Deep Learning as an Opportunity in Virtual Screening

Thomas Unterthiner *
RISC Software GmbH &
Institute of Bioinformatics
Johannes Kepler University Linz, Austria
unterthiner@bioinf.jku.at

Andreas Mayr *
RISC Software GmbH &
Institute of Bioinformatics
Johannes Kepler University Linz, Austria
mayr@bioinf.jku.at

Günter Klambauer
Institute of Bioinformatics
Johannes Kepler University Linz, Austria
klambauer@bioinf.jku.at

Marvin Steijaert
OpenAnalytics, Belgium
marvin.steijaert@openanalytics.eu

Jörg K. Wegner
Johnson & Johnson
Pharmaceutical Research & Development
jwegner@its.jnj.com

Hugo Ceulemans
Johnson & Johnson
Pharmaceutical Research & Development
hceulema@its.jnj.com

Sepp Hochreiter
Institute of Bioinformatics
Johannes Kepler University Linz, Austria
hochreit@bioinf.jku.at

Abstract

Deep learning excels in vision and speech applications where it pushed the state-of-the-art to a new level. However its impact on other fields remains to be shown. The Merck Kaggle challenge on chemical compound activity was won by Hinton's group with deep networks. This indicates the high potential of deep learning in drug design and attracted the attention of big pharma. However, the unrealistically small scale of the Kaggle dataset does not allow to assess the value of deep learning in drug target prediction if applied to in-house data of pharmaceutical companies. Even a publicly available drug activity data base like ChEMBL is magnitudes larger than the Kaggle dataset. ChEMBL has 13 M compound descriptors, 1.3 M compounds, and 5 k drug targets, compared to the Kaggle dataset with 11 k descriptors, 164 k compounds, and 15 drug targets.

On the ChEMBL database, we compared the performance of deep learning to seven target prediction methods, including two commercial predictors, three predictors deployed by pharma, and machine learning methods that we could scale to this dataset. Deep learning outperformed all other methods with respect to the area under ROC curve and was significantly better than all commercial products. Deep learning surpassed the threshold to make virtual compound screening possible and has the potential to become a standard tool in industrial drug design.

*These authors contributed equally to this work

1 Introduction

The pharmaceutical industry is currently challenged to increase the efficiency of drug development, since every year fewer drugs reach the market [1, 2, 3, 4]. Machine learning methods could exploit a wealth of measurements that were accumulated by pharma companies and, thereby, offer Big Pharma alternatives.

Recently, Merck has organized a Kaggle challenge to involve the machine learning community in tackling drug discovery tasks. Deep neural networks won this challenge. This attracted the attention of Big Pharma toward such technologies. However, the Kaggle dataset does not match the size and characteristics of in-house data of pharmaceutical companies. Therefore it remains to be shown whether deep learning can have indeed an impact on the drug discovery process. Furthermore, it is unclear at which step in the drug design pipeline deep networks should be employed for predicting compound-target interactions.

The first step of a drug design pipeline is to identify a biomolecular *target* upon which a potential drug can act, e.g. a protein whose activity can be modified by a compound to achieve a beneficial therapeutic effect. The next step is to screen tens of thousands of chemical compounds by biological high-throughput assays for interactions with this target — typically measured via IC_{50} or EC_{50} values. Finally, a target-interacting lead compound is selected. It's chemical structure, the scaffold, is modified in order to optimize its efficacy and to reduce its side effects. The high-throughput screening generates a rich source of measurement data which may serve for training machine learning models. Furthermore, for many targets high-throughput assays are not available, consequently screening is time and cost intensive. At this step, biological screening can be substituted by virtual screening, that is, in-silico predictions of compound-target interactions. However, in commercial drug design, virtual screening is only acceptable if the prediction accuracy is high. Another important criterion for the success of virtual screening is its ability to detect new scaffolds interacting with the target.

Virtual screening has another advantage: compounds can not only be tested for interactions with the primary target but also for interactions with other proteins. Typically, chemical compounds interact with more than one protein, and most of these interactions result in unwanted side-effects. Off-target binding can also cause a lack of efficacy as the compound is not available for binding to the target. Currently, many drugs fail in the clinical trials because of undetected side effects. These failures are heavily time- and cost-intensive and the main reason for the decreased efficiency of drug discovery. Virtual screening allows prioritizing drug candidates based on their promiscuity profile and, thereby, it would derisk the drug design process.

Approaches to virtual screening, also known as *target prediction*, can be grouped into structure- and ligand-based. The structure-based methods simulate physical interactions between the compound and a biomolecular target [5] but are only applicable if the complete 3D structure of all interacting molecules are known, and they are infeasible for larger compound data bases. Ligand-based approaches predict the activity of a compound on a biomolecular target based on previous measurements [6]. Machine learning for target prediction is almost always ligand-based, for example scoring approaches like the Naive Bayes statistics [7, 8, 9], density estimation [10, 11], nearest neighbor, support vector machines, and artificial neural networks [12, 13]. Pharma industry often applies commercial software for target prediction, such as SEA [14, 15, 16], PredictFX [17], or scoring by the Naive Bayesian statistics of the PipelinePilot software.

Motivated by the success at the Merck Kaggle challenge, we assess the applicability and performance of deep networks at target prediction and compare them to state-of-the-art as well as commercial target prediction methods. Toward this goal we compiled a benchmark data set from ChEMBL, a database which resembles in-house databases of Big Pharma, though it still is considerably smaller. The Kaggle challenge comprised 15 targets, 164,024 compounds, and 11,081 features, while our ChEMBL benchmark contains more than 1,200 targets, 1.3 M compounds with 13 M ECFP12 features. The ChEMBL dataset serves to assess not only the performance, including finding new scaffolds but also whether the methods scale to pharma in-house data.

Deep learning architectures seem to be well suited for target prediction because they (1) allow for multi-task learning [18, 19, 20] and (2) automatically construct complex features [20], which for target prediction are assumed to resemble pharmacophore descriptors. First, multiple target learning

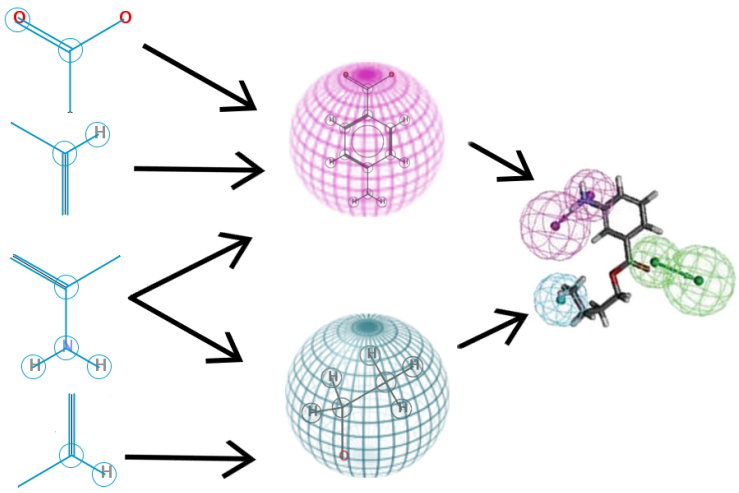

Figure 1: Hierarchical nature of fingerprint features: by combining the ECFP features we can build reactive centers. By pooling specific reactive centers together we obtain a pharmacophore that encodes a specific pharmacological effect.

has two advantages: (a) it naturally allows for multi-label information and therefore can utilize relations between targets; (b) it allows to share hidden unit representations among prediction tasks. The latter item is particularly important as for some targets very few measurements are available, therefore single target prediction may fail to construct an effective representation. In contrast, deep networks exploit representations learned across different tasks and can boost the performance on tasks with few training examples. Secondly, deep networks provide hierarchical representations of a compound, where higher levels represent more complex concepts [21]. In pharmaceutical research complex representations of compounds have a long tradition: A major goal of drug design is the identification of pharmacophores, [22, 23] which are the sets of steric and electronic properties that together enable an interaction with a target. These properties include hydrophobic regions, aromatic rings, electron acceptors or donors, which in turn can be described by substructures yielding these properties. Deep networks with ECFP12 fingerprints (chemical substructures) are ideally suited to represent properties in their first layer and in turn form pharmacophores in higher layers, as seen in Figure 1. The potential of deep learning is to find novel pharmacophores or representations of comparable complexity.

2 Experiments

Formally, the task of target prediction presents itself as follows: given a chemical compound i, we want to predict whether the compound is active on a target t. We encode this information in the binary value y_{it}, where $y_{it} = 1$ if the compound is active on a target and $y_{it} = 0$ otherwise. We are interested in predicting the behavior of a compound on m targets at the same time (for realistic tasks, m ranges in the thousands).

Each compound is represented using a number of binary features described later in this section. As training data, we are given a numerical representation $\mathbf{x_i} \in \mathbb{R}^d$ of n training compounds as well as a sparsely populated matrix $\mathbf{Y} \in \mathbb{R}^{n \cdot m}$ of previous measurements.

2.1 Dataset

We compiled a target prediction benchmark dataset out of the ChEMBL database [24], a manually curated database of bioactivity measurements, which aims to centrally store the high-quality measurements of other chemistry resources such as PubChem [25]. We extracted all pharmaceutically relevant measurements from ChEMBL. Target measurements are reported in ChEMBL as continuous values, however for a classification task we require binary labels. We thus rely on explicit

Table 1: Activity classes in the ChEMBL database.

Class	Threshold in log10(nM)	Class size
active	≤ 3.5	972,268
weakly active	> 3.5 and ≤ 4.0	143,446
weakly inactive	> 4.0 and ≤ 4.5	272,081
inactive	> 4.5	1,130,750

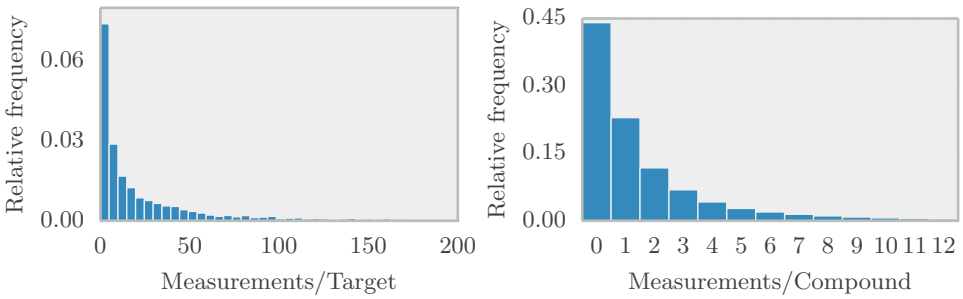

Figure 2: Properties of the extracted ChEMBL measurements: (a) Number of compounds that were measured per target., i.e., the amount of available data for each single target-prediction task. (b) Number of measurements that were taken for each compound, i.e., to how many different tasks each single compound contributes. A lot of targets had no viable measurements, and a lot of compounds had to be discarded because they weren't measured on targets of interest or the quality of the measurement was too low to warant inclusion in our final dataset.

activity comments where provided, and defined a threshold as listed in Table 1 otherwise. This yielded 2,103,018 measurements distributed across 5,069 targets and 743,336 compounds. Additionally there are 415,527 measurements which exhibit a very weak signal. These are not used for testing as their signal is no reliable but they may still be a valuable enhancement of the training set. The whole dataset as well as detailed notes on preprocessing are freely available at our homepage [1]. Figure 2 shows properties of the extracted ChEMBL measurements in terms of the number of measured targets per compound and number of compounds measured per target.

ChEMBL stores compounds as connected graphs of atoms, which we transformed into a high-dimensional binary representation using Extended Connectivity FingerPrints (ECFP12) [26] features, the currently best performing compound description in drug design applications. Each feature/fingerprint denotes the presence or absence of a certain chemical substructure. This yielded a total of 13,558,545 sparse features.

It is important that compounds which share a scaffold are not shared across training and test set, in order to guarantee that our dataset reflects the challenges of the daily drug development reality. As already mentioned, the value of virtual screening is determined by the ability to find new scaffolds with target activity. Thus, we clustered compounds using single linkage clustering to guarantee a minimal distance between training and test set. Clustering yielded 400,000 clusters which were partitioned into three folds of approximately equal size for cross-validation. Therefore the benchmark dataset automatically assesses whether previously unseen scaffolds are detected by different methods.

The number of measurements varied across several orders of magnitude between targets, with the smallest targets having only one or very few measurements and the largest ones having over 50,000 (c.f. Figure 2). Additionally, the label distribution is heavily skewed for many of the targets. This is an inherent bias of the underlying data: researchers are more likely to investigate compounds that pontentially exhibit high target activity. In order to make sure each target was realistically learnable, we discarded all targets with less than 15 samples per label, leaving 1,230 targets.

[1] `http://filled.in.after.review`

Table 2: Hyperparameters considered for the Neural Net

Hyperparameter	Considered values
Number of Hidden Units	{1024, 4096, 16356, 8192-8192}
Learning Rate	{10, 20, 30, 50}
Dropout [30]	{no, yes (50% Hidden Dropout, 20% Input Dropout)}

The performance of a classifier is evaluated by the AUC (area under the ROC curve) separately for each target. We report the mean AUC for each method and present boxplots of the AUC distribution for each method.

2.2 Methods

2.2.1 Deep Neural Network

Our network consists of one or multiple layers of ReLU hidden units [27, 28] and makes use of an Nvidia Tesla K40 GPU with 12 GB RAM to speed up the computations. However, in order to accommodate the large, multi-task target prediction task, several modifications to the commonly used neural network architectures were necessary.

Multi-Task Learning Our network consists of one or multiple layers of ReLU hidden units [27, 28], followed by one layer of 1,230 sigmoid output units, one for each molecular target or classification task. Each single training sample contributed only to a few of these tasks. Thus output units that were not active during a training sample were masked during backpropagation by multiplying their δ error by 0.

The available training data points varied greatly between the tasks, from only a handful to several thousand. However, we want to make sure that the network does not downweight the smaller tasks in favor of the larger ones. Therefore we weigh the influence of each task on the hidden layers by the amount of available training data points.

Taking these two issues into account gave us a training objective that is the weighted sum of the cross-entropies over all targets t:

$$E(\mathbf{x_i}, \mathbf{y_i}, \mathbf{m_i}) = -\sum_{t}^{T} m_{ti} \cdot w_t \left(y_{it} \log \left(\sigma_t(\mathbf{x_i}) \right) + (1 - y_{it}) \log \left(1 - \sigma_t(\mathbf{x_i}) \right) \right) \tag{1}$$

Where the weights $w_t = \frac{1}{N_t}$ ensure that across the whole training set each target has the same amount of influence on the hidden representation. The scaling factor N_t is the number of compounds that have been measured on target t[2]. The binary variable m_{ti} is 1 if sample i was measured on target t and 0 otherwise, thus masking out predictions on targets that are irrelevant for the sample at hand.

Hyperparameters We used large learning rates to compensate the scaling influence on the gradient by the w_t. In order to determine the final quality of a method, we used cross validation across the three folds of our data set. We selected the hyperparameters from Table 2 for each fold independently by using one of the two remaining folds as training set and evaluating on the other. Due to the computational demands of the task, we selected each hyperparameter separately, which underestimates the performance of the deep network. Once the hyperparameters were finalized, we trained on both of these folds together, using the number of epochs that had shown the best performance during hyperparameter selection.

Using all the 7 M inputs for the deep net were infeasible on our hardware, therefore we removed features that were present in less than 100 compounds. 43,340 input features were kept. We stored the weight parameters on a single GPU with 12 GB RAM and used mini-batches of 1,024 samples for stochastic gradient descent learning. Since storing our input data in dense format requires about

[2]Training without these weights led to networks that focused too much on the large targets

5 TB of disk space, we used a sparse storage format. However, it proved to be faster to upload a mini-batch in sparse format to the GPU and then convert it to dense format instead of using sparse matrix multiplication. Overall, training a network takes between 3 to 4 days.

2.3 Other Methods

Support Vector Machines (SVM) We used the LIBSVM implementation [31]. Since run time complexity is quadratic in the number of samples, we had to integrate the LIBSVM library into an OpenMP application that was run on a supercomputer. Moreover, to obtain further speedup, we computed all the training set similarities in advance, which took a substantial amount of memory space. In the inner cross-validation loop, we applied cross-validation again to the remaining two folds to select the best cost parameter from the set $\{0.001, 0.01, 0.1, 1.0, 2.0, 5.0, 10.0, 50.0, 100.0, 1000.0\}$ for each target independently.

Binary Kernel Discrimination (BKD) Harper et al. [11] applied Binary Kernel discrimination for target prediction. The method uses density estimators where density estimation is done for the active compounds as well as inactive compounds per target. Using these densities, the posterior probability of activity for a prediction compound are evaluated. The authors gave a formula that determines the ranking, which was implemented as part of our comparison.

Logistic Regression We implemented a L2 regularized logistic regression algorithm with a fixed bias. The fixes bias makes sure that the method is able to automatically predict the larger class – typically the inactives – correctly (in case the method has not learned anything).

Since the feature space is very high dimensional, we filtered features based on Fisher's exact test and kept the most informative features. The optimization is based on gradient descent with a line search procedure to determine the optimal step size. We again applied cross validation to select the best number of features from $\{10, 100, 1000, 10000\}$, the best bias from $\{0.0, -0.1, -1.0, -10.0, -100.0, -1000.0\}$ as well as the best prior weight for the features from $\{0.1, 1.0, 10.0, 100.0, 1000.0\}$.

k-nearest neighbour A standard k-nearest neighbour approach was implemented, where the ranking of compounds concerning target activity is given by the ratio of actives and inactives in the neighbour set. We used Tanimoto similarity to determine the similarity of a prediction compound to a training compound. The Hyperparameter for the number of neighbours was selected from the set $\{1, 3, 5, 10\}$.

Parzen-Rosenblatt Lowe et al.[10] used the Parzen Rosenblatt kernel density estimator to model the density of active molecules for a target. A Gaussian kernel was used to measure the similarity between a compound to be predicted and an active compound from the target set compounds. The final prediction score for a target is based on computing the conditional distribution of a particular target given the compound to be predicted by using Bayes theorem and marginalizing over all targets. The best bandwidth of the Gaussian kernel was determined by cross validation for each target from the set $\{0.1, 1.0, 10.0, 100.0, 1000.0, 10000.0\}$.

Pipeline Pilot Bayesian Classifiers (PNPBC) The commercial product "Pipeline Pilot" [7] uses a Naive Bayes statistics based approach, which essentially contrasts the active samples of a target with the whole (background) compound database. It does not explicitly consider the samples labelled as incative. Laplacian-adjusted probability estimates for the features lead to individual feature weights which are finally summed up to give the prediction. We re-implemented the "Pipeline Pilot" Naive Bayes statistics in order to use it on a multi-core supercomputer, which allowed us to compare this method on our benchmark dataset.

Similarity Ensemble Approach (SEA) SEA [14, 15, 16] is based on the idea that two targets are similar if the ligand sets of a target are similar to one another. The similarity of two ligand sets is computed by the sum of ligand pair similarities that exceed a certain threshold. The ligand pair similarity is measured by Tanimoto similarity. To correct for size or chemical composition bias a correction technique is intrudiced, which is based on the similarity obtained from randomly drawn ligand sets is. This leads to z-scores for similarity between the sets. It is argued that the z-scores

conform an extreme value distribution. Using this extreme value distribution the probability that a compound is active on a certain target is calculated by assuming that one of the two ligand sets consists only of the compound to predict. We implemented the SEA method efficiently for using it on a multi-core supercomputer, enabling us to compare it to the other target prediction methods.

2.4 Results

Table 3: Performance of target prediction methods in terms of mean AUC across targets. The first column gives the method, the second column the AUC value, and the third column the p-value of a paired Wilcoxon test with the The alternative hypothesis that the deep neural network has on average a larger AUC than the other method.

Method	**AUC**	p**-value**
Deep network	**0.830**	
SVM	0.816	1.0e-07
BKD	0.803	1.9e-67
Logistic Regression	0.796	6.0e-53
k-NN	0.775	2.5e-142
Pipeline Pilot Bayesian Classifier	0.755	5.4e-116
Parzen-Rosenblatt	0.730	1.8e-153
SEA	0.699	1.8e-173

Table 3 shows the mean AUC values across 1,230 targets for each of the classifiers we used. The deep neural network significantly outperformed its competitors, including two commercial methods with respect to the area under ROC curve (AUC) averaged over the prediction tasks, i.e. targets. Other well-established machine learning methods that could be scaled to the data set, such as SVMs, also performed better than the commercial methods.

The neural net achieves an AUC ≥ 0.8 on 813 out of the 1,230 targets, or $\approx 66\,\%$ of the time. The median AUC lies at 0.8588. On 12 targets we achieve perfect prediction accuracy (AUC = 1.0). This is in stark contrast to current commercial solutions, where the median AUC lies below 0.8. As shown in Figure 3, almost all methods suffered from severe outliers. Of the methods that achieved an average AUC of over 0.8, the Deep Network has the least severe outliers. We hypothesize that the network could leverage its shared hidden representation to predict tasks which are difficult to solve when tackled in isolation.

3 Conclusion

Our experiments have shown that deep neural networks outperformed all other methods concerning the AUC and can be a valuable tool in industrial drug design. In particular, deep nets surpassed existing commercial solutions by a large margin. On many targets it achieves nearly perfect prediction quality which qualifies it for usage as virtual screening device. The results of machine learning methods are even better on Big Pharma in-house data (not presented due to confidential reasons). The reason for this improved performance on in-house data is that it is better balanced than our ChEMBL benchmark data and it has much more inactive compounds. In summary, deep learning provides the opportunity to establish virtual screening as a standard step in drug design pipelines.

Acknowledgements

This work was supported in part by a grant from the *Mr.SymBioMath* project being funded by the European Union's Seventh Framework Programme for research, technological development and demonstration as an Industry-Academia Partnerships and Pathways (IAPP) project under grant agreement number 324554. The authors also gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU used for this research.

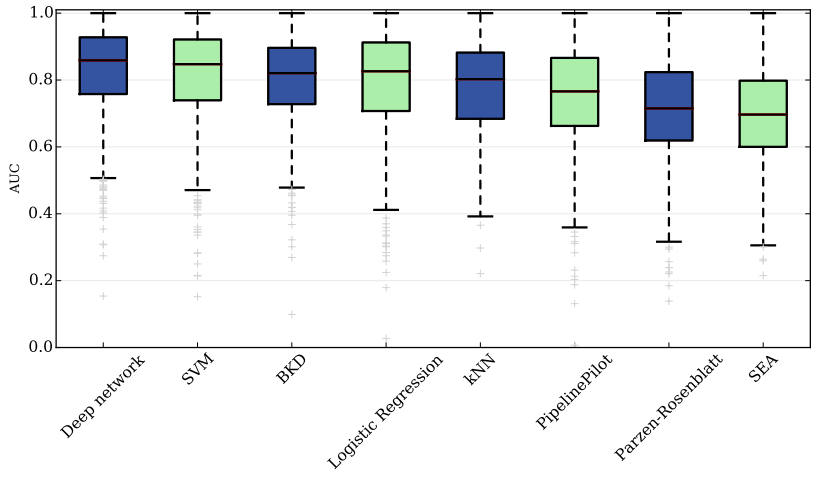

Figure 3: Distribution of AUC values across targets for each classifier.

References

[1] J. Arrowsmith, "Trial watch: phase III and submission failures: 2007–2010," *Nature Reviews Drug Discovery*, vol. 10, no. 2, pp. 87–87, 2011.

[2] J. W. Scannell, A. Blanckley, H. Boldon, and B. Warrington, "Diagnosing the decline in pharmaceutical r&d efficiency," *Nature Reviews Drug Discovery*, vol. 11, no. 3, pp. 191–200, 2012.

[3] M. Hay, D. W. Thomas, J. L. Craighead, C. Economides, and J. Rosenthal, "Clinical development success rates for investigational drugs," *Nature Biotechnology*, vol. 32, no. 1, pp. 40–51, 2014.

[4] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, and A. L. Schacht, "How to improve r&d productivity: the pharmaceutical industry's grand challenge," *Nature Reviews Drug Discovery*, vol. 9, no. 3, pp. 203–214, 2010.

[5] D. B. Kitchen, H. Decornez, J. R. Furr, and J. Bajorath, "Docking and scoring in virtual screening for drug discovery: methods and applications," *Nature Reviews Drug discovery*, vol. 3, no. 11, pp. 935–949, 2004.

[6] J. L. Jenkins, A. Bender, and J. W. Davies, "In silico target fishing: Predicting biological targets from chemical structure," *Drug Discovery Today: Technologies*, vol. 3, no. 4, pp. 413–421, 2007.

[7] X. Xia, E. G. Maliski, P. Gallant, and D. Rogers, "Classification of Kinase Inhibitors Using a Bayesian Model," *Journal of Medicinal Chemistry*, vol. 47, pp. 4463–4470, Aug. 2004.

[8] F. Nigsch, A. Bender, J. L. Jenkins, and J. B. O. Mitchell, "Ligand-target prediction using winnow and naive bayesian algorithms and the implications of overall performance statistics," *Journal of Chemical Information and Modeling*, vol. 48, no. 12, pp. 2313–2325, 2008.

[9] H. Y. Mussa, J. B. O. Mitchell, and R. C. Glen, "Full "Laplacianised" posterior naive Bayesian algorithm," *Journal of Cheminformatics*, vol. 5, pp. 37+, Aug. 2013.

[10] R. Lowe, H. Y. Mussa, F. Nigsch, R. C. Glen, and J. B. Mitchell, "Predicting the mechanism of phospholipidosis," *Journal of Cheminformatics*, vol. 4, no. 1, p. 2, 2012.

[11] G. Harper, J. Bradshaw, J. C. Gittins, D. V. S. Green, and A. R. Leach, "Prediction of biological activity for high-throughput screening using binary kernel discrimination," *Journal of Chemical Information and Computer Sciences*, vol. 41, no. 5, pp. 1295–1300, 2001.

[12] E. Byvatov, U. Fechner, J. Sadowski, and G. Schneider, "Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification," *Journal of Chemical Information and Computer Sciences*, vol. 43, pp. 1882–1889, Sept. 2003.

[13] R. Lowe, H. Y. Mussa, J. B. O. Mitchell, and R. C. Glen, "Classifying molecules using a sparse probabilistic kernel binary classifier," *Journal of Chemical Information and Modeling*, vol. 51, no. 7, pp. 1539–1544, 2011.

[14] M. J. Keiser, B. L. Roth, B. N. Armbruster, P. Ernsberger, J. J. Irwin, and B. K. Shoichet, "Relating protein pharmacology by ligand chemistry," *Nature Biotechnology*, vol. 25, pp. 197–206, Feb. 2007.

[15] M. J. Keiser, V. Setola, J. J. Irwin, C. Laggner, A. I. Abbas, S. J. Hufeisen, N. H. Jensen, M. B. Kuijer, R. C. Matos, T. B. Tran, R. Whaley, R. A. Glennon, J. Hert, K. L. Thomas, D. D. Edwards, B. K. Shoichet, and B. L. Roth, "Predicting new molecular targets for known drugs.," *Nature*, vol. 462, pp. 175–181, Nov. 2009.

[16] M. Keiser and J. Hert, "Off-target networks derived from ligand set similarity," in *Chemogenomics* (E. Jacoby, ed.), vol. 575 of *Methods in Molecular Biology*, pp. 195–205, Humana Press, 2009.

[17] Gregori-Puigjane, Elisabet, Mestres, and Jordi, "A Ligand-Based approach to mining the chemogenomic space of drugs," *Combinatorial Chemistry & High Throughput Screening*, vol. 11, pp. 669–676, Sept. 2008.

[18] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, p. 41–75, 1997.

[19] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at microsoft," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 8604–8608, 2013.

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives.," *IEEE Trans Pattern Anal Mach Intell*, Feb 2013.

[21] Y. Bengio, "Deep learning of representations: Looking forward," in *Proceedings of the First International Conference on Statistical Language and Speech Processing*, SLSP'13, (Berlin, Heidelberg), pp. 1–37, Springer-Verlag, 2013.

[22] L. Kier, *Molecular orbital theory in drug research*. Medicinal chemistry, Academic Press, 1971.

[23] S.-K. Lin, "Pharmacophore perception, development and use in drug design. edited by osman f. güner," *Molecules*, vol. 5, no. 7, pp. 987–989, 2000.

[24] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington, "ChEMBL: a large-scale bioactivity database for drug discovery," *Nucleic Acids Research*, vol. 40, pp. gkr777–D1107, Sept. 2011.

[25] E. Bolton, Y. Wang, T. PA, and B. SH, "Pubchem: Integrated platform of small molecules and biological activities.," *Annual Reports in Computational Chemistry*, vol. 4, pp. 217–240, 2008.

[26] D. Rogers and M. Hahn, "Extended-connectivity fingerprints.," *Journal of Chemical Information and Modeling*, vol. 50, pp. 742–754, May 2010.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

[28] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, pp. 315–323, 2011.

[29] P. Baldi, P. Sadowski, and D. Whiteson, "Deep learning in high-energy physics: Improving the search for exotic particles," Feb. 2014.

[30] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," July 2012.

[31] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.