1

Accessible Machine Learning Approaches for Toxicology

Sean Ekins[1], *Alex M. Clark*[2], *Alexander L. Perryman*[3], *Joel S. Freundlich*[3,4], *Alexandru Korotcov*[5], *and Valery Tkachenko*[6]

[1] *Collaborations Pharmaceuticals, Inc., Raleigh, NC, USA*
[2] *Molecular Materials Informatics, Inc., Montreal, Quebec, Canada*
[3] *Department of Pharmacology & Physiology, New Jersey Medical School, Rutgers University, Newark, NJ, USA*
[4] *Division of Infectious Disease, Department of Medicine and the Ruy V. Lourenço Center for the Study of Emerging and Re-emerging Pathogens, New Jersey Medical School, Rutgers University, Newark, NJ, USA*
[5] *Gaithersburg, MD, USA*
[6] *Rockville, MD, USA*

CHAPTER MENU

1.1 Introduction

Computational approaches have in recent years played an increasingly important role in the drug discovery process within large pharmaceutical firms. Virtual screening of compounds using ligand-based and structure-based methods to predict potency enables more efficient utilization of high throughput screening (HTS) resources, by enriching the set of compounds physically screened with those more likely to yield hits [1–4]. Computation of absorption, distribution, metabolism, excretion, and toxicity (ADME/Tox) properties exploiting statistical techniques greatly reduces the number of expensive assays that must be performed, now making it practical to consider these factors very early in the discovery process to minimize late-stage failures of potent lead compounds that are not drug-like [5–11]. Large pharma have successfully

Computational Toxicology: Risk Assessment for Chemicals, First Edition. Edited by Sean Ekins.

integrated these *in silico* methods into operational practice, validated them, and then realized their benefits, because these firms have (i) expensive commercial software to build models, (ii) large, diverse proprietary datasets based on consistent experimental protocols to train and test the models, and (iii) staff with extensive computational and medicinal chemistry expertise to run the models and interpret the results. Drug discovery efforts centered in universities, foundations, government laboratories, and small biotechnology companies, however, generally lack these three critical resources and, as a result, have yet to exploit the full benefits of *in silico* methods. For close to a decade, we have aimed to used machine learning approaches and have evaluated how we could circumvent these limitations so that others can benefit from current and emerging best industry practices.

The current practice in pharma is to integrate *in silico* predictions into a combined workflow together with *in vitro* assays to find "hits" that can then be reconfirmed and optimized [12]. The incremental cost of a virtual screen is minimal, and the savings compared with a physical screen are magnified if the compound would also need to be synthesized rather than purchased from a vendor. Imagine if the blind hit rate against some library is 1%, and the *in silico* model can pre-filter the library to give an experimental hit rate of 2%, then significant resources are freed up to focus on other promising regions of chemical property space [13]. Our past pharmaceuticals collaborations [14, 15] have suggested that computational approaches are critical to making drug discovery more efficient.

The relatively high cost of *in vivo* and *in vitro* screening of ADME and toxicity properties of molecules has motivated our efforts to develop *in silico* methods to filter and select a subset of compounds for testing. By relying on very large, internally consistent datasets, large pharma has succeeded in developing highly predictive proprietary models [5–8]. At Pfizer (and probably other companies), for example, many of these models (e.g., those that predict the volume of distribution, aqueous kinetic solubility, acid dissociation constant, and distribution coefficient) [5–8, 16] are believed (according to discussions with scientists) to be so accurate that they have essentially put experimental assays out of business. In most other cases, large pharma perform experimental assays for a small fraction of compounds of interest to augment or validate their computational models. Efforts by smaller pharma and academia have not been as successful, largely because they have, by necessity, drawn upon much smaller datasets and, in a few cases, tried to combine them [11, 17–22]. However, this is changing rapidly, and public datasets in PubChem, ChEMBL, Collaborative Drug Discovery (CDD) and elsewhere are becoming available for ADME/Tox properties. For example, the CDD public database has >100 public datasets that can be used to generate community-based models, including extensive neglected infectious disease structure–activity relationship (SAR) datasets (malaria, tuberculosis, Chagas disease, etc.), and ADMEdata.com

datasets that are broadly applicable to many projects. Recent efforts with them have led to a platform that enables drug discovery projects to benefit from open source machine learning algorithms and descriptors in a secure environment, which allows models to be shared with collaborators or made accessible to the community.

In the area of pharmaceutical research and development and specifically that of cheminformatics, there are many machine learning methods, such as support vector machines (SVM), *k*-nearest neighbors, naïve Bayesian, and decision trees, [23] which have seen increasing use as our datasets, have grown to become "big data" [24–27]. These methods [23] can be used for binary classification, multiple classes, or continuous data. In more recent years, the biological data amassed from HTS and high content screens has called for different tools to be used that can account for some of the issues with this bigger data [26]. Many of these resulting machine learning models can also be implemented on a mobile phone [28, 29].

1.2 Bayesian Models

Our machine learning experience over a decade [14, 30–46] has focused on Bayesian approaches (Figure 1.1). Bayesian models classify data as active or inactive on the basis of user-defined thresholds using a simple probabilistic classification model based on Bayes' theorem. We initially used the Bayesian modeling software within the Pipeline Pilot and Discovery Studio (BIOVIA) with many ADME/Tox and drug discovery datasets. Most of these models have used molecular function class fingerprints of maximum diameter 6 and several other simple descriptors [47, 48]. The models were internally validated through the generation of receiver operator characteristic (ROC) plots. We have also compared single- and dual-event Bayesian models utilizing published screening data [49, 50]. As an example, the single-event models use only whole-cell antitubercular activity, either at a single compound concentration or as a dose–response IC_{50} or IC_{90} (amount of compound inhibiting 50% or 90% of growth, respectively), while the dual-event models also use a selectivity index ($SI = CC_{50}/IC_{90}$, where CC_{50} is the compound concentration that is cytotoxic and inhibits 50% of the growth of Vero cells). While single-event models [13, 51, 52] are widely published, dual-event models [53] attempt to predict active compounds with acceptable relative activity against the pathogen (in this case, *Mtb*), versus the model mammalian cell line (e.g., Vero cells). Our models identified 4–10 times more active compounds than random screening did and the models also had relatively high hit rates, for example, 14% [54], 71% (Figure 1.1) [53], or intermediate [55] for *Mtb*. Recent machine learning work on Chagas disease has identified *in vivo* active compounds [56], one of which is an approved antimalarial in Europe. Most recently, we

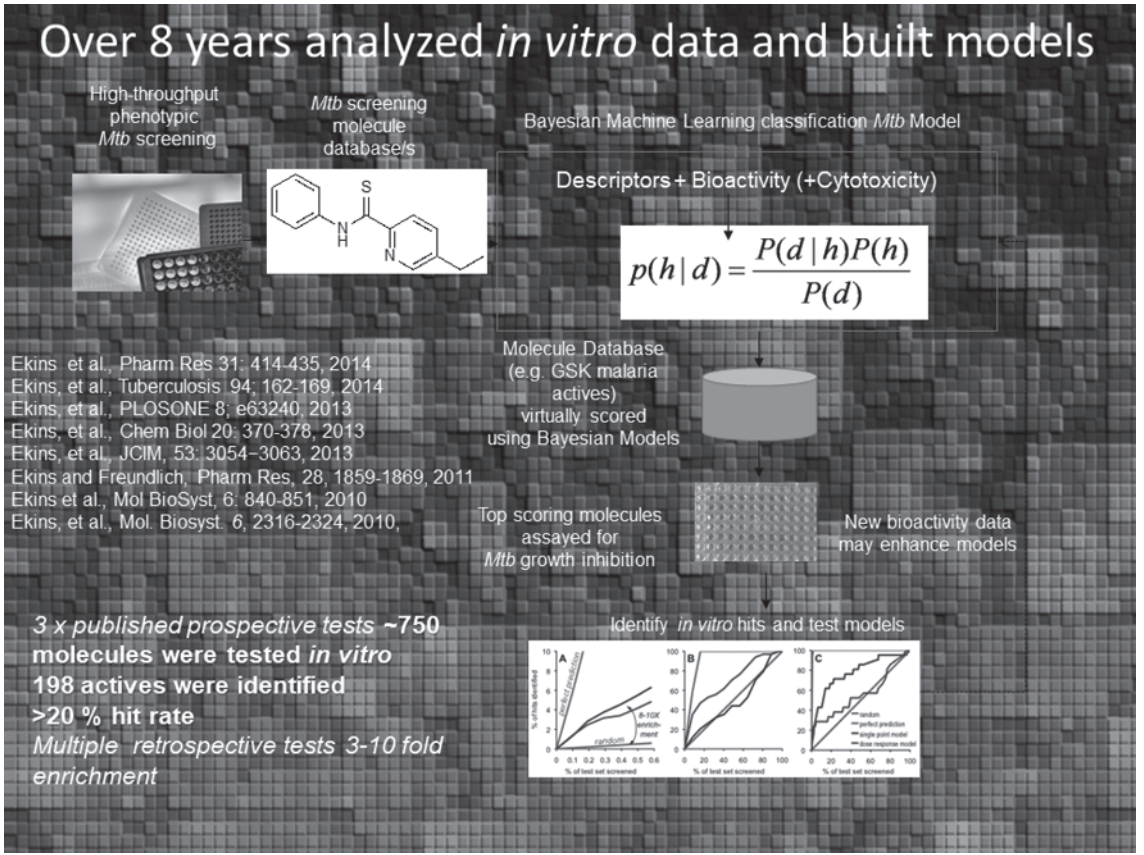

Figure 1.1 Summary of machine learning models generated for *Mycobacterium tuberculosis in vitro* data. This approach has also been applied to ADME/Tox datasets.

have been actively constructing Bayesian models for ADME properties such as aqueous solubility, mouse liver microsomal stability [57], and Caco-2 cell permeability [30], which complement our earlier ADME/Tox machine learning work [13, 52, 58–64]. We have also summarized the application of these methods to toxicology datasets [58] and transporters [34, 59, 62, 63, 65–67]. This has led to models with generally good to acceptable ROC scores > 0.7 [30]. Open source implementation of the ECFP6/FCFP6 fingerprints [28] and Bayesian model building module [25, 30] has also enabled their use in new software implementations (see later). We are keen to explore machine learning algorithms and make them accessible for seeding drug discovery projects, as we have demonstrated.

1.2.1 CDD Models

ADME properties have been modeled by us with collaborators [30] and others using an array of machine learning algorithms, such as SVMs [68], Bayesian modeling [69], Gaussian processes [70], or others [71]. A major challenge remains the ability to share such models. CDD has developed and marketed a robust, innovative commercial software platform that enables scientists to archive, mine, and (optionally) share SAR, ADME/Tox, and other types of preclinical research data [72]. CDD hosts the software and customers' data vaults on its secure servers. CDD collaborated with computational chemists at Pfizer in a proof of concept study. This demonstrated that models constructed with open descriptors and keys (chemical development kit, CDK + SMARTS) using open software (C5.0 - once built, models can be made open) performed essentially identically to expensive proprietary descriptors and models (MOE2D + SMARTS + Rulequest's Cubist) across all metrics of performance when evaluated on multiple Pfizer-proprietary ADME datasets: human liver microsomal (HLM) stability, RRCK passive permeability, P-gp efflux, and aqueous solubility [14]. Pfizer's HLM dataset, for example, contained more than 230,000 compounds and covered a diverse range of chemistry, as well as many therapeutic areas. The HLM dataset was split into a training set (80%) and a test set (20%) using the venetian blind splitting method; in addition, a newly screened set of 2310 compounds was evaluated as a blind dataset. All the key metrics of model performance - for example, R^2, root-mean-square error (RMSE), kappa, sensitivity, specificity, positive predictive value (PPV) - were nearly identical for the open source approach versus the proprietary software (e.g., PPV of 0.80 vs 0.82). The open source approach even computed slightly faster (0.2 vs 0.3 s/compound). All the datasets studied yielded the same conclusion, that is, models built with open descriptors and models are as predictive as the commercial tools [14].

This result is an important prerequisite for a goal of creating a machine learning model exchange platform that can be deployed without requiring

licenses for other software or algorithms, which would otherwise make it too expensive to achieve widespread adoption [73, 74]. This preliminary study did not directly address the issue of whether the descriptors mask the underlying data sufficiently well that structure identities cannot be reverse-engineered, but others have begun to assess this question with respect to an array of molecular descriptor types [75] and open source descriptors and models could be used in any other software (GLP license).

Compared to the large datasets available in pharma, there are few that are freely available. Jean Claude Bradley, Andrew Lang, and Antony Williams have, however, provided a curated dataset of melting points for the community using several open data sources, which was then used for modeling. A training set comprising 2205 compounds and a test set of 500 compounds with doubly validated melting points were used with 132 Open CDK [76] descriptors and the RandomForest package (v4.5-34) in R. The resulting RandomForest model had an RMSE of 40.9 °C and an R^2 value of 0.82 when used to predict the test set. We then compared these results to what could be obtained in the commercial SAS JMP (v8.0.1, SAS, Cary, NC) and Discovery Studio (v2.5.5. San Diego, CA). A neural network model in SAS had an RMSE of 48.5 °C and an R^2 value of 0.75. In comparison, a backpropagation neural network model in Discovery Studio had an RMSE of 40.8 °C and an R^2 value of 0.83 for the same test set. These melting point models are all superior to 17 models identified in 10 papers between 2003 and 2011 using commercial and other tools [77]. The results also suggested that open descriptors and algorithms can produce models that are comparable to those generated with commercial tools.

Similarly, we have curated PubChem BioAssay data on mouse liver microsomal (MLM) stability. Our curated training set with MLM half-life values on 894 compounds (from a compilation of 99 different sets of assay results), our external test set with MLM half-life values on 30 antitubercular compounds, and our independent, external validation set with percentage that compounds the remaining data on 571 compounds (from combining 78 different sets of assay results) are all freely available as sdf files in the supplementary material [57]. We hypothesized that when constructing a binary classifier model, the moderately stable/moderately unstable compounds might generate confusion or even disinformation during the machine learning process. Consequently, we proposed that a novel data "pruning" strategy should be investigated: the conventional, or "full," model was constructed using a training set in which stable compounds were defined as having a $t_{1/2} \geq 60$ min and unstable compounds had a $t_{1/2} < 60$ min, while the new "pruned" model had a training set that used the same stable compounds with a $t_{1/2} \geq 60$ min, but only the compounds with a $t_{1/2} < 30$ min were used as unstable compounds. Compounds with a half-life between 30 and 59.4 min were simply deleted from the full training set in order to create the pruned training set. The pruned MLM Bayesian model displayed superior predictive power versus the full model (in terms of internal

and external statistics, as well as histogram-based analyses), even though less information was used to train the pruned model [57]. Since then, we have continued to explore our novel data pruning strategy when constructing Bayesian models to predict other types of properties: in some cases, the pruned models are significantly more accurate, while in one case, the pruning process did not improve predictive power (but it did not substantially degrade performance, either). Pruning is a simple protocol but perhaps a counterintuitive notion (i.e., the machine can learn more by teaching it with less data). Our results thus far indicate that this pruning strategy merits further investigation.

We have recently integrated validated computational models for ADME/Tox and physicochemical properties, for example, human metabolic stability, Caco-2 permeability, protein binding, solubility, melting point, hERG, pregnane X receptor (PXR), cytotoxicity, CYP3A4 inhibition, CYP2D6 inhibition, CYP2C9 inhibition, drug induced liver injury (DILI) [52], and P-gp (and other transporters) [34, 63, 66, 67]. NCGC and others have generated large, open or published datasets for Cytochrome P450's, PXR, hERG [78], aggregation, [79] and so on, which can also be used for modeling, although the structures used may need additional curation based on our recent findings that lead us to question the structure quality [80, 81]. Molecule quality could adversely affect computational models, so it will be important to run these through new tools for structure assessment, such as those available in ChemSpider, among others [82]. One of the key reasons for using open source tool kits is that this will allow big pharma companies to share their models with outside groups more readily, whereas different vendor tools for building models are generally incompatible.

We will now provide some additional detail to justify why we think it is important to put considerable effort into building this model-sharing capability and community. In this case, we considered how models could be shared and the outputs visualized. In general, the quality of model scales with leave-one-out or fivefold cross-validation ROC (values > 0.7 to 0.8 would be ideal). Using models with ROC > 0.7, we have demonstrated that these models can reliably rank molecules such that the users can either take the top N% of compounds or use medicinal chemistry intuition to filter them, with essentially the same hit rates observed [53, 54, 56, 83].

A number of modeling projects in recent years have successfully made use of the extended connectivity fingerprints, commonly referred to as ECFP_n or FCFP_n (n = 2, 4, or 6, etc.). For example, we have amassed experience in applying the FCFP_6 descriptors to modeling phenotypic HTS data for *Mtb* and other datasets. These fingerprints are created by enumerating a collection of substructures using breadth-first expansion from a starting atom. The fingerprint method was originally made available as part of the Pipeline Pilot project and similar methods have been made available from ChemAxon's proprietary JChem and RDKit. The Accelrys fingerprint methodology used by us in all our

previous modeling work was published in detail, but the disclosure omitted a number of trade secrets, which means that while it is now straightforward to implement an algorithm that generates fingerprints that are similarly effective, it is not possible to produce results that can be directly comparable between the two different implementations.

We therefore created a drop-in replacement for the ECFP_6 fingerprints that can be readily ported between multiple toolkits and programming languages. We have thus built and validated an algorithm that follows the published references for ECFP and FCFP fingerprints as closely as possible, and we made the resulting code available to the public as a feature in the CDK project under an open source license. We have evaluated the ROC of models built previously in the literature and with our own Bayesian and open source descriptors and found them to be near identical. While this is in itself a valuable addition to the popular Java-based toolkit, we have taken care to implement the algorithm in a concise manner with few external dependencies. Avoiding toolkit-specific supporting algorithms has allowed us to port the ECFP_6 algorithm to other platforms. As part of the model building software, we have initially opted for the Bayesian algorithm, as we found little difference between the Bayesian, SVM, and recursive partitioning algorithms when tested on external datasets or using internal cross-validation.

We have coded the software and implemented a version of CDD models. The source code for the Bayes model is open source (MIT license), https://github.com/cdd/modified-bayes. Creating a model requires two sets of molecules to train the model: the "good or active" molecules and a previously screened training set. CDD Vault uses the FCFP_6 structural fingerprints to build a Bayesian statistical model. The model then generates a score that can be used to rank compounds that have not yet been screened. The model is stored as a special type of protocol (category = quantitative structure–activity relationship (QSAR) model), and it provides an ROC plot, so its effectiveness can be gauged. ROC curves are graphic representations of the relationship existing between the sensitivity (i.e., the true positive rate on the y-axis) and the specificity (i.e., the false positive rate on the x-axis) of a statistical test. It is generated by plotting the fraction of true positives out of the total number of actual positives (sensitivity) versus the fraction of false positives out of the total actual negatives (1 − specificity). Each molecule receives a relative score, applicability number, and maximum similarity number. The model will automatically score all compounds in the project that is selected, while creating it. It can subsequently be shared with other projects to score more molecules.

A naïve Bayesian model is optimized for sparse datasets. The learned models are created with a straightforward learn-by-example paradigm: give it a set of hit compounds (the "good" samples), and the system learns to distinguish them from other baseline data. The learning process generates a large set of Boolean

features from the input FCFP_6 fingerprints, then collects the frequency of occurrence of each feature in the "good" subset and in all data samples. To apply the model to a particular compound, the features of the compound are generated and a weight is calculated for each feature using a Laplacian-adjusted probability estimate. The model reports a score, which is calculated by normalizing the probability, taking the natural log, and summing the results. This score is a relative predictor of the likelihood of that sample being from the "good" subset: the higher the score, the higher the likelihood. Once trained, the model can be applied to a set of compounds whose activity is unknown, and it provides a score whose value gives a prediction of the likelihood that the molecule will be a hit in the modeled protocol.

To get an idea of the range of scores, the user can sort the score column by clicking on the header in the search results table. By clicking again one can sort from the highest number to the lowest. Now that the user has an idea of the range of possible scores, the molecules can be filtered to show only high values. The Applicability score is the fraction of structural features that a particular compound shared with the entire training set of molecules. Maximum Tanimoto/Jaccard similarity to any of the "good" molecules in the training set is also calculated. This value is independent of the Bayesian model, and it provides a way to perform a similarity search that compares it to all of the active compounds at once. It is also a way to identify whether a compound was in the training set for the model, in which case, the similarity value is equal to 1.

We have described the testing of this software using datasets for malaria, tuberculosis, cholera, Ames mutagenicity, mouse intrinsic clearance, human intrinsic clearance, Caco-2 cell permeability, 5-HT2B, solubility, PXR activation, maximum recommended therapeutic dose, and blood-brain barrier permeability. In most cases, the threefold cross-validation ROC values are greater than 0.75. The ROC values were comparable to models previously published by us using the commercial descriptors and Bayesian algorithm. In addition to making the technologies open source, we have also described how the models can be built and implemented in a mobile app called mobile molecular datasheet (MMDS) (Figure 1.2). Models for solubility, probe-likeness, hERG, KCNQ1, bubonic plague, Chagas disease, tuberculosis, and malaria were created and also made open source (http://molsync.com/bayesian1). As a follow-up to this work, (and not using the CDD platform), we have now undertaken a large-scale validation study [25] in order to ensure that the Bayesian modeling technique generalizes to a broad variety of drug discovery datasets and the open source software can be used in different scenarios. Most recently, we have been involved in developing semiquantitative Bayesian models and making these open source, as well [84].

These efforts would suggest that a modeling ecosystem can be created, with multiple software being able to use the open source descriptors and algorithms, so that a consistent model format is achieved.

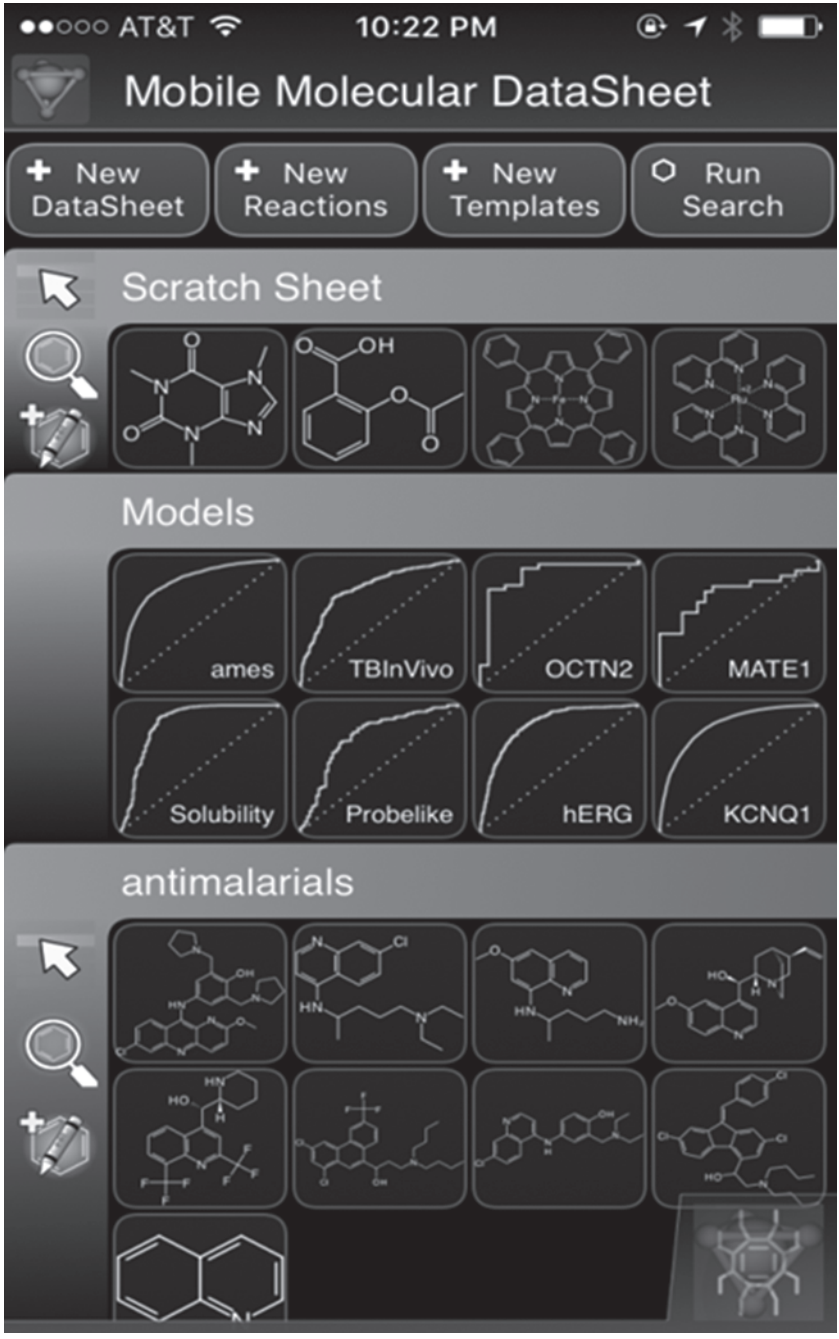

Figure 1.2 Example of Bayesian models implemented in MMDS. (*See color plate section for the color representation of this figure.*)

1.3 Deep Learning Models

In recent years, there has been increasing use of an approach called *deep learning* (*DL*), which builds on many years of artificial neural network research [85] and which has shown powerful advantages in learning from images and languages [86]. This may represent the next era of cheminformatics and pharmaceutical research in general, which is focused on mining the heterogeneous big data that is accumulating, using more sophisticated algorithms such as DL.

Widely described artificial neural networks (ANN) approaches use an input layer, hidden layer, and output layer (Figure 1.3a), where each connection has a weight, and these vary during training in order to connect input to output data. This method has been used extensively, but it suffers from overfitting of data and a poor ability to generalize with an external dataset [23], although more recent versions such as Bayesian regularized artificial neural networks are less prone to being overtrained [87]. DL or deep neural networks (DNNs) [23] are in many ways similar to ANN in that they mimic how the brain works and take information via an input layer. But unlike ANN, DL has many hidden layers [88] to combine signals with different weights, passing the results successively deeper in the network until reaching an output layer (Figure 1.3b). The DL model is trained with a dataset by adjusting the weights to give the response expected for a certain input (e.g., whether a compound is active or inactive or the level of activity/inactivity). The ability to have multiple learnable stages makes this approach more useful for tackling more complex problems. DL can be used for unsupervised learning and appears to work well with noisy data. However, it still suffers from the potential to overfit data, besides displaying higher computational cost than ANN or other methods [89]. To date, there has been relatively limited application of DL to pharmaceutical problems and very few studies in the area of cheminformatics, as compared with other machine

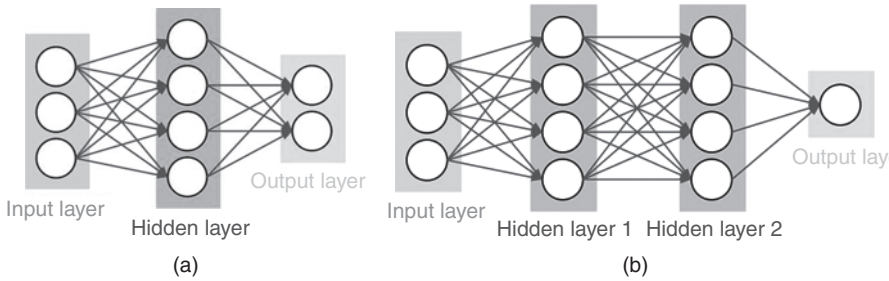

Figure 1.3 (a) A two-layer neural network (one hidden layer of four neurons (or units) and one output layer with two neurons), and three inputs. (b) A three-layer neural network with three inputs, two hidden layers of four neurons each and one output layer. In both cases, there are connections (synapses) between neurons across layers, but not within a layer. Source: Adapted from http://cs231n.github.io/neural-networks-1/.

learning methods [85]. DL tools are available in popular open source statistical software, such as R [90]. In addition, we have TensorFlow [91], Deeplearning4j [92] and Facebook, who made their DL software (Torch) open source [93, 94], followed a year later by Microsoft (CNTK) [95]. Some of these methods have been summarized in a recent review [96]. While these are open source, they need some considerable expertise to utilize, or they require the employment of a specialist that is skilled in integrating these with cheminformatics data such as molecular descriptors.

We are currently developing an open science data repository (OSDR) [97] for connecting scientists and sharing data for many types of projects relevant to drug discovery (see also Chapter 13). OSDR represents a general platform for acquisition, curation, semantic enrichment, and management of various scientific data related to chemistry, bioinformatics, and pharmacology. OSDR also provides a powerful and extensible framework for hosting not just data but also various prediction algorithms, as well as previously generated models.

We have integrated DL into OSDR to provide a user-friendly implementation of the technology. There is increasing interest from big pharma companies working on new methods for QSAR [98, 99]. While such experts have ready access to a wide variety of in-house and commercial software, smaller companies may be at a disadvantage as these skills and software may be less accessible. It is our goal to make DL for cheminformatics accessible to non-experts in academia and industry. In addition, while there are many proponents of DL and other machine learning techniques, they do not have the advantage of drug discovery expertise; consequently, they frequently oversell the utility of such technology or misuse public datasets. It is therefore important to access and test DL. Adding machine learning methods and DL to OSDR would clearly differentiate it from capabilities found elsewhere (e.g., Figshare, Mendeley, CDD, and many other systems, both commercial and open source) for depositing data. It would enable the ability to learn from data, to build and share models, as well as make predictions that could enable many uses in drug discovery and similar areas where it is important to learn from molecular structures. It should be noted that the open source DL toolkits described earlier are far from "plug and play" type software tools for the average scientist, in which their molecules and data are input to train a model (or for that matter in any training or test datasets) and then generate predictions. Significant expertise in using these software toolkits is needed and integrating them with molecular descriptor software is a problem in itself, requiring deep knowledge of cheminformatics toolkit(s) and their capabilities. It is more likely that a specialized programmer/statistician/cheminformatician with knowledge of the software tools will be needed to generate the models, which can then be made available for others to use. Conversely, our approaches described herein could facilitate making DL more accessible to non-expert users by developing easy to use, fully integrated

tools, which can be applied with any dataset in OSDR or used as standalone software to produce models.

There have been very few discussions of the potential for using DL in pharmaceutical research [88, 89]. The results obtained thus far have admittedly focused on internal validation with little prospective testing, as seen with other machine learning methods [53, 100]. DL appears promising and will likely see greater application in the years ahead. So how long will it be before DL is widespread in pharmaceutical research [88] and what can we expect? It is possible that DL could be the source of more predictive models, but hurdles remain in the implementation and accessibility of these models. In addition, there is also the healthy skepticism of any new computational technology that has to be addressed before it is able to be used widely in the industry. What is clearly needed is software that is tightly integrated with the data to be modeled. This data would most frequently reside in private or public databases and could represent many different endpoints, both quantitative and qualitative. Therefore, any efforts to bring the molecules, sources of data, and DL algorithms together would greatly streamline model generation and make it more accessible to other scientists. However, as with other computational modeling approaches, we may also want to consider the applicability domain [101] and various critical factors, such as the quality of the underlying data [80, 102], which may determine the utility and relevance of a DL model for making a prospective prediction [103]. Already, comparisons of DL with other machine learning algorithms have shown that it frequently improves upon the state of the art, when using predominantly internal cross-validation as the form of evaluation. At the time of this writing, there are over 100 DL start-up companies globally, but few are focused on pharmaceutical applications alone [104, 105].

Presently, there are a variety of open source libraries implementing DL algorithms. There is also a set of mature and well-recognized open source cheminformatics toolkits which are able to generate feature sets for chemical structures that, when combined with labeling information on properties or descriptors, can be used to train machine learning algorithms to generate predictive models. Unfortunately, these two areas usually have to be manually connected to support the overall pipeline of drug discovery. DL algorithms need to be accessible to readily scour libraries of compounds for the property of interest. OSDR provides a powerful and extensible framework for hosting not just data but also various prediction algorithms as well as previously generated models. We have built a Jupyter Notebook directly into OSDR to seamlessly integrate chemical operations, datasets manipulation, and machine learning models (DL, as well as Bayesian, trees, etc.) within one framework. As DL methods have not been widely assessed using prospective validation, we can use our approach to take previously published and novel data input in

OSDR, build models, and evaluate them for internal quality, before validating them using prospective predictions on vendor libraries.

1.4 Comparison of Different Machine Learning Methods

We have been interested in comparing DNNs with classic machine learning (CML) methods with different datasets of toxicological relevance for future embedding into the OSDR [97].

Diverse publicly available datasets for different types of ADME/Tox activities were used to develop prediction pipelines [30, 106] (Table 1.1). The ECFP6 fingerprints, consisting of 1024-bin datasets, were computed from sdf files using RDKit (http://www.rdkit.org/). A typical frequency of fingerprints occurrence in the 1024 bin compound representation in a dataset is shown in

Table 1.1 Comparison of machine learning methods using FCFP6 1024 bit descriptors on ADME/Tox properties using fivefold cross-validation ROC values.

Models	BNB	LLR	ABDT	RF	SVM	DNN-2	DNN-3	Active/ inactive and ratio
Solubility train	0.9594	0.9911	0.9963	0.9336	0.9833	0.9996	0.9996	1144/155, 7.38
Solubility test	0.8621	0.9375	0.9323	0.8738	0.9267	0.9349	0.9332	
hERG Train	0.9302	0.9162	0.9916	0.9219	0.9600	1.0000	1.0000	373/433, 0.86
hERG Test	0.8424	0.8529	0.8436	0.8343	0.8637	0.8400	0.8409	
KCNQ Train	0.7951	0.8637	0.8087	0.7644	0.8638	1.0000	1.0000	301, 737/3878, 77.81
KCNQ Test	0.7855	0.8256	0.8012	0.7321	0.8318	0.8608	0.8559	
ERα agonist train	0.9320	0.9820	0.9730	0.9300	0.9920	0.9986	0.9986	966/1178, 0.82
ERα agonist test	0.9120	0.9340	0.9370	0.9120	0.9280	0.9360	0.9364	

The test set consists of 20-25% of the original records, separated before training and used for validation. BNB, Bernoulli naive Bayes; LLR, logistic linear regression; ABDT, AdaBoost decision trees; RF, random forest; SVM, support vector machines; DNN-*N*, DNN with two or three hidden layers. The solubility dataset consisted of 1299 molecules, hERG had 806 molecules, KCNQ1 had 305,615 molecules, and the ERα agonist dataset had 2144 molecules. Note: The active/inactive ratios for hERG and KCNQ1 are reversed as we are trying to obtain compounds that are more desirable (active = noninhibitors).

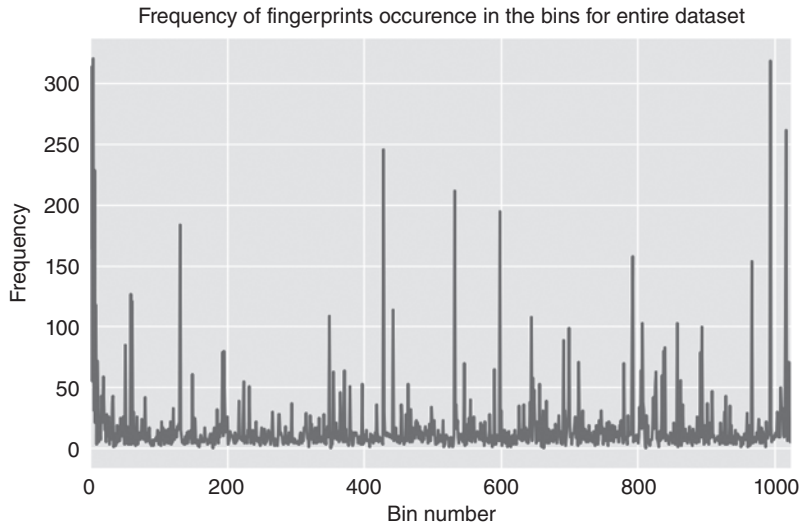

Figure 1.4 Typical frequency of fingerprints occurrence in the 1024-bin compounds in a dataset.

Figure 1.4. Two general prediction pipelines were developed. The first pipeline used only CML methods, such as Bernoulli naive Bayes (BNB), linear logistic regression, AdaBoost decision tree, Random Forest (RF), and SVM. The open source Scikit-learn (http://scikit-learn.org/stable/) ML python library was used for building, tuning, and validating all these CML models. The second pipeline used DNN learning models using Keras (https://keras.io/), a DL library, and Tensorflow (www.tensorflow.org) as a backend. The developed pipeline consists of stratified splitting of the input dataset into train (80%) and test (20%) datasets. Hence tuning of all the models and the search for hyper parameters were conducted solely on the training dataset for better model generalization. The ROC curve and the area under the curve (AUC) were computed for each model.

1.4.1 Classic Machine Learning Methods

The following details the classic machine learning methods used in the first pipeline.

1.4.1.1 Bernoulli Naive Bayes

Naive Bayes method is a supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. BNB implements the naive Bayes training and classification algorithms for data that are distributed according to multivariate Bernoulli distributions; that is, there may be multiple features but each one is assumed

to be a binary-valued (Bernoulli, Boolean) variable. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. On the other hand, although naive Bayes is known as a decent classifier, it is known to be not a very good estimator, so the class probability outputs are not very accurate. The BNB model was tuned and trained using the *BernoulliNB()* method from Naïve Bayes module of Scikit-learn. The fourfold stratified cross-validation with a nonparametric approach based on isotonic regression for balancing classes (most of datasets are heavily imbalanced) was used. The cross-validation generator estimates the model parameter on the training portions of the cross-validation split for each split, and the calibration is done on the test cross-validation split of the training dataset, the probabilities predicted for the folds are then averaged. AUC was computed using those probabilities.

1.4.1.2 Linear Logistic Regression with Regularization

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution, thus predicting the probability of particular outcomes. The L2 binominal regularized logistic regression method was used to classify the activities. A stochastic average gradient optimizer was used in the *LogisticRegressionCV()* method from the linear module of Scikit-learn. A fourfold stratified cross-validation method was used in a grid search of the best regularization parameter (L2 penalties were in logarithmic scale between 1e−5 and 1e−1). The AUC of ROC was used for scoring the classification (maximizing AUC) performance for each fold of balanced classes' classification task.

1.4.1.3 AdaBoost Decision Tree

AdaBoost is a type of "ensemble learning" where multiple learners are employed to build a stronger learning algorithm by conjugating many weak classifiers. The decision tree (DT) was chosen as a base algorithm in our implementation of the AdaBoost method (ABDT). The *AdaBoostClassifier()* method with 100 estimators and 0.9 learning rate from Scikit-learn ensemble methods was used. Similarly to naïve Bayes, the ABDT model was tuned using isotonic calibration for the imbalanced classes with the fourfold stratified cross-validation method.

1.4.1.4 Random Forest

The RF method is another ensemble method, which fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. The *RandomForestClassifier()* method with maximum depth of tree 5 and balanced classes weights

was used to build the model. The fourfold stratified cross-validation grid search was done using 5, 10, 25, and 50 estimators with the AUC of ROC as a scoring function of the estimator.

1.4.1.5 Support Vector Machine

SVM is one of the most popular supervised machine learning algorithms used mostly in classification problems and it is quite effective in high-dimensional spaces. The learning of the hyperplane in SVM algorithm can be done using different kernel functions for the decision function. The C SVM classification with libsvm implementation method from Scikit-learn was also used (*svm.SVC()*). The fourfold stratified cross-validation grid search using weighted classes was done for two kernels (linear, rbf), *C* (1, 10, 100), and gamma values (1e−2, 1e−3, 1e−4). The parameter *C*, common to all SVM kernels, trades off misclassification of training examples against simplicity of the decision surface. A low *C* makes the decision surface smooth, while a high *C* aims at classifying all training examples correctly. Gamma defines how much influence a single training example has. The larger gamma is, the closer other examples must be to be affected. The implementation of SVM automatically finds the best parameters and saves the best SVM model for activity predictions.

1.4.2 Deep Neural Networks

N-layer neural networks are shown in Figure 1.2. It is worth noting that a single-layer neural network describes a network with no hidden layers where the input is directly mapped to the output layer. In that sense, the logistic regression or SVM methods are simply a special case of single-layer neural networks. In this work, for simplification of the DNN representation, we will be counting hidden layers only. Neural networks with 1–2 hidden layers are often called *shallow neural networks* and those with three or more hidden layers are known as the *DNNs*. Two basic approaches to avoid DNN moel overfitting used in training are the L2 norm and dropout regularizaton for all hidden layers. The following hyperparameter optimization was performed using a DNN with three hidden layers: Keras with Tensorflow backend and the grid-search method from Scikit-learn. The following parameters were optimized prior to final model training:

- optimization algorithm: *SGD*, *Adam*, *Nadam*
- learning rate: 0.05, 0.025, 0.01, 0.001
- network weight initialization: *uniform*, *lecun_uniform*, *normal*, *glorot_normal*, *he_normal*, *he_normal*
- hidden layers activation function: *relu*, *tanh*, *LeakyReLU*, *SReLU*
- output function: *softmax*, *softplus*, *sigmoid*
- L2 regularization: 0.05, 0.01, 0.005, 0.001, 0.0001
- dropout regularization: 0.2, 0.3, 0.5, 0.8

- the number of nodes in a hidden layer (all hidden layers): 512, 1024, 2048, 4096
- The following hyperparameters were used for further DNN training: *SGD*, learning rate 0.01 (automatically 10% reduced on plateau of 50 epochs), weight initialization he_normal, hidden layers activation SReLU, output layer function *sigmoid*, L2 regularization 0.001, dropout 0.5. The *binary crossentropy* was used as a loss function. In order to save training time, an early training termination was implemented by stopping the training if no change in loss was observed after 200 epochs. The number of hidden nodes in all hidden layers was set equal to the number of input features (number of bins in the fingerprints).

1.4.3 Comparing Models

The AUC values of the all trained models for compounds represented as ECFP6 in 1024-bin fingerprints are summarized in Table 1.1 and the F1 scores [107] are summarized in Table 1.2. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0. In all cases, the SVM models were better than any other CLM models including naïve Bayes for the test set ROC values. The DNN models were also better than the SVM for three out of four datasets based on ROC values. Using the F1 scores, DNN outperformed all methods for the solubility using KCNQ data and ARα agonist datasets, while the Bayesian method performed well for the hERG data (Table 1.2). The 1024-bin fingerprints may not however be sufficient for maximizing DNN performance, thus other 2D or 3D

Table 1.2 Comparison of machine learning methods using FCFP6 1024-bit descriptors on ADME/Tox properties using fivefold cross-validation F1 values at $p = 0.5$.

Models	BNB	LLR	ABDT	RF	SVM	DNN-2	DNN-3
Solubility train	0.9417	0.9626	0.9595	0.9559	0.9539	0.9917	0.9917
Solubility test	0.9087	0.9446	0.9457	0.9451	0.9396	0.9586	0.9609
hERG Train	0.8536	0.8406	0.9562	0.8249	0.8852	1.0000	1.0000
hERG Test	0.7976	0.7975	0.7152	0.7799	0.7843	0.7763	0.7843
KCNQ train	0.7962	0.8646	0.8193	0.8332	0.8558	0.9991	0.9999
KCNQ test	0.7938	0.8578	0.8157	0.8251	0.8508	0.9911	0.9923
ERα agonist train	0.8355	0.9173	0.8927	0.8304	0.9705	0.9697	0.9697
ERα agonist test	0.8017	0.8201	0.8330	0.7881	0.8535	0.8542	0.8542

The test set consists of 20-25% of the original records, separated before training and used for validation. BNB, Bernoulli naive Bayes; LLR, logistic linear regression; ABDT, AdaBoost decision trees; RF, random forest; SVM, support vector machines; DNN-*N*, DNN with two or three hidden layers.

fingerprints may need to be tried in future with this method. In addition, a far larger number of datasets need to be assessed across the multiple machine learning methods. This work suggests DNN and SVM generally outperform all other machine learning methods when dealing with this selection of four small to very large toxicology datasets and does not depend on whether the datasets are balanced or not.

1.5 Future Work

Doubtless there will be new machine learning algorithms developed in the coming decade. The key for computational toxicology will be to integrate these into cheminformatics workflows and tools that are used in decision making. Our efforts have lead us to providing as open source some of the software tools we have previously taken for granted. Sustaining software companies and the very developers of these tools will require some intelligent choices of how to monetize this work as services such as training and customization of the tools. As scientists, we are driven to solve problems and having the best software available as we deal with different datasets for toxicology will enable us to come up with solutions and hypotheses which we can test experimentally. Clearly, trying out more machine learning approaches in parallel may lead to the selection of the best model per endpoint. Readily accessible machine learning models are likely to be an increasingly important tool for drug discovery in general and these may fuse public and private data. Such models will still require some expertize to use and interpret, thus creating new opportunities for cheminformaticians.

Acknowledgments

S.E. acknowledges support from NIH Grants 9R44TR000942-02 (while at CDD). S.E. also acknowledges many fruitful discussions with Dr Barry Bunin and Dr Antony Williams. Kimberley Zorn is acknowledged for providing the ERα agonist dataset.

References

1 Oprea, T.I. and Matter, H. (2004) Integrating virtual screening in lead discovery. *Curr. Opin. Chem. Biol.*, **8**, 349–358.
2 Ekins, S., Mestres, J., and Testa, B. (2007) In silico pharmacology for drug discovery: applications to targets and beyond. *Br. J. Pharmacol.*, **152**, 21–37.

3 Ekins, S., Mestres, J., and Testa, B. (2007) In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling. *Br. J. Pharmacol.*, **152**, 9–20.

4 McGaughey, G.B., Sheridan, R.P., Bayly, C.I. *et al.* (2007) Comparison of topological, shape, and docking methods in virtual screening. *J. Chem. Inf. Model.*, **47**, 1504–1519.

5 Lombardo, F., Obach, R.S., Dicapua, F.M. *et al.* (2006) A hybrid mixture discriminant analysis-random forest computational model for the prediction of volume of distribution of drugs in human. *J. Med. Chem.*, **49**, 2262–2267.

6 Lombardo, F., Obach, R.S., Shalaeva, M.Y., and Gao, F. (2004) Prediction of human volume of distribution values for neutral and basic drugs. 2. Extended data set and leave-class-out statistics. *J. Med. Chem.*, **47**, 1242–1250.

7 Lombardo, F., Obach, R.S., Shalaeva, M.Y., and Gao, F. (2002) Prediction of volume of distribution values in humans for neutral and basic drugs using physicochemical measurements and plasma protein binding. *J. Med. Chem.*, **45**, 2867–2876.

8 Lombardo, F., Shalaeva, M.Y., Tupper, K.A., and Gao, F. (2001) ElogDoct: a tool for lipophilicity determination in drug discovery. 2 Basic and neutral compounds. *J. Med. Chem.*, **44**, 2490–2497.

9 Lombardo, F., Blake, J.F., and Curatolo, W.J. (1996) Computation of brain–blood partitioning of organic solutes via free energy calculations. *J. Med. Chem.*, **39**, 4750–4755.

10 Lipinski, C.A., Lombardo, F., Dominy, B.W., and Feeney, P.J. (1997) Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Del. Rev.*, **23**, 3–25.

11 Ekins, S., Ring, B.J., Grace, J. *et al.* (2000) Present and future in vitro approaches for drug metabolism. *J. Pharm. Tox. Meth.*, **44**, 313–324.

12 Tanrikulu, Y., Kruger, B., and Proschak, E. (2013) The holistic integration of virtual screening in drug discovery. *Drug Discov. Today*, **18**, 358–364.

13 Zientek, M., Stoner, C., Ayscue, R. *et al.* (2010) Integrated in silico-in vitro strategy for addressing cytochrome P450 3A4 time-dependent inhibition. *Chem. Res. Toxicol.*, **23**, 664–676.

14 Gupta, R.R., Gifford, E.M., Liston, T. *et al.* (2010) Using open source computational tools for predicting human metabolic stability and additional ADME/TOX properties. *Drug Metab. Dispos.*, **38**, 2083–2090.

15 Ekins, S., Gupta, R.R., Gifford, E. *et al.* (2010) Chemical space: missing pieces in cheminformatics. *Pharm. Res.*, **27**, 2035–2039.

16 Lombardo, F., Shalaeva, M.Y., Tupper, K.A. *et al.* (2000) ElogPoct a tool for lipophilicity determination in drug discovery. *J. Med. Chem.*, **43**, 2922–2928.
17 Lagorce, D., Sperandio, O., Galons, H. *et al.* (2008) FAF-Drugs2: free ADME/tox filtering tool to assist drug discovery and chemical biology projects. *BMC Bioinformatics*, **9**, 396.
18 Villoutreix, B.O., Renault, N., Lagorce, D. *et al.* (2007) Free resources to assist structure-based virtual ligand screening experiments. *Curr. Protein Pept. Sci.*, **8**, 381–411.
19 Ekins, S. (2007) *Computational Toxicology: Risk Assessment for Pharmaceutical and Environmental Chemicals*, John Wiley & Sons, Hoboken, NJ.
20 Balani, S.K., Miwa, G.T., Gan, L.S. *et al.* (2005) Strategy of utilizing in vitro and in vivo ADME tools for lead optimization and drug candidate selection. *Curr. Top. Med. Chem.*, **5**, 1033–1038.
21 van De Waterbeemd, H., Smith, D.A., Beaumont, K., and Walker, D.K. (2001) Property-based design: optimization of drug absorption and pharmacokinetics. *J. Med. Chem.*, **44**, 1313–1333.
22 Walters, W.P. and Murcko, M.A. (2002) Prediction of 'drug-likeness'. *Adv. Drug Del. Rev.*, **54**, 255–271.
23 Mitchell, J.B. (2014) Machine learning methods in chemoinformatics. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **4**, 468–481.
24 Zhu, H., Zhang, J., Kim, M.T. *et al.* (2014) Big data in chemical toxicity research: the use of high-throughput screening assays to identify potential toxicants. *Chem. Res. Toxicol.*, **27**, 1643–1651.
25 Clark, A.M. and Ekins, S. (2015) Open source Bayesian models: 2. Mining a 'big dataset' to create and validate models with ChEMBL. *J. Chem. Inf. Model.*, **55**, 1246–1260.
26 Ekins, S., Clark, A.M., Swamidass, S.J. *et al.* (2014) Bigger data, collaborative tools and the future of predictive drug discovery. *J. Comput. Aided Mol. Des.*, **28**, 997–1008.
27 Ekins, S., Freundlich, J.S., and Reynolds, R.C. (2014) Are bigger data sets better for machine learning? Fusing single-point and dual-event dose response data for *Mycobacterium tuberculosis*. *J. Chem. Inf. Model.*, **54**, 2157–2165.
28 Clark, A.M., Sarker, M., and Ekins, S. (2014) New target predictions and visualization tools incorporating open source molecular fingerprints for TB Mobile 2.0. *J. Cheminform.*, **6**, 38.
29 Ekins, S., Clark, A.M., and Wright, S.H. (2015) Making transporter models for drug–drug interaction prediction mobile. *Drug Metab. Dispos.*, **43**, 1642–1645.
30 Clark, A.M., Dole, K., Coulon-Spector, A. *et al.* (2015) Open source bayesian models: 1. Application to ADME/Tox and drug discovery

datasets. *J. Chem. Inf. Model.*, **55**, 1231–1245. doi: 10.1021/acs.jcim.5b00143. Epub 2015 Jun 3.

31 Kortagere, S. and Ekins, S. (2010) Troubleshooting computational methods in drug discovery. *J. Pharmacol. Toxicol. Methods*, **61**, 67–75.

32 Ekins, S. and Williams, A.J. (2010) Precompetitive preclinical ADME/Tox data: set it free on the web to facilitate computational model building to assist drug development. *Lab Chip*, **10**, 13–22.

33 Ekins, S., Honeycutt, J.D., and Metz, J.T. (2010) Evolving molecules using multi-objective optimization: applying to ADME. *Drug Discov. Today*, **15**, 451–460.

34 Bahadduri, P.M., Polli, J.E., Swaan, P.W., and Ekins, S. (2010) Targeting drug transporters – combining in silico and in vitro approaches to predict in vivo. *Methods Mol. Biol.*, **637**, 65–103.

35 Ekins, S., Bugrim, A., Brovold, L. *et al.* (2006) Algorithms for network analysis in systems-ADME/Tox using the MetaCore and MetaDrug platforms. *Xenobiotica*; the fate of foreign compounds in biological systems, **36**, 877–901.

36 Ekins, S., Andreyev, S., Ryabov, A. *et al.* (2006) A combined approach to drug metabolism and toxicity assessment. *Drug Metab. Dispos.*, **34**, 495–503.

37 Ekins, S. (2006) Systems-ADME/Tox: resources and network approaches. *J. Pharmacol. Toxicol. Methods*, **53**, 38–66.

38 Chang, C. and Ekins, S. (2006) Pharmacophores for human ADME/Tox-related proteins, in *Pharmacophores and Pharmacophore Searches* (eds T. Langer and R.D. Hoffman), Wiley-VCH, Weinheim, pp. 299–324.

39 Ekins, S., Nikolsky, Y., and Nikolskaya, T. (2005) Techniques: application of systems biology to absorption, distribution, metabolism, excretion and toxicity. *Trends Pharmacol. Sci.*, **26**, 202–209.

40 Ekins, S., Andreyev, S., Ryabov, A. *et al.* (2005) Computational prediction of human drug metabolism. *Expert Opin. Drug Metab. Toxicol.*, **1**, 303–324.

41 Balakin, K.V., Ivanenkov, Y.A., Savchuk, N.P. *et al.* (2005) Comprehensive computational assessment of ADME properties using mapping techniques. *Curr. Drug Discov. Technol.*, **2**, 99–113.

42 Ekins, S. and Swaan, P.W. (2004) Computational models for enzymes, transporters, channels and receptors relevant to ADME/TOX. *Rev. Comput. Chem.*, **20**, 333–415.

43 Ekins, S., Boulanger, B., Swaan, P.W., and Hupcey, M.A. (2002) Towards a new age of virtual ADME/TOX and multidimensional drug discovery. *Mol. Divers.*, **5**, 255–275.

44 Ekins, S. and Wrighton, S.A. (2001) Application of in silico approaches to predicting drug–drug interactions. *J. Pharmacol. Toxicol. Methods*, **45**, 65–69.

45 Ekins, S., Waller, C.L., Swaan, P.W. *et al.* (2000) Progress in predicting human ADME parameters in silico. *J. Pharmacol. Toxicol. Methods*, **44**, 251–272.

46 Ekins, S., Ring, B.J., Grace, J. *et al.* (2000) Present and future in vitro approaches for drug metabolism. *J. Pharmacol. Toxicol. Methods*, **44**, 313–324.

47 Ekins, S., Bradford, J., Dole, K. *et al.* (2010) A collaborative database and computational models for tuberculosis drug discovery. *Mol. Biosyst.*, **6**, 840–851.

48 Ekins, S., Kaneko, T., Lipinksi, C.A. *et al.* (2010) Analysis and hit filtering of a very large library of compounds screened against *Mycobacterium tuberculosis*. *Mol. Biosyst.*, **6**, 2316–2324.

49 Ananthan, S., Faaleolea, E.R., Goldman, R.C. *et al.* (2009) High-throughput screening for inhibitors of *Mycobacterium tuberculosis* H37Rv. *Tuberculosis*, **89**, 334–353.

50 Maddry, J.A., Ananthan, S., Goldman, R.C. *et al.* (2009) Antituberculosis activity of the molecular libraries screening center network library. *Tuberculosis*, **89**, 354–363.

51 Langdon, S.R., Mulgrew, J., Paolini, G.V., and van Hoorn, W.P. (2010) Predicting cytotoxicity from heterogeneous data sources with Bayesian learning. *J. Cheminform.*, **2**, 11.

52 Ekins, S., Williams, A.J., and Xu, J.J. (2010) A predictive ligand-based Bayesian model for human drug-induced liver injury. *Drug Metab. Dispos.*, **38**, 2302–2308.

53 Ekins, S., Reynolds, R., Kim, H. *et al.* (2013) Bayesian models leveraging bioactivity and cytotoxicity information for drug discovery. *Chem. Biol.*, **20**, 370–378.

54 Ekins S, Reynolds RC, Franzblau SG, Wan B, Freundlich JS, Bunin BA. Enhancing hit identification in *Mycobacterium tuberculosis* drug discovery using validated dual-event Bayesian models *PLOS ONE*. 2013;**8**:e63240.

55 Ekins, S., Casey, A.C., Roberts, D. *et al.* (2014) Bayesian models for screening and TB mobile for target inference with *Mycobacterium tuberculosis*. *Tuberculosis*, **94**, 162–169.

56 Ekins, S., Lage de Siqueira-Neto, J., McCall, L.-I. *et al.* (2015) Machine learning models and pathway genome data base for *Trypanosoma cruzi* drug discovery. *PLoS* neglected tropical diseases, **9**, e0003878.

57 Perryman, A.L., Stratton, T.P., Ekins, S., and Freundlich, J.S. (2015) Predicting mouse liver microsomal stability with 'pruned' machine learning models and public data. *Pharm. Res.*, **33**, 433–449.

58 Ekins, S. (2014) Progress in computational toxicology. *J. Pharmacol. Toxicol. Methods*, **69**, 115–40.

59 Dong, Z., Ekins, S., and Polli, J.E. (2013) Structure-activity relationship for FDA approved drugs as inhibitors of the human sodium taurocholate cotransporting polypeptide (NTCP). *Mol. Pharm.*, **10**, 1008–1019.

60 Astorga, B., Ekins, S., Morales, M., and Wright, S.H. (2012) Molecular determinants of ligand selectivity for the human multidrug and toxin extrusion proteins, MATE1 and MATE-2K. *J. Pharmacol. Exp. Ther.*, **341**, 743–755.

61 Pan, Y., Li, L., Kim, G. *et al.* (2011) Identification and validation of novel hPXR activators amongst prescribed drugs via ligand-based virtual screening. *Drug Metab. Dispos.*, **39**, 337–344.

62 Diao, L., Ekins, S., and Polli, J.E. (2010) Quantitative structure activity relationship for inhibition of human organic cation/carnitine transporter. *Mol. Pharm.*, **7**, 2120–2130.

63 Zheng, X., Ekins, S., Raufman, J.P., and Polli, J.E. (2009) Computational models for drug inhibition of the human apical sodium-dependent bile acid transporter. *Mol. Pharm.*, **6**, 1591–1603.

64 Ekins, S., Kortagere, S., Iyer, M. *et al.* (2009) Challenges predicting ligand–receptor interactions of promiscuous proteins: the nuclear receptor PXR. *PLoS Comput. Biol.*, **5**, e1000594.

65 Dong, Z., Ekins, S., and Polli, J.E. (2014) Quantitative NTCP pharmacophore and lack of association between DILI and NTCP inhibition. *Eur. J. Pharm. Sci.*, **66C**, 1–9.

66 Ekins, S., Diao, L., and Polli, J.E. (2012) A substrate pharmacophore for the human organic cation/carnitine transporter identifies compounds associated with rhabdomyolysis. *Mol. Pharm.*, **9**, 905–913.

67 Diao, L., Ekins, S., and Polli, J.E. (2009) Novel inhibitors of human organic cation/carnitine transporter (hOCTN2) via computational modeling and in vitro testing. *Pharm. Res.*, **26**, 1890–1900.

68 Kortagere, S., Chekmarev, D.S., Welsh, W.J., and Ekins, S. (2008) New predictive models for blood brain barrier permeability of drug-like molecules. *Pharm. Res.*, **25**, 1836–1845.

69 Klon, A.E., Lowrie, J.F., and Diller, D.J. (2006) Improved naive Bayesian modeling of numerical data for absorption, distribution, metabolism and excretion (ADME) property prediction. *J. Chem. Inf. Model.*, **46**, 1945–1956.

70 Obrezanova, O., Csanyi, G., Gola, J.M., and Segall, M.D. (2007) Gaussian processes: a method for automatic QSAR modeling of ADME properties. *J. Chem. Inf. Model.*, **47**, 1847–1857.

71 Zhang, L., Zhu, H., Oprea, T.I. *et al.* (2008) QSAR modeling of the blood–brain barrier permeability for diverse organic compounds. *Pharm. Res.*, **25**, 1902–1914.

72 Ekins, S., Freundlich, J.S., Choi, I. *et al.* (2011) Computational databases, pathway and cheminformatics tools for tuberculosis drug discovery. *Trends Microbiol.*, **19**, 65–74.

73 Hull, D., Wolstencroft, K., Stevens, R. *et al.* (2006) Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.*, **34**, W729–W732.

74 Kuhn, T., Willighagen, E.L., Zielesny, A., and Steinbeck, C. (2010) CDK-Taverna: an open workflow environment for cheminformatics. *BMC Bioinformatics.*, **11**, 159.

75 Masek, B.B., Shen, L., Smith, K.M., and Pearlman, R.S. (2008) Sharing chemical information without sharing chemical structure. *J. Chem. Inf. Model.*, **48**, 256–261.

76 Steinbeck, C., Hoppe, C., Kuhn, S. *et al.* (2006) Recent developments of the chemistry development kit (CDK) – an open-source java library for chemo- and bioinformatics. *Curr. Pharm. Des.*, **12**, 2111–2120.

77 Bradley JC. 2011; http://usefulchem.blogspot.com/2011/06/open-melting-points-on-iphone-via-mmds.html.

78 Xia, M., Shahane, S.A., Huang, R. *et al.* (2011) Identification of quaternary ammonium compounds as potent inhibitors of hERG potassium channels. *Toxicol. Appl. Pharmacol.*, **252**, 250–258.

79 Feng, B.Y., Simeonov, A., Jadhav, A. *et al.* (2007) A high-throughput screen for aggregation-based inhibition in a large compound library. *J. Med. Chem.*, **50**, 2385–2390.

80 Williams, A.J., Ekins, S., and Tkachenko, V. (2012) Towards a gold standard: regarding quality in public domain chemistry databases and approaches to improving the situation. *Drug Discov. Today*, **17**, 685–701.

81 Williams, A.J. and Ekins, S. (2011) A quality alert and call for improved curation of public chemistry databases. *Drug Discov. Today*, **16**, 747–750.

82 Pence, H.E. and Williams, A.J. (2010) ChemSpider: an online chemical information resource. *J. Chem. Educ.*, **87**, 1123–1124.

83 Ekins, S., Freundlich, J., Clark, A. *et al.* (2015) Machine learning models identify molecules active against Ebola virus in vitro. *F1000Res.*, **4**, 1091.

84 Clark, A.M., Dole, K., and Ekins, S. (2016) Open source Bayesian models: 3. Composite models for prediction of binned responses. *J. Chem. Inf. Model.*, **56**, 275–285.

85 Baskin, I.I., Winkler, D., and Tetko, I.V. (2016) A renaissance of neural networks in drug discovery. *Expert Opin. Drug Discov.*, **11**, 785–795.

86 LeCun, Y., Bengio, Y., and Hinton, G. (2015) Deep learning. *Nature*, **521**, 436–444.

87 Burden, F. and Winkler, D. (2008) Bayesian regularization of neural networks. *Methods Mol. Biol.*, **458**, 25–44.

88 Gawehn, E., Hiss, J.A., and Schneider, G. (2016) Deep learning in drug discovery. *Mol. inform.*, **35**, 3–14.

89 Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016) Applications of deep earning in biomedicine. *Mol. Pharm.*, **13**, 1445–1454.

90 Chow J.-F. (2014) *Things to Try After useR! – Part 1: Deep Learning with H_2O*, http://www.r-bloggers.com/things-to-try-after-user-part-1-deep-learning-with-h2o/ (accessed August 10, 2017).

91 Anon (2016). TensorFlow, https://www.tensorflow.org/ (accessed August 10, 2017).

92 Anon. (2016) Deeplearning4j, http://deeplearning4j.org/ (accessed August 10, 2017).

93 Novet J. (2015) Facebook Open-Sources its Cutting-Edge Deep Learning Tools, http://venturebeat.com/2015/01/16/facebook-opens-up-about-more-of-its-cutting-edge-deep-learning-tools/ (accessed August 10, 2017).

94 Chintala S. (2015) FAIR Open Sources Deep-Learning Modules for Torch, https://research.facebook.com/blog/fair-open-sources-deep-learning-modules-for-torch/ (accessed August 10, 2017).

95 Linn A. (2016) Microsoft Releases CNTK, its Open Source Deep Learning Toolkit, on GitHub, http://blogs.microsoft.com/next/2016/01/25/microsoft-releases-cntk-its-open-source-deep-learning-toolkit-on-github/#sm.00013j280xp1sdctrgg21w81es5ov (accessed August 10, 2017).

96 Angermueller, C., Parnamaa, T., Parts, L., and Stegle, O. (2016) Deep learning for computational biology. *Mol. Syst. Biol.*, **12**, 878.

97 Tkachenko V. (2017) OSDR, https://github.com/scidatasoft/OSDR (accessed August 10, 2017).

98 Ma, J., Sheridan, R.P., Liaw, A. *et al.* (2015) Deep neural nets as a method for quantitative structure–activity relationships. *J. Chem. Inf. Model.*, **55**, 263–274.

99 Sheridan, R.P., Wang, W.M., Liaw, A. *et al.* (2016) Extreme gradient boosting as a method for quantitative structure–activity relationships. *J. Chem. Inf. Model.*, **56**, 2353–2360.

100 Zhang, L., Fourches, D., Sedykh, A. *et al.* (2013) Discovery of novel antimalarial compounds enabled by QSAR-based virtual screening. *J. Chem. Inf. Model.*, **53**, 475–492.

101 Tetko, I.V., Bruneau, P., Mewes, H.W. *et al.* (2006) Can we estimate the accuracy of ADME-Tox predictions? *Drug Discov. Today*, **11**, 700–707.

102 Fourches, D., Muratov, E., and Tropsha, A. (2010) Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research. *J. Chem. Inf. Model.*, **50**, 1189–1204.

103 Vracko, M., Bandelj, V., Barbieri, P. *et al.* (2006) Validation of counter propagation neural network models for predictive toxicology according to the OECD principles: a case study. *SAR QSAR Environ. Res.*, **17**, 265–284.

104 Murnane K. (2016) *What is Deep Learning and How is it Useful?*, http://www.forbes.com/sites/kevinmurnane/2016/04/01/what-is-deep-learning-and-how-is-it-useful/#715d1eaf10f0 (accessed August 10, 2017).

105 Murnane K. (2016) *Thirteen Companies That Use Deep Learning To Produce Actionable Results*, http://www.forbes.com/sites/kevinmurnane/2016/04/01/thirteen-companies-that-use-deep-learning-to-produce-actionable-results/#4e710eb07967 (accessed August 10, 2017).

106 Huang, R., Sakamuru, S., Martin, M.T. *et al.* (2014) Profiling of the Tox21 10K compound library for agonists and antagonists of the estrogen receptor alpha signaling pathway. *Sci. Rep.*, **4**, 5664.

107 Van Rijsbergen, C.J. (1979) *Information retrieval*, 2nd edn, Butterworth.