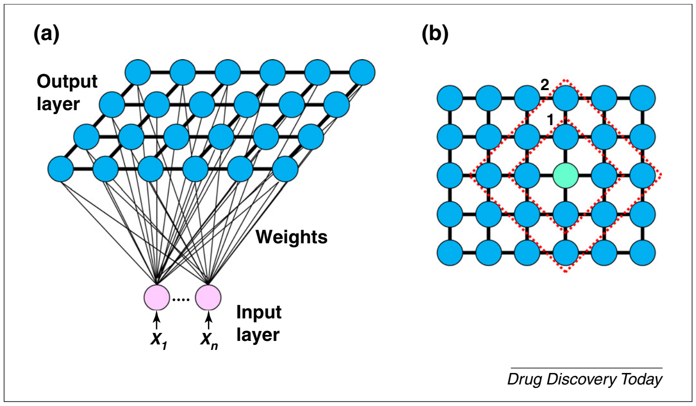

FIGURE 6

Q9 Kohonen's SOM networks. (a) Topology of a simple Kohonen's 2D SOM. It comprises two layers: an input layer and an output layer. Each input layer neuron (pink circle) has a feed-forward connection to each output layer neuron (blue circle). The output neurons that win the competition are called 'winning neurons', where a winning neuron is chosen by selecting a neuron whose weight vector has a minimum Euclidean distance (or maximum similarity) from the input vector. **(b)** A 5 × 6 Kohonen's Layer with two neighborhood sizes around the winning neuron, identified as a green circle.

which increases the resolution of the molecular classification scheme. SOMs ultimately assign similar molecules to regions of similar nodes and additional compounds can be mapped based on their descriptor vectors. Thus, SOMs can be used as a clustering tool. A key concept in training SOM is the neighborhood N_k around a winning neuron, k, which is the collection of all nodes with the same radial distance. Fig. 6b gives an example of neighborhood nodes for a 5 × 6 Kohonen's layer at radius of 1 and 2. The main advantage of the SOM, in comparison to other projection methods, is that the algorithm is simple, straightforward to implement, and fast to compute.

Given that SOMs are capable of projecting compound distributions in high-dimensional descriptor spaces on 2D arrays of nodes, this methodology is also useful as a dimension reduction technique. SOMs have also been adapted for LBVS. For example, Hristozov *et al*. [118] used a Kohonen's SOM as a model to identify and discard compounds that are unlikely to have a given biological activity. In addition, SOMs have been used to concentrate LBVS calculations on the structural proximity of reference compounds [119]. Unlike other machine-learning methods, SOM built on a relatively small but diverse training set might be an effective LBVS enhancer of a much larger, independent database. Moreover, the use of oversized SOM training sets was shown to be not only unhelpful to further increase map performance, but also often detrimental [119]. Empirical evidence shows that the quality of many machine-learning algorithms improves with the size of the training data, and that a simple algorithm is likely to outperform a more complex one if it gets more training data, but this is not always the case.

Counterpropagation ANNs (CP-ANNs) are an extension of Kohonen's SOMs for classification models [120] that, in addition to the Kohonen's layer, contain a set of output layers, called 'Grosberg layers'. A CP-ANN comprises [121] two layers: the input layer, which is a Kohonen's network, and an associated output layer containing the values of the properties to be predicted. The number of Grosberg layers is equal to the number of classes. The learning in CP-ANN has an additional step. The first step runs in the input layer and is the same as in SOM (i.e., the objects are arranged into the map accordingly to similarity relations among them). In the second step of learning, the positions of objects are projected from the input layer to the output layer and the weights there are modified to become equal to corresponding output values. The reader can find more details about architecture and learning strategy of SOM and CP-ANN in many textbooks and articles [122–124]. For instance, Kohonen's maps and CP-ANNs have been successfully applied in LBVS for the prediction and identification of novel amyloid β-A4 protein (ABPP) inhibitors [125]. In such work, the inhibitory activity of a series of 62 N-phenylanthranilic acids using Kohonen's maps and CP-ANNs was explored. The effects of various structural modifications on biological activity were investigated and novel structures then designed using the in silico model.

Several variations and extensions of Kohonen's original SOM algorithm have been published and applied to drug discovery [126]. Such developments include self-organizing networks with an adapting grid size [127], cascaded SOMs [128], and hybrid neural networks [129,130]. These systems might provide alternative approaches to VS, although their practical usefulness and applicability to hit and lead finding still need to be rigorously assessed.

Concluding remarks and future directions

LBVS techniques are widely used for hit identification. The methodological spectrum of these techniques is wide, ranging from rather simplistic fingerprint-based approaches to highly complex machine-learning methods. In this article, I have emphasized the theoretical foundations and exemplary recent developments of five advanced machine-learning approaches that are commonly used in chemoinformatics and in drug discovery: SVM, DT, *k*-NN, naïve Bayesian methods, and AANs. These tools have become popular because they are easily accessible both as open source and commercial distributions [131], statistically consistent, computationally efficient, but simple to implement and interpret. Multiple variant open-source and commercial algorithms can be used to implement each approach, and specialized method-specific software is available to support more flexible configurations (Table 1). Machine-learning algorithms also can be implemented in a variety of programming languages. Data-mining software enables users to implement versions of these algorithms via point-and-click graphic user interfaces. However, these algorithms can also be written and executed using packages such as R, Matlab, and Octave. It is important that users understand how to apply each unique method properly to produce optimal models and avoid spurious results.

Here, I have also evaluated critically the opportunities and limitations of these methods, with a particular focus on their practical relevance and value in LBVS. Table 2 lists common classification methods and provides a comparison of their performance, computational cost, and other factors. Taken together, SVMs and Bayesian methods currently dominate the LBVS field. However, there is no single approach that is superior, and LBVS success strongly depends on the size and diversity of the training data set, the linearity of the chemical problem to be solved, the correlation of the descriptor set available, and the importance of nonlocal information. For linear problems, a simple multiple

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

TABLE 1

Examples of available machine-learning programs that implement the methods discussed in this review.

Software	Learning algorithms	License	Website
Matlab	SVM, ANN, Naïve Bayes, DT, and *k*-NN	Commercial	http://www.mathworks.com/products/matlab/
TreeNet	RF	Commercial	http://www.salford-systems.com/products/treenet
R	RF,SVM, Naïve Bayesian, and ANN	Open source	http://www.r-project.org/
libSVM	SVM		http://www.csie.ntu.edu.tw/~cjlin/libsvm
Orange	RF, SVM, and Naïve Bayesian	Open source	http://www.ailab.si/orange/
RapidMiner	SVM, RF, Naïve Bayes, DT, ANN, and *k*-NN	Open source	http://rapid-i.com/
Weka	RF, SVM, and Naïve Bayes	Open source	http://www.cs.waikato.ac.nz/ml/weka/
Knime	DT, Naïve Bayes, and SVM	Open source	http://www.knime.org/
AZOrange	RT, SVM, ANN, and RF	Open source	http://www.jcheminf.com/content/3/1/28
SciTegic Pipeline Pilot	SVM, Naïve Bayes, and DT	Commercial	http://www.accelrys.com
Tanagra	SVM, RF, Naïve Bayes, and DT	Open source	http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html
Elki	*k*-NN	Open source	http://elki.dbs.ifi.lmu.de/

linear regression approach proved to be superior over more complex machine-learning approaches [132]. For nonlocal problems, some studies have reported that SVM outperforms RF [133,134], whereas others suggest that RF and SVM give similar prediction quality [135,136]. Where local data structures are not best summarized linearly (yet are important to the interpretation of the experimental results), a nonlinear method, such as *k*-NN, can be more appropriate. Although benchmark studies have revealed

TABLE 2

Comparison of various classification algorithms.

Method	Classification error	Computational cost	Memory requirements	Difficult to implement	Online?	Easy to interpret?	Advantages	Disadvantages
SVM	Low	Medium	Low	Medium	Yes	No	Does not make any assumption about type of relation between target property and molecular descriptors; low risk of overfitting; able to provide expected classification accuracies for individual compounds	Training speed can be slow with large training sets; predominantly binary classification only
DT	Medium	Medium	Medium	Low	No	Yes	Does not make any assumption about type of relation between target property and molecular descriptors; fast classification speed; multiclass classification	Might have overfitting when training set is small and number of molecular descriptors is large; ranks molecular descriptors using information gain, which might not be the best for some problems
Naïve Bayesian	Low	Low	Low	High	Yes	Yes	Fast to train (single scan); fast to classify; not sensitive to irrelevant features; handles real and discrete data	Assumes independence of features
***k*-NN**	Medium–low	High	High	Low	No	No	Does not make any assumption about type of relation between target property and molecular descriptors; fast training time; multiclass classification	Classification speed can be slow with large training sets; classification is sensitive to type of distance measures used
ANN	Low	Medium	Low	High	Yes	No	Does not make any assumption about type of relation between target property and molecular descriptors	Difficult to design an optimal architecture; risk of overfitting

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

some overall winners [137], the choice of a learning algorithm must be made in light of the characteristics of a given prediction problem, data source, and prediction performance [138,139].

In addition to the methods discussed above, several new algorithms and approaches are continually under development in the LBVS arena. A recent tendency is to assemble different classifiers and to construct a metaclassifier, which combines the predictions of the base classifiers [140]. Furthermore, Swamidass *et al.* [141] introduced the Influence Relevance Voter (IRV), a new exemplary method that uses ANN architecture to learn how to best integrate information from the nearest structural neighbors contained in the training set. The IRV tunes itself to each data set by a simple gradient descent-learning procedure and produces continuous outputs that can be interpreted probabilistically and used to rank all the compounds. The IRV performance was shown to be at least comparable to other machine-learning methods, such as SVMs. Moreover, the IRV approach has several other important advantages over SVMs and other methods: it is trained much more quickly, it provides a framework that easily allows the incorporation of additional information, beyond the chemical structures; and its predictions are interpretable.

With the exponential growth of data sets over the past decade and the increased use of medical data-mining applications, the data-mining community has moved into high-performance settings, including accelerators that are characterized as hardware that perform certain computations faster than the computer processing units (CPU). Examples of such accelerators include Field Programmable Gate Arrays, the Cell Broadband Engine Architecture (CBEA), and Graphical Processing Units (GPUs). Liao *et al.* [142] demonstrated the power of GPU acceleration of molecular similarity calculations for SVMs.

In the future, we are likely to see more focus on the development of machine learnings that reflect domain knowledge and utilize output from several lower-level algorithms. Given that the current trend toward wider, faster, and deeper surveys in the LBVS field is expected to accelerate, the importance of machine-learning tools will continue to grow.

Acknowledgments

I dedicate this work to the memory of my uncle, Francesco Lavecchia, who passed away on 1 May 2014, while this article was being prepared. His inquisitive mind, brightness, open mind, work ethics, and dedication to the medical profession will be greatly missed. This research was supported by the Ministero dell'Istruzione, Università e Ricerca (MIUR-PRIN 2010-2011, prot. 2010W7YRLZ_003).

Q4

References

1 Weaver, D.C. (2004) Applying data mining techniques to library design, lead generation and lead optimization. *Curr. Opin. Chem. Biol.* 8, 264–270

2 Yang, Y. *et al.* (2009) Target discovery from data mining approaches. *Drug Discov. Today* 14, 147–154

3 Campbell, S.J. *et al.* (2010) Visualizing the drug target landscape. *Drug Discov. Today* 15, 3–15

4 Geppert, H. *et al.* (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

5 Hasan, S. (2012) Network analysis has diverse roles in drug discovery. *Drug Discov. Today* 17, 869–874

6 Reddy, S. *et al.* (2007) Virtual screening in drug discovery: a computational perspective. *Curr. Prot. Pept. Sci.* 8, 329–351

7 Freitas, R.F. *et al.* (2008) 2D QSAR and similarity studies on cruzain inhibitors aimed at improving selectivity over cathepsin L. *Bioorg. Med. Chem.* 16, 838–853

8 Lavecchia, A. and Di Giovanni, C. (2013) Virtual screening strategies in drug discovery: a critical review. *Curr. Med. Chem.* 20, 2839–2860

9 Kubinyi, H. (1998) Similarity and dissimilarity: a medicinal chemist's view. *Persp. Drug Discov. Des.* 11, 225–252

10 Cruz-Monteagudo, M. *et al.* (2014) Activity cliffs in drug discovery: Dr. Jekyll or Mr. Hyde? *Drug Discov. Today* http://dx.doi.org/10.1016/j.drudis.2014.02.003

11 Maggiora, G.M. (2006) On outliers and activity cliffs – why QSAR often disappoints. *J. Chem. Inf. Model.* 46 1535–1535

12 Stumpfe, D. and Bajorath, J. (2012) Exploring activity cliffs in medicinal chemistry. *J. Med. Chem.* 55, 2932–2942

13 Eckert, H. and Bajorath, J. (2007) Molecular similarity analysis in virtual screening: foundations, limitations and novel approaches. *Drug Discov. Today* 12, 225–233

14 Bajorath, J. (2002) Integration of virtual and high-throughput screening. *Nat. Rev. Drug Discov.* 1, 882–894

15 Willett, P. (2005) Searching techniques for databases of two- and three-dimensional chemical structures. *J. Med. Chem.* 48, 4183–4199

16 Willett, P. (2006) Similarity-based virtual screening using 2D fingerprints. *Drug Discov. Today* 11, 1046–1053

17 Mason, J.S. *et al.* (2001) 3-D pharmacophores in drug discovery. *Curr. Pharm. Des.* 7, 567–597

18 Gillet, V.J. *et al.* (2003) Similarity searching using reduced graphs. *J. Chem. Inf. Comput. Sci.* 43, 338–345

19 Cramer, R.D. *et al.* (1999) Prospective identification of biologically active structures by topomer shape similarity searching. *J. Med. Chem.* 42, 3919–3933

20 Hawkins, P.C.D. *et al.* (2007) Comparison of shape-matching and docking as virtual screening tools. *J. Med. Chem.* 50, 74–82

21 Totrov, M. (2008) Atomic property fields: generalized 3D pharmacophoric potential for automated ligand superposition, pharmacophore elucidation and 3D QSAR. *Chem. Biol. Drug. Des.* 71, 15–27

22 Kufareva, I. *et al.* (2012) Compound activity prediction using models of binding pockets or ligand properties in 3D. *Curr. Top. Med. Chem.* 12, 1869–1882

23 Mitchell, J.B.O. (2014) Machine learning methods in chemoinformatics. *WIREs Comput. Mol. Sci.* 4, 468–481

24 Melville, J.L. *et al.* (2009) Machine learning in virtual screening. *Comb. Chem. High Throughput Screen.* 12, 332–343

25 Bajorath, J. (2001) Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *J. Chem. Inf. Comput.* 41, 233–245

26 Schnur, D. *et al.* (2004) Approaches to target class combinatorial library design. chemoinformatics. *Methods Mol. Biol.* 275, 355–378

27 Cramer, R.D. *et al.* (1974) Substructural analysis. A novel approach to the problem of drug design. *J. Med. Chem.* 17, 533–535

28 Duda, R.O. *et al.* (2000) *Pattern Classification.* Wiley Interscience

29 Hand, D. *et al.* (2001) *Principles of Data Mining.* MIT Press

30 Vapnik, V.N. (2000) *The Nature of Statistical Learning Theory.* Springer

31 Vapnik, V.N. (1998) *Statistical Learning Theory.* Wiley

32 Byvatov, E. (2003) Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *J. Chem. Inf. Comput. Sci.* 43, 1882–1889

33 Zernov, V.V. *et al.* (2003) Drug discovery using support vector machines. The case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. *J. Chem. Inf. Comput. Sci.* 43, 2048–2056

34 Warmuth, M.K. *et al.* (2003) Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.* 43, 667–673

35 Jorissen, R.N. and Gilson, M.K. (2005) Virtual screening of molecular databases using a support vector machine. *J. Chem. Inf. Model.* 45, 549–561

36 Podolyan, Y. *et al.* (2010) Assessing synthetic accessibility of chemical compounds using machine learning methods. *J. Chem. Inf. Model.* 50, 979–991

37 Cheng, T. *et al.* (2011) Binary classification of aqueous solubility using support vector machines with reduction and recombination feature selection. *J. Chem. Inf. Model.* 51, 229–236

38 Camps-Valls, G. and Bruzzone, L. (2005) Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 43, 1351–1362

39 Hinselmann, G. *et al.* (2011) Large-scale learning of structure–activity relationships using a linear support vector machine and problem-specific metrics. *J. Chem. Inf. Model.* 51, 203–213

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

40 Foody, G.M. and Mathur, A. (2006) The use of small training sets containing mixed pixels for accurate hard image classification: training on mixed spectral responses for classification by a SVM. *Remote Sens. Environ.* 103, 179–189

41 Geppert, H. *et al.* (2008) Support-vector-machine-based ranking significantly improves the effectiveness of similarity searching using 2D fingerprints and multiple reference compounds. *J. Chem. Inf. Model.* 48, 742–746

42 Agarwal, S. *et al.* (2010) Ranking chemical structures for drug discovery: a new machine learning approach. *J. Chem. Inf. Model.* 50, 716–731

43 Rathke, F. *et al.* (2011) StructRank: a new approach for ligand-based virtual screening. *J. Chem. Inf. Model.* 51, 83–92

44 Jacob, L. *et al.* (2008) Virtual screening of GPCRs: an in silico chemogenomics approach. *BMC Bioinformatics* 9, 363

45 Jacob, L. and Vert, J.-P. (2008) Protein–ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics* 24, 2149–2156

46 Vert, J.-P. and Jacob, L. (2008) Machine learning for in silico virtual screening and chemical genomics: new strategies. *Comb. Chem. High. Throughput Screen.* 11, 677–685

47 Gärtner, T. *et al.* (2003) On graph kernels: hardness results and efficient alternatives. In *Learning Theory and Kernel Machines* (Schölkopf, B. and Warmuth, M.K., eds), pp. 129–143, Springer-Verlag

48 Kashima, H. *et al.* (2003) Marginalized kernels between labeled graphs. In Q5 *Proceedings of the 21st International Conference on Machine Learning (ICML 2003)*, AAAI Press. pp. 320–321

49 Ralaivola, L. *et al.* (2005) Graph kernels for chemical informatics. *Neural Netw.* 18, 1093–1110

50 Mahé, P. *et al.* (2006) The pharmacophore kernel for virtual screening with support vector machines. *J. Chem. Inf. Model.* 46, 2003–2014

51 Azencott, C.-A. *et al.* (2007) One-to four-dimensional kernels for virtual screening and the prediction of physical, chemical, and biological properties. *J. Chem. Inf. Model.* 47, 965–974

52 Erhan, D. *et al.* (2006) Collaborative filtering on a family of biological targets. *J. Chem. Inf. Model.* 46, 626–635

53 Wassermann, A.M. *et al.* (2009) Ligand prediction for orphan targets using support vector machines and various target-ligand kernels is dominated by nearest neighbor effects. *J. Chem. Inf. Model.* 49, 2155–2167

54 Meslamani, J. and Rognan, D. (2011) Enhancing the accuracy of chemogenomic models with a three-dimensional binding site kernel. *J. Chem. Inf. Model.* 5, 1593–1603

55 Heikamp, K. *et al.* (2012) Prediction of activity cliffs using support vector machines. *J. Chem. Inf. Model.* 52, 2354–2365

56 Plewczynski, D. (2011) Brainstorming: weighted voting prediction of inhibitors for protein targets. *J. Mol. Model.* 17, 2133–2141

57 Cheng, F. *et al.* (2011) Classification of cytochrome P450 inhibitors and noninhibitors using combined classifiers. *J. Chem. Inf. Model.* 51, 99–1011

58 Xie, Q.-Q. *et al.* (2011) Combined SVM-based and docking-based virtual screening for retrieving novel inhibitors of c-Met. *Eur. J. Med. Chem.* 46, 3675–3680

59 Meslamani, J. *et al.* (2013) Computational profiling of bioactive compounds using a target-dependent composite workflow. *J. Chem. Inf. Model.* 53, 2322–2333

60 Tipping, M. (2001) Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* 1, 211–244

61 Lowe, R. *et al.* (2011) Classifying molecules using a sparse probabilistic kernel binary classifier. *J. Chem. Inf. Model.* 51, 1539–1544

62 Klekota, J. and Roth, F.P. (2008) Chemical substructures that enrich for biological activity. *Bioinformatics* 24, 2518–2525

63 Schneider, N. *et al.* (2008) Gradual in silico filtering for druglike substances. *J. Chem. Inf. Model.* 48, 613–628

64 Hou, T. *et al.* (2007) ADME evaluation in drug discovery. 8. The prediction of human intestinal absorption by a support vector machine. *J. Chem. Inf. Model.* 47, 2408–2415

65 Deconinck, E. *et al.* (2006) Classification tree models for the prediction of blood–brain barrier passage of drugs. *J. Chem. Inf. Model.* 46, 1410–1419

66 Gleeson, M.P. *et al.* (2006) In silico human and rat Vss quantitative structure–activity relationship models. *J. Med. Chem.* 49, 1953–1963

67 Lamanna, C. *et al.* (2008) Straightforward recursive partitioning model for discarding insoluble compounds in the drug discovery process. *J. Med. Chem.* 51, 2891–2897

68 de Cerqueira Lima, P. *et al.* (2006) Combinatorial QSAR modeling of P-glycoprotein substrates. *J. Chem. Inf. Model.* 46, 1245–1254

69 Mente, S.R. *et al.* (2005) A recursive-partitioning model for blood–brain barrier permeation. *J. Comput. Aided Mol. Des.* 19, 465–481

70 Sakiyama, Y. *et al.* (2008) Predicting human liver microsomal stability with machine learning techniques. *J. Mol. Graph. Model.* 26, 907–915

71 Riddle, P. *et al.* (1994) Representation design and brute-force induction in a Boeing manufacturing design. *Appl. Artif. Intell.* 8, 125–147

72 Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning.* Morgan Kaufmann

73 Raileanu, L.E. and Stoffel, K. (2004) Theoretical comparison between the Gini Index and Information Gain criteria. *Ann. Math. Artif. Intell.* 41, 77–93

74 Ho, T.K. (1998) The random subspace method for constructing decision forests. *ITPAM* 20, 832–844

75 Breiman, L. (1996) Bagging predictors. *Mach. Learn.* 24, 123–140

76 Freund, Y. and Schapire, R.E. (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory* (Vitányi, P.M.B., ed.), In pp. 23–37, Springer-Verlag

77 Breiman, L. (1996) Stacked regressions. *Mach. Learn.* 24, 49–64

78 Breiman, L. (2001) Random forests. *Mach. Learn.* 45, 5–32

79 Svetnik, V. *et al.* (2003) Random forest: a classification and regression tool for compound classification and QSAR modelling. *J. Chem. Inf. Comput. Sci.* 43, 1947–1958

80 Tong, W.D. *et al.* (2003) Decision forest: combining the predictions of multiple independent decision tree models. *J. Chem. Inf. Comput. Sci.* 43, 525–531

81 Ballester, P.J. and Mitchell, J.B.O. (2010) A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* 26, 1169–1175

82 Teramoto, R. and Fukunishi, H. (2007) Supervised consensus scoring for docking and virtual screening. *J. Chem. Inf. Model.* 47, 526–534

83 Springer, C. *et al.* (2005) PostDOCK: a structural, empirical approach to scoring protein ligand complexes. *J. Med. Chem.* 48, 6821–6831

84 Shoichet, B.K. and Kuntz, I.D. (1993) Matching chemistry and shape in molecular docking. *Protein Eng.* 6, 723–732

85 Sato, T. *et al.* (2010) Combining machine learning and pharmacophore-based interaction fingerprint for in silico screening. *J. Chem. Inf. Model.* 50, 170–185

86 von Korff, M. and Sander, T. (2006) Toxicity-indicating structural patterns. *J. Chem. Inf. Model.* 46, 536–544

87 Lowe, R. *et al.* (2012) Predicting the mechanism of phospholipidosis. *J. Cheminformatics* 4, 2

88 Koutsoukas, A. *et al.* (2013) In silico target predictions: defining a benchmarking dataset and comparison of performance of the multiclass Naïve Bayes and Parzen-Rosenblatt Window. *J. Chem. Inf. Model.* 53, 1957–1966

89 Nigsch, F. *et al.* (2008) Ligand-target prediction using Winnow and naïve Bayesian algorithms and the implications of overall performance statistics. *J. Chem. Inf. Model.* 48, 2313–2325

90 Frank, E. *et al.* (2000) Technical note: naïve Bayes for regression. *Mach. Learn.* 41, 5–25

91 Jensen, F.V. (2001) *Bayesian Networks and Decision Graphs.* Springer

92 Dempster, A.P. (1968) A generalization of Bayesian Inference. *J. Royal Stat. Soc. B* 30, 205–247

93 Watson, P. (2008) Naïve Bayes classification using 2D pharmacophore feature triplet vectors. *J. Chem. Inf. Model.* 48, 166–178

94 Willett, P. *et al.* (2007) Prediction of ion channel activity using binary kernel discrimination. *J. Chem. Inf. Model.* 47, 1961–1966

95 Abdo, A. *et al.* (2010) Ligand-based virtual screening using Bayesian networks. *J. Chem. Inf. Model.* 50, 1012–1020

96 Wasserman, L. (2000) Bayesian model selection and model averaging. *J. Math. Psychol.* 44, 92–107

97 Angelopoulos, N. *et al.* (2009) Bayesian model averaging for ligand discovery. *J. Chem. Inf. Model.* 49, 1547–1557

98 Abdo, A. and Salim, N. (2009) Similarity-based virtual screening with a Bayesian inference network. *ChemMedChem* 4, 210–218

99 Lounkine, E. *et al.* (2011) Activity-aware clustering of high throughput screening data and elucidation of orthogonal structure–activity relationships. *J. Chem. Inf. Model.* 51, 3158–3168

100 Kauffman, G.W. and Jurs, P.C. (2001) *QSAR and k-nearest neighbor* classification analysis of selective cyclooxygenase-2 inhibitors using topologically-based numerical descriptors. *J. Chem. Inf. Comp. Sci.* 41, 1553–1560

101 Konovalov, D.A. *et al.* (2007) Benchmarking of QSAR models for blood–brain barrier permeation. *J. Chem. Inf. Comp. Sci.* 47, 1648–1656

102 Votano, J.R. *et al.* (2004) Three new consensus QSAR models for the prediction of Ames genotoxicity. *Mutagenesis* 19, 365–377

103 Nigsch, F. *et al.* (2006) Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization. *J. Chem. Inf. Model.* 46, 2412–2422

104 De Ferrari, L. *et al.* (2012) EnzML: multi-label prediction of enzyme classes using InterPro signatures. *BMC Bioinf.* 13, 61

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012

105 Itskowitz, P. and Tropsha, A. (2005) k-nearest neighbors QSAR modeling as a variational problem: theory and applications. *J. Chem. Inf. Model.* 45, 777–785

106 Briem, H. and Günther, J. (2005) Classifying ‘‘kinase inhibitor likeness’’ by using machine-learning methods. *Chembiochem* 6, 558–566

107 Honório, K.M. and da Silva, A.B. (2005) A study on the influence of molecular properties in the psychoactivity of cannabinoid compounds. *J. Mol. Model.* 11, 200–209

108 Ajmani, S. *et al.* (2006) Three-dimensional QSAR using the k-nearest neighbor method and its interpretation. *J. Chem. Inf. Model.* 46, 24–31

109 Li, H. *et al.* (2006) Prediction of estrogen receptor agonists and characterization of associated molecular descriptors by statistical learning methods. *J. Mol. Graph. Model.* 25, 313–323

110 Patel, J. and Chaudhari, C. (2005) Introduction to the artificial neural networks and their applications in QSAR studies. *ALTEX* 22, 271

111 Patel, J.L. and Patel, L.D. (2007) Artificial neural networks and their applications in pharmaceutical research. *Pharmabuzz.* 2, 8–17

112 Patel, J.L. and Goyal, R.K. (2007) Applications of artificial neural networks in medical science. *Curr. Clin. Pharmacol.* 2, 217–226

113 Soyguder, S. (2011) Intelligent control based on wavelet decomposition and neural network for predicting of human trajectories with a novel vision-based robotic. *Expert Syst. Appl.* 38, 13994–14000

114 Aitkenhead, M.J. and McDonald, A.J.S. (2006) The state of play in machine/environment interactions. *Artif. Intell. Rev.* 25, 247–276

115 Fogel, G.B. (2008) Computational intelligence approaches for pattern discovery in biological systems. *Brief Bioinform.* 9, 307–316

116 Perlovsky, L.I. (2006) Toward physics of the mind: concepts, emotions, consciousness, and symbols. *Phys. Life. Rev.* 3, 23–55

117 Haykin, S.S. (1999) *Neural Networks: A Comprehensive Foundation.* Prentice Hall

118 Hristozov, D. *et al.* (2007) Ligand-based virtual screening by novelty detection with self-organizing maps. *J. Chem. Inf. Model.* 47, 2044–2062

119 Bonachera, F. *et al.* (2012) Using self-organizing maps to accelerate similarity search. *Bioorg. Med. Chem.* 20, 5396–5409

120 Zupan, J. *et al.* (1995) Neural networks with counter-propagation learning strategy used for modelling. *Chemom. Intell. Lab. Syst.* 27, 175–187

121 Vracko, M. (2005) Kohonen artificial neural network and counter propagation neural network in molecular structure-toxicity studies. *Curr. Comput. Aided Drug Des.* 1, 73–78

122 Hecht-Nielsen, R. (1987) Counterpropagation networks. *Appl. Optics* 26, 4979–4984

123 Zupan, J. and Gasteiger, J. (1999) *Neural Networks in Chemistry and Drug Design.* Wiley-VCH

124 Zupan, J. (2003) Basics of artificial neural network. In *Nature-inspired Methods in Chemometrics: Genetic Algorithms and Artificial Neural Networks* (Leardi, R., ed.), pp. 199–229, Elsevier

125 Afantitis, A. *et al.* (2011) Ligand-based virtual screening procedure for the prediction and the identification of novel beta-amyloid aggregation inhibitors using Kohonen maps and counterpropagation artificial neural networks. *Eur. J. Med. Chem.* 46, 497–508

126 Selzer, P. and Ertl, P. (2006) Applications of self-organizing neural networks in virtual screening and diversity selection. *J. Chem. Inf. Model.* 46, 2319–2323

127 Wu, Z. and Yen, G.G. (2003) A SOM projection technique with the growing structure for visualizing high-dimensional data. *Int. J. Neural Syst.* 13, 353–365

128 Furukawa, T. (2009) SOM of SOMs. *Neural Netw.* 22, 463–478

129 Tetko, I.V. (2002) Associative neural network. *Neural Process. Lett* 16, 187–199

130 Gupta, S. *et al.* (2006) QSAR analysis of phenolic antioxidants using MOLMAP descriptors of local properties. *Bioorg. Med. Chem.* 14, 1199–1206

131 Karthikeyan, M. and Vyas, R. (2014) Machine learning methods in chemoinformatics for drug discovery. In *Practical Chemoinformatics* (Karthikeyan, M. and Vyas, R., eds),
pp. 133–194, Springer

132 Hewitt, M. *et al.* (2009) In silico prediction of aqueous solubility: the solubility challenge. *J. Chem. Inf. Model.* 49, 2572–2587

133 Statnikov, A. *et al.* (2008) A comprehensive comparison of Random Forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics* 9, 319

134 Hughes, L.D. *et al.* (2008) Why are some properties more difficult to predict than others? A study of QSPR models of solubility, melting point, and Log P. *J. Chem. Inf. Model.* 48, 220–232

135 Uriarte, R.D. and de Andres, S.A. (2006) Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7, 3

136 Lowe, R. *et al.* (2010) Predicting phospholipidosis using machine learning. *Mol. Pharm.* 7, 1708–1718

137 Smusz, S. *et al.* (2013) A multidimensional analysis of machine learning methods performance in the classification of bioactive compounds. *Chemom. Intell. Lab. Syst.* 128, 89–100

138 King, R. *et al.* (1995) Statlog: comparison of classification algorithms on large real-world problems. *Appl. Artificial Intell.* 9, 259–287

139 Caruana, R. and Niculescu-Mizil, A. (2006) An empirical comparison of supervised learning algorithms. *ICML 06 Proceedings of the 23rd International Conference on Machine Learning*, ACM. pp. 161–168 Q6

140 Cheng, F. *et al.* (2011) Classification of cytochrome P450 inhibitors and noninhibitors using combined classifiers. *J. Chem. Inf. Model.* 51, 996–1011

141 Swamidass, S.J. *et al.* (2009) Influence relevance voting: an accurate and interpretable virtual high throughput screening method. *J. Chem. Inf. Model.* 49, 756–766

142 Liao, Q. *et al.* (2009) GPU accelerated support vector machines for mining high-throughput screening data. *J. Chem. Inf. Model.* 49, 2718–2725

Please cite this article in press as: Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications, Drug Discov Today (2014), http://dx.doi.org/10.1016/j.drudis.2014.10.012