DeltaDelta Neural Networks for Lead Optimization of Small Molecule Potency

José Jiménez[1], Laura Pérez-Benito[2, 3], Gerard Martínez-Rosell[4], Simone Sciabola[5], Rubben Torella[6], Gary Tresadern[3], and Gianni De Fabritiis[1, 4, 7, *]

[1]Computational Science Laboratory, Universitat Pompeu Fabra, PRBB, Carrer del Dr. Aiguader 88, Barcelona, 08003, Spain.
[2]Laboratori de Medicina Computacional, Unitat de Bioestadística, Facultat de Medicina, Universitat Autònoma de Barcelona, Spain.
[3]Janssen Research and Development, Turnhoutseweg 30, 2340 Beerse, Belgium.
[4]Acellera, PRBB, Carrer del Dr. Aiguader 88, Barcelona, 08003, Spain.
[5]Biogen Chemistry and Molecular Therapeutics, 115 Broadway Street, Cambridge, MA 02142, USA.
[6]Pfizer I&I, 610 Main Street, Cambridge, MA 02139, USA.
[7]Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, 08010 Barcelona, Spain.
[*]E-mail: gianni.defabritiis@upf.edu

Abstract

The capability to rank different potential drug molecules against a protein target for potency has always been a fundamental challenge in computational chemistry due to its importance in drug design. While several simulation-based methodologies exist, they are hard to use prospectively and thus predicting potency in lead optimization campaigns remains an open challenge. Here we present the first machine learning approach specifically tailored for ranking ranking congeneric series based on deep 3D-convolutional neural networks. Furthermore we prove its effectiveness by blindly testing it on datasets provided by Janssen, Pfizer and Biogen totalling over 3246 ligands and 13 targets as well as several well-known openly available sets, representing one the largest evaluations ever performed. We also performed online learning simulations of lead optimization using the approach in a predictive manner obtaining significant advantage over experimental choice. We believe that the evaluation performed in this study is strong evidence of the usefulness of a modern deep learning model in lead optimization pipelines against more expensive simulation-based alternatives.

In the lead optimization phase of drug discovery, the chemical structure of a molecule is typically modified by a medicinal chemist team with the intent of improving its potency, selectivity, and many other pharmacokinetic and toxicological parameters [1–3]. These modifications result in congeneric series, a set of ligands with few atom changes between them, usually around a unique or small number of different scaffolds for which there are experimental structures of the complex with the target protein. Series range from few hundreds to thousands of compounds and require considerable human, time and financial resources for synthesis and assays. It is therefore of great value to have *in silico* predictive tools to accelerate this process. Series typically feature very small potency differences, which in turn is a challenge for predictors, as having what could be considered a low error in other scenarios (e.g. below 1 kcal/mol) is not a guarantee for successful ranking.

It is therefore common to focus on relative binding free energy (RBFE) simulation methods [4–13], where the difference in affinity between two ligands is computed using a thermodynamic cycle that alchemically perturbs only the small region associated with the changing atoms. RBFE methods have shown good results in several studies, with accuracy close to 1 kcal/mol and reasonable correlations. Despite this, these methods suffer from several issues, such as system preparation, treatment of waters, force-field selection, protein flexibility and computational cost, making their prospective application difficult in practice [14]. On the other side, many empirical [15, 16], knowledge-based [17, 18] and machine learning [19–24] scoring functions have been designed for the task of predicting absolute binding affinities. They mostly tackle the problem in a regression setup, where the binding affinity is to be predicted using a set of protein-ligand descriptors, modelling the interaction among both. The fact that they model absolute affinities and are trained on very chemically diverse bodies of data, such as iterations of the PDBbind [25] database, limits their applicability when predicting small structural differences between two ligands, such in the congeneric series case. While other machine learning approaches have been presented for this task [26–28], here we propose a modern 3D-convolutional-neural-network-based continuous learning approach for relative binding affinity prediction in congeneric series and show strong predictive power using multiple blind benchmarks as well as public datasets at negligible computational costs. This study serves as a very large evaluation of a modern machine-learning pipeline for lead optimization in a real-life drug

discovery scenario, thanks to the joint collaboration with several pharmaceutical companies.

The BindingDB protein-ligand validation sets [29] were used to pretrain our models, see methods for details. For testing, we also extracted well-known publicly-available literature test sets [30] used for benchmarking RBFE calculations. Furthermore we include a recent freely-available BRD4 bromodomain dataset [31]. In regards to internal pharmaceutical data, we tested on five different congeneric series from Janssen R&D. Three chemical series (sets 1, 2 and 3) were phosphodiesterase 2 (PDE2) inhibtors with bioactivity versus PDE2, PDE3, and PDE10 [32,33] (publication number WO2018083103A1), the fourth series were proto-oncogene tyrosine kinase (ROS1) inhibitors (publication number WO2015144799A1) and the final beta-secretase 1 (BACE1) inhibitors [34]. We tested six congeneric series with Pfizer, three of which target a kinase, and the remaining an enzyme, a phosphodiesterase (PDE) and an activator of transcription. The sizes of these vary from 93 molecules up to 362, for a total of 955 tested compounds. Lastly, Biogen tested the proposed procedure on two different series, composed of 196 and 220 analogues targeting a tyrosine-protein kinase and a receptor-associated kinase, respectively. All the tests regarding internal pharmaceutical data were carried out blindly by providing fully-containerized software to our collaborators, who executed the application and reported corresponding results. Furthermore only one pretrained model model was provided without any opportunity to overfit to each specific test set. The size of the sets presented here allow, to the best of our knowledge the largest evaluation yet of a modern machine learning pipeline in lead optimization.

We have recently reported a machine learning approach that can learn based on 3D features of the binding site interactions [20]. A similar encoding was used here that represents the protein-ligand binding by voxelizing both using a 24Å pocket centered box. The contribution of each atom to each voxel is inversely proportional to their euclidean distance r and the van der Waals radius r_{vdw} of the first (see methods). We use several *channels* for both protein and ligand, in the sense that the atomic contribution to each voxel depends on their type, which are thoroughly defined in the methods section.

The neural network we propose has a novel zero-symmetric architecture whose main building blocks are 3D-convolution operations. Convolutional neural network (CNN) architectures have become the de-facto workhorse in computer vision problems [35–37], providing state-of-the-art performance. Following this success, many applications in bioinformatics and computational chemistry followed [38–47]. In this work we focus on predicting relative affinities for close analogues in lead optimization, therefore, our approach is to build a network whose input is a pair of ligand binding voxelized representations belonging to the same series. A two-input convolutional neural network is designed, with fixed weights on both legs (see methods). The inputs are forwarded through several convolution and pooling operations and then flattened into a 192-dimensional latent vector. The symmetry property of relative binding affinity requires that inverting the order of the ligands should change the sign of the predicted value. We embed such symmetry in the network by computing the difference between these latent vectors, representing a latent description of difference in binding. A final linear layer with no bias is then applied to the result of this difference, ensuring zero-symmetry by design and producing the desired predicted difference in affinity. Calculating relative affinities from an absolute prediction leads to concatenation of the errors from two separate predictions. Here, the model itself only focuses on what contributes to the differences, and errors of absolute binding are canceled.

In the proposed continuous-learning approach, we explicitly use the fact that congeneric series are sequentially generated in a lead optimization campaign, and follow an incremental training and testing procedure. For each congeneric series at a given time the affinity of previously tested ligands is known experimentally: differences for these are taken as training data, while for test data we predict differences between unknown and known ones. While this approach is less ambitious than having a predictor for relative affinity with no experimentally tested data (such as a physical-based model), its applicability is general, since it is the common scenario that medicinal chemists face in lead optimization campaigns. The training for the BindingDB sets starts with a reference structure in each series, for which we take the crystal structure ligand if available or the structure with the lowest average maximum common substructure (MCS) distance to the rest. Ligands from the rest of the series are then sequentially added in a random order. It is well known that either a random [48] or scaffold-based training test split produce overoptimistic results when testing machine-learning algorithms on activity benchmarks. Since the industrial datasets in our study include a compound creation time-stamp, we also evaluate a more realistic temporal split [49], where at each training step we consider the first n tested ligands and the differences of the posterior ones against the first are taken. The performance of the machine learned models is reported as the root mean squared error (RMSE) and either Pearson's correlation coefficient R or Spearman's ρ between experimental and predicted affinity differences. We note that in all blind tests a single model was provided, and no explicit attempt to optimize hyperparameters in each set was made.

We first present results concerning our validation on the 495 protein-series datasets from the BindingDB, where the proposed model achieves an average correlation coefficient above 0.4 and an RMSE below 1.25 (pIC_{50} units) even when only one binding-energy difference is taken per congeneric series (Supplementary Fig. 1). This suggests that the method works reasonably well in the very low-data scenario, such as the beginning of a lead optimization campaign. A noticeable performance boost is seen as more differences are included in training, with a correlation coefficient above 0.62 and an RMSE below 1.05 when another four different ligands from the same congeneric series are known in advance, with performance plateauing beyond five additional training ligands. A comparison against an absolute affinity model is also provided (i.e. one of the legs of the architecture), where as expected it can be appreciated that it performs considerably worse than its relative counterpart.

Now we present results on the Wang *et. al.* [30] and BRD4 inhibitor datasets [31]. In this and the rest of cases, we pretrained a model with all difference pairs available

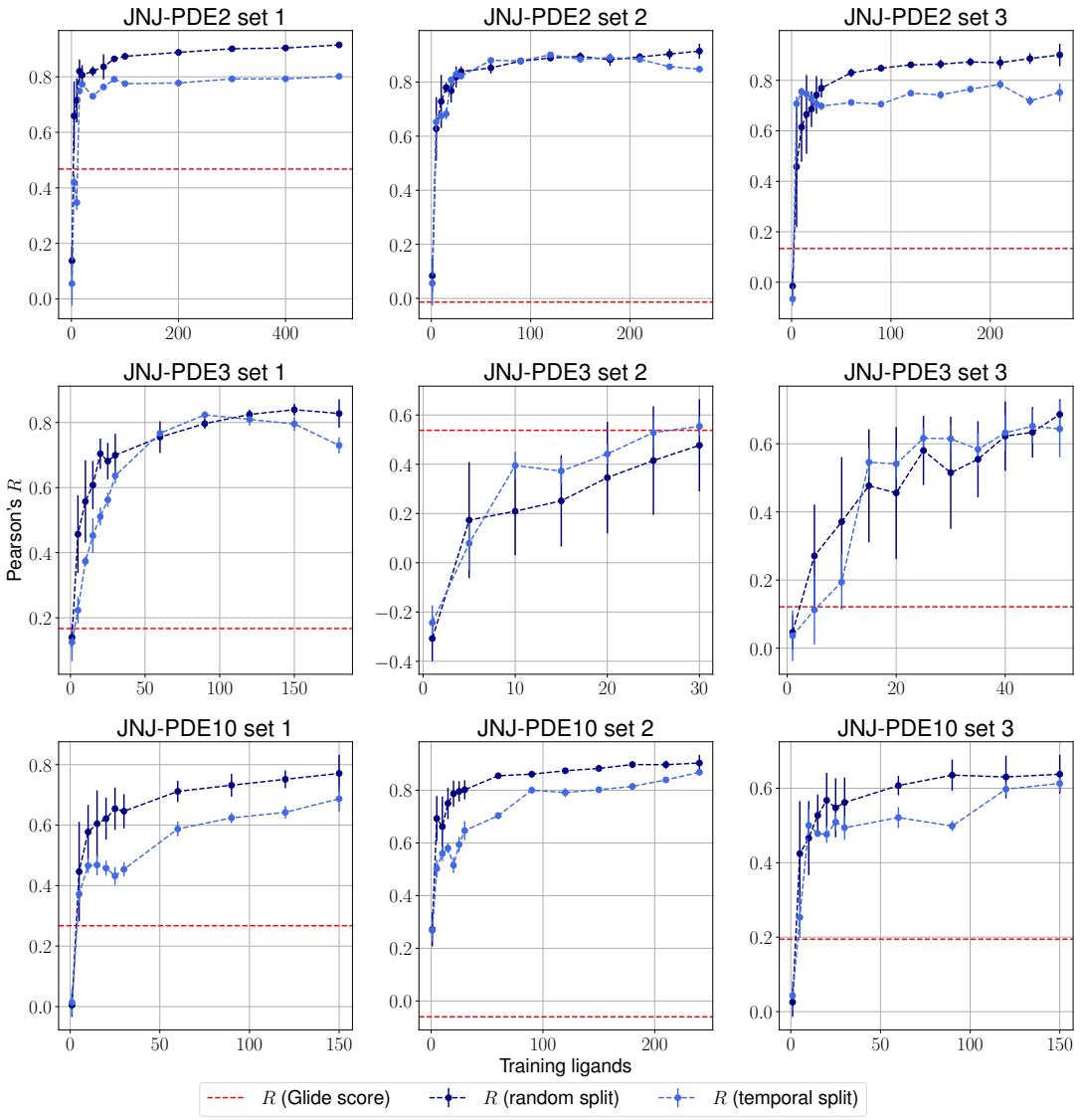

Figure 1: Average Pearson's correlation coefficient R (± 1 standard deviation) based on 25 independent runs on different sets for the Janssen PDE2, PDE3 and PDE10 targets.

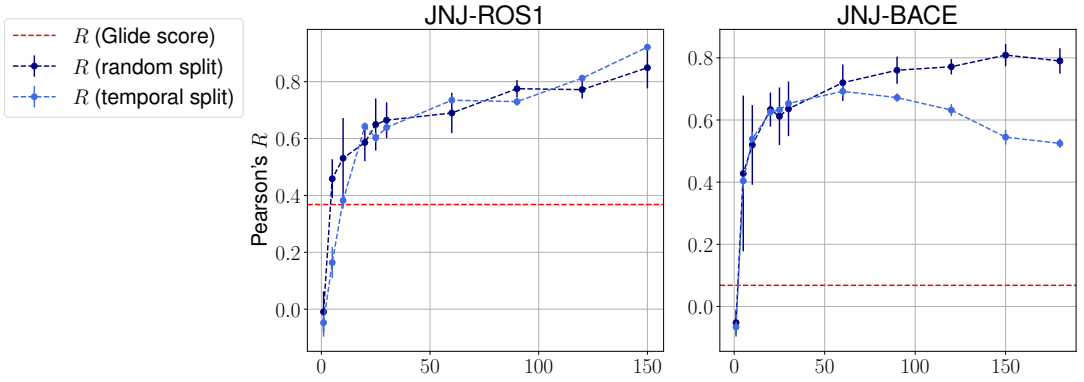

Figure 2: Average Pearson's correlation coefficient R (± 1 standard deviation) based on several independent runs on two sets for the Janssen ROS1 and BACE targets.

in the BindingDB database, which provides a prior for further fine-tuning. We then mixed new available data as training in each sequential iteration of each set with the rest of the BindingDB database for only 3 epochs, significantly reducing computational overhead. A FEP baseline provided by Wang *et. al.* [30] is used for comparison. The model efficiently interpolates differences for unseen ligands, achieving considerably high correlation coefficients and low errors in all series with as few as 3-4 additional ligands and associated activity pairs, surpassing in many cases the much more expensive FEP baseline (Supplementary Fig. 2). For instance, for the MCL1 target, after testing 3 ligands, the correlation coefficient is above 0.8, surpassing the FEP baseline, and the RMSE is

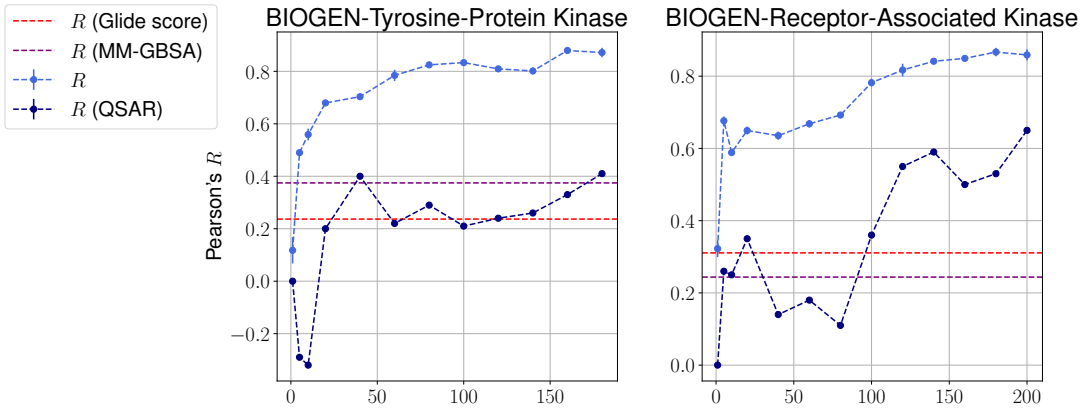

Figure 3: Results over 5 runs on Biogen's Tyrosine-Protein Kinase and Receptor-Associated Kinase using a temporal split, and MM-GBSA and QSAR random forest pipelines as baselines.

Table 1: Spearman's ρ performance results between experimental and predicted absolute affinities provided by Pfizer I&I, where other empirical, simulation, and machine-learning based affinity prediction methods are compared on several congeneric series. Performance is poor for most tested model except for the sequential approach proposed here, with Pearson correlations averaging over 0.5 with as few as 10% used analogues from the congeneric series at hand. * Calculated LogP as available in rdkit

Target	**# ligands**	**Mol. Weight (ρ)**	**cLogP (ρ) ***	**MM-GBSA (ρ)**	K_{DEEP} **(ρ)**	**This work (10% training, ρ)**	**This work (20% training, ρ)**	**This work (30% training, ρ)**
Kinase #1	362	0.19	0.06	0.56	0.42	0.49	0.64	0.73
Kinase #2	106	0.1	0.28	0.25	0.25	0.25	0.41	0.51
Kinase #3	95	0	0.04	0.25	-0.27	0.3	0.3	0.31
Enzyme	93	0.43	0.24	0.01	0.49	0.43	0.26	0.59
Phosphodiesterase	100	0.37	0.36	0.67	0	0.49	0.64	0.73
Activator of transcriptions	199	0.13	0.08	0.66	0.29	0.72	0.84	0.94
Weighted avg.		0.19	0.14	0.47	0.25	**0.49**	**0.59**	**0.69**
Simple avg.		0.2	0.18	0.4	0.18	**0.45**	**0.52**	**0.64**

below 1.2 (pIC_{50} units).

The same evaluation procedure was taken for the compounds available in the Janssen PDE sets (Fig. 1 and Supplementary Fig. 3) for both a random and a temporal split, where a baseline against Glide score [50] is also added. Excellent performance was seen on a random split given enough training data, and as expected, although the temporal split performance is lower, it is still sufficiently high to be used in a real-life prospective lead optimization scenario. For instance, for the first PDE2 activity set after 20 ligands sorted by time, the Pearson's correlation coefficient R and RMSE were 0.77 and 1.35 (in pIC_{50} units) respectively. Results for the ROS1 and BACE sets, show a similar trend and insights (Fig. 2 and Supplementary Fig. 4). Furthermore, we also provide a type of split where only differences among the most chemically close ligands are predicted, based on ECFP4 fingerprint similarity, as available in rdkit. That is, in each training step we predict from the remaining untested pool of ligands those that are closest to the ones in our training set, with the intention of resembling a real-life lead optimization RBFE scenario, typically applied to close analogues. Split-based results on fingerprint similarity for the first PDE2 set (Supplementary Fig. 5), show that after 20 ligands sorted by chemical similarity the R and RMSE were 0.83 and 1.12 (in pIC_{50} units). These suggest better performance in this scenario than the proposed temporal split, and closer to the random one.

We then present the results provided by Pfizer using a temporal split in Table 1, where specific target names cannot be disclosed. We compare such results with several baselines such as molecular weight, cLogP, a MM-GBSA pipeline [51, 52] and deep-learning absolute affinity predictor K_{DEEP} [20], trained on the v.2016 iteration of the PDBbind database. The model proposed here performs considerably better than the rest when given only 10% of the training data, again highlighting the importance of incrementally training these on the congeneric series of interest. An exception, however, is found in the Kinase #3 series, for which no significant improvement is observed when providing extra training data. We provide results using a temporal split for the last two congeneric series provided by Biogen, for which we also compare against several baselines: (a) Glide score, (b) an MM-GBSA pipeline, and (c) a standard QSAR approach using MACCS, ECFP4 and rdkit descriptors with a random forest model (Fig. 3 and Supplementary Fig. 6). In these our model reveals similar conclusions, significantly outperforming all baselines. Curiously, it can also be seen that the proposed method does not perform significantly worse than the aforementioned baselines in the second target when no training data is used. When some is used, such as only 5 analogues, our proposed machine-learning model significantly outperforms all baselines.

One aim of our study was to test whether machine-learning driven relative affinity predictions could efficiently identify key high potency compounds in a close to real-life lead optimization scenario, by retrospectively comparing them to the experimental order of synthesis. With some of the large industrial datasets it was possible

Table 2: Simulation-based benchmark results over 10 independent runs for the different datasets. We show the amount of molecules the model is allowed to pick at each synthesis epoch, the experimental order of the compound with the highest affinity in the series, the average synthesis epoch our model found said molecule, the total necessary sampled ligands the proposed model has chosen before the target compound, and the sampling advantages over the experimental and random orders.

Target	Set	# ligands	Chosen per synthesis epoch	Experimental order	Found at synthesis epoch	Total sampled ligands	Advantage over experimental choice	Advantage over random choice
PDE2	1	900	10	766	12.2	132	**634**	**318**
PDE2	2	303	10	61	1	20	**41**	**131.5**
PDE2	3	278	10	253	5.9	69	**184**	**70**
ROS1	-	165	10	73	3.1	41	**32**	**41.5**
BACE	-	229	10	190	20.8	218	-28	-103.5

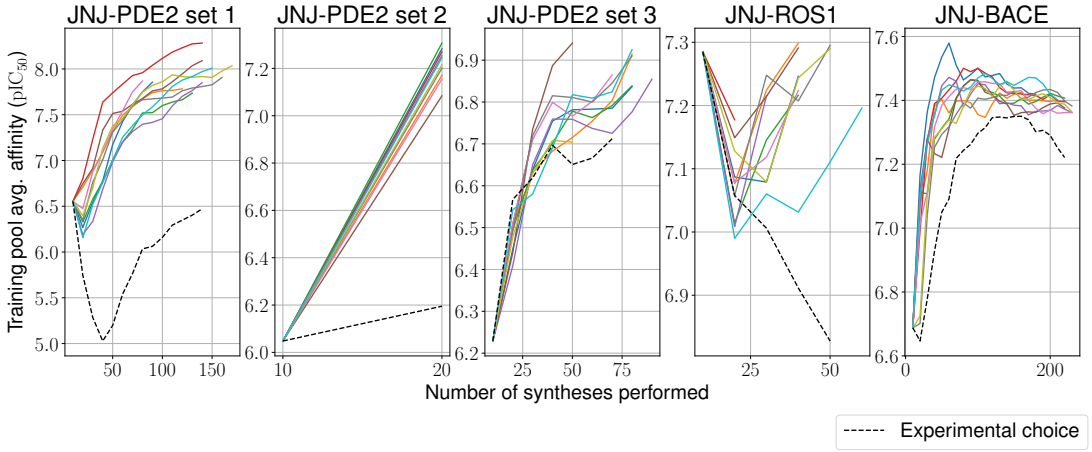

Figure 4: Average model-picked training set affinity per number of compounds synthesized for the Janssen PDE2, ROS1 and BACE sets, as well as a baseline based on the actual experimental choice order of compounds

to test this and we used the most active compound as a surrogate interesting lead molecule. The model is trained on the first experimentally tested compounds, and then is incrementally trained by choosing from the remaining ones based on a upper confidence bound (UCB)-like criterion [53], described in the methods section. We stop the procedure once the model retrieves the analogue with the highest associated affinity, and compare this with its original synthesis experimental order in its corresponding series. We present results for this simulation-based benchmark in Table 2. In 4 out of 5 sets our proposed model is able to reach the compound with the highest affinity faster than its experimental order or by random selection. Surprisingly, in all ten independent runs of the second set for the PDE2 target, the compound with the highest affinity was found after only a single synthesis epoch. Furthermore, one would expect the average affinity in the training set to increase at each synthesis epoch (as the model is tasked to pick compounds with increasingly higher UCB). This is the case for 4 out of 5 sets again (Fig. 4), with the exception of the ROS1 target, which shows a non-monotonic trend, albeit its model reaches the compound with highest affinity before its experimental order. In all tested cases, the average training pool affinity for the ligands selected by the model is higher than experimental choice. Overall results are very promising and suggest that the proposed method could be applied in a prospective scenario successfully. Particularly, in the first PDE2 set, we were able to reach potent compounds synthesizing up to six times less molecules than the baseline method used by the medicinal chemistry team.

In this work we have designed and tested a deep-learning based model for the task of predicting relative binding affinity predictions in congeneric series. This work provides evidence that the method is able to efficiently rank compounds as shown by an evaluation on both publicly available and industrial data and can be of use by computational and medicinal chemists in early drug-discovery projects by providing informed choices of future compounds to synthesize, as suggested by our simulation-based benchmark. The accuracy of the method heavily depends on the amount of available data but can be trained and applied in minutes on a single GPU, offering a substantial improvement in performance compared with physics-based RBFE calculations which can take days for a small number of analogues. While the results presented here are encouraging, it is important to note that they remain retrospective: a proper prospective validation of the model, which would entail chemists synthesizing compounds according to the decisions taken by the trained model, remains a topic of future study. In the long term, however, we expect that improving molecular simulations accuracy [54, 55] by the integration of physics and machine learning approaches would produce a more convenient approach for engineering drug discovery. In the meantime, methods such as the one proposed here provide accurate performance at a fraction of the computational cost of other approaches.

Methods

Data filtering and cleaning. Out of the total 645 available congeneric series available in BindingDB, 495 with IC_{50} affinity values were extracted and processed for further evaluation, as it was the unit with most data available, containing a diverse set of targets. The majority of these sets encompass a single protein-ligand crystal structure, the rest of the ligands modelled against the reference using the Surflex docking software [56]. We then assign each protein structure in the database to a family cluster using a 90% sequence similarity threshold, as per PDB conventions [57]. For each series in the same protein cluster we use a maximum common substructure (MCS) protocol as available in rdkit [58] to remove identical ligands. This procedure ensures that the same ligand is not repeated against similar targets, avoiding potential overfitting problems and overoptimistic evaluations [59]. Affinity values were log-converted to avoid target scaling issues ($pIC_{50} = -\log_{10} IC_{50}$). Ligands that could not be read by rdkit were removed. Histograms of the number of ligands and their affinity range per series are provided in Supplementary Fig. 7, with the average available number of ligands per series being 8.84. In the Schrödinger and BRD4 sets, since only ΔG (per kcal/mol) information was available, we converted affinity values to the pIC_{50} range assuming non-competitive binding. Descriptive information on these series is provided in Supplementary Table 1. Compounds provided by Janssen were docked using a common scaffold structure via the Glide software. These congeneric series range from 48 up to a 900 different compounds with varying affinity ranges (Supplementary Table 2).

Descriptor calculation. The contribution of each atom to each voxel is assigned according to a pair correlation function defined by:

$$n(r) = 1 - \exp\left(-\left(\frac{r_{\text{vdw}}}{r}\right)^{12}\right) \tag{1}$$

We define several *channels* for both protein and ligand, in the sense that the atomic contribution to each voxel depends on their type. For the protein we define eight pharmacophoric-like descriptors, as detailed in Supplementary Table 3. For the ligands we use a simpler representation based on atom types contained in the set $\{\text{C}, \text{N}, \text{O}, \text{F}, \text{P}, \text{S}, \text{Cl}, \text{Br}, \text{I}, \text{H}\}$, for a total of 18 stacked channels. We note that there is no particular reasoning behind this choice of descriptors other than they showed promising practical performance in previous studies. The proposed network architecture could easily be adapted to work with other representations.

Network architecture and training. Neural networks are universal function approximators [60,61], the output of each neuron being a dot product of some inputs $\boldsymbol{x}$ with some weights $\boldsymbol{w}$ plus a bias b, followed by a non-linearity f:

$$\phi = f\left(\sum_i w_i x_i + b\right). \tag{2}$$

Regular feed-forward neural networks, however, do not scale well when the input is high dimensional (as in images, or in this case atomic interactions). CNNs on the other hand are specifically designed for handling lattices, where local spatial information needs to be preserved. While a feed-forward network would ignore such interactions, a convolutional one arranges its neurons spatially, and only connects locally to the output of the previous layer. In practice, building a neural network from scratch entails many architectural choices, and for this work, since on average the depth of the network should be roughly proportional to data size, we chose to keep our network as shallow as possible.

A schema of our architectural choice is provided in Fig. 5 is provided in the Supplementary Information. It features two convolution operations with a kernel size of 3 in each leg, followed by a max-pooling operation, and finally another convolution operation with the same kernel size for both before flattening and performing the latent difference between analogues. The ReLU activation function was used for all layers in the network except for the last, which does not feature one. We include a dropout layer in the end to control for overfitting. Xavier initialization was used for the weights. Training is performed using the Adam stochastic gradient descent optimizer [62] with standard hyperparameters ($\beta_1 = .9, \beta_2 = .999, \epsilon = 10^{-4}$) using a batch size of 32 samples for 50 epochs. Furthermore, given a set of relative binding predictions, its absolute counterparts can always be retrieved given a single experimentally determined absolute reference, such as the one provided by a lead. If more than one is available, absolute affinities can be computed towards each, in practice providing a predictive absolute affinity distribution, whose average can then be interpreted as a maximum a posteriori (MAP) estimate of the absolute affinity and its standard deviation as a measure of its uncertainty, given the current model state.

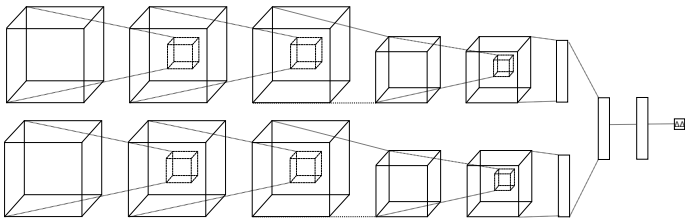

Figure 5: Architecture of the proposed model. A two-legged neural network with tied weights was constructed, and a pair of protein-ligand voxelization is feed-forwarded through it to later perform a latent space difference.

All the models here were developed using the PyTorch package for tensor computation and neural network training [63].

Upper-confidence bound criteria. The UCB-like criterion is defined as:

$$\text{UCB} = \mu(x) + \beta\sigma(x), \tag{3}$$

where μ and σ are the average and standard deviation predicted absolute affinities provided by the model for ligand x and β is a user-chosen factor controlling the balance between exploitation and exploration, that we fix in our study to $\beta = 1.64$.

Data availability. BindingDB, Wang *et al.* and Mobley *et al.* set results are available upon reasonable request.

Code availability. Python code for generating the proposed featurization is available within the open-source HTMD software [64]. The code of the network architecture in a PyTorch implementation is provided in the Supplementary Information. An implementation of this application is available through the PlayMolecule.org repository of applications, where users can freely submit their protein in PDB format and two sets of the same congeneric series, for training and validation respectively in SDF format. Depending on the size of these last two, training and prediction time may vary, as the order of data for training increases by $\frac{n(n-1)}{2}$, and for testing nm factors, where n and m are the number of training and testing instances respectively. At the moment, predictions are limited to a default total of a 1000 molecules per congeneric series, with runtimes averaging and hour on a modern GeForce 1080Ti GPU. Larger experiments can be arranged for users willing to run more computationally demanding experiments.

Acknowledgments

The authors thank Acellera Ltd. for funding. G.D.F. acknowledges support from MINECO (BIO2014-53095-P), MICINN (PTQ-17-09079) and FEDER. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675451 (CompBioMed project).

Author contributions

J.J. and G.D.F. developed the method and designed the testing methodology. L.P.B. and G.T. tested models on internal Janssen data and provided insight into the validation procedure. S.S. and R.B. tested the method on Biogen and Pfizer data, respectively, while G.M.R. managed code containerization and the development of the online PlayMolecule.org application. J.J. wrote the manuscript with help from all the other authors.

Competing interests

G.D.F is a founder and current CEO of Acellera Ltd. J.J. receives funding from Acellera Ltd. and G.M.R. is an employee.

Additional information

Supplementary information is available for this paper at: https://doi.com/XXXXXX/XXXXXX
Correspondence and requests for materials should be addressed to G.D.F.

References

[1] Christos A Nicolaou and Nathan Brown. *Drug Discovery Today: Technologies*, 10(3):e427–e435, 2013.

[2] I Kola and J Landis. *Nature Reviews Drug Discovery*, 3.

[3] Sean Ekins, J Dana Honeycutt, and James T Metz. *Drug discovery today*, 15(11-12):451–460, 2010.

[4] L. Wang, B. J. Berne, and R. A. Friesner. *Proceedings of the National Academy of Sciences*, 109(6):1937–1942, 2012.

[5] Eelke B. Lenselink, Julien Louvel, Anna F. Forti, et al. *ACS Omega*, 1(2):293–304, 2016.

[6] Shunzhou Wan, Agastya P. Bhati, Sarah Skerratt, et al. *Journal of Chemical Information and Modeling*, 57(4):897–909, 2017.

[7] Dahlia A. Goldfeld, Robert Murphy, Byungchan Kim, et al. *Journal of Physical Chemistry B*, 119(3):824–835, 2015.

[8] Laura Pérez-Benito, Henrik Keränen, Herman van Vlijmen, and Gary Tresadern. *Scientific Reports*, 8(1):4883, 2018.

[9] Myriam Ciordia, Laura Pérez-Benito, Francisca Delgado, Andrés A Trabanco, and Gary Tresadern. *Journal of Chemical Information and Modeling*, 56(9):1856–1871, 2016.

[10] Christina Schindler, Friedrich Rippmann, and Daniel Kuhn. *Journal of Computer-Aided Molecular Design*, 32(1):1–8, 2017.

[11] Henrik Keränen, Laura Pérez-Benito, Myriam Ciordia, et al. *Journal of Chemical Theory and Computation*, 13(3):1439–1453, 2017.

[12] Germano Heinzelmann, Niel M. Henriksen, and Michael K. Gilson. *Journal of Chemical Theory and Computation*, 13(7):3260–3275, 2017.

[13] Matteo Aldeghi, Alexander Heifetz, Michael J. Bodkin, Stefan Knapp, and Philip C. Biggin. *Chemical Science*, 7(1):207–218, 2016.

[14] Zoe Cournia, Bryce Allen, and Woody Sherman. *Journal of Chemical Information and Modeling*, 57(12):2911–2937, 2017.

[15] Yang Cao and Lei Li. *Bioinformatics*, 30(12):1674–1680, 2014.

[16] Michael P Brenner, Lucy J Colwell, et al. *Proceedings of the National Academy of Sciences*, 113(48):13564–13569, 2016.

[17] Thomas A. Halgren, Robert B. Murphy, Richard A. Friesner, et al. *Journal of Medicinal Chemistry*, 47(7):1750–1759, 2004.

[18] Oleg Trott and Aj Olson. *Journal of Computational Chemistry*, 31(2):455–461, 2010.

[19] Pedro J. Ballester and John B. O. Mitchell. *Bioinformatics*, 26(9):1169–1175, 2010.

[20] José Jiménez, Miha Škalič, Gerard Martínez-Rosell, and Gianni De Fabritiis. *Journal of Chemical Information and Modeling*, 58(2):287–296, 2018.

[21] Evan N Feinberg, Debnil Sur, Zhenqin Wu, et al. *ACS central science*, 4(11):1520–1530, 2018.

[22] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. *Journal of Chemical Information and Modeling*, 57(4):942–957, 2017.

[23] Duc Duy Nguyen, Zixuan Cang, Kedi Wu, et al. *Journal of computer-aided molecular design*, 33(1):71–82, 2019.

[24] Zied Gaieb, Conor D Parks, Michael Chiu, et al. *Journal of computer-aided molecular design*, 33(1):1–18, 2019.

[25] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. *Journal of Medicinal Chemistry*, 47(12):2977–2980, 2004.

[26] Wenhu Zhan, Daqiang Li, Jinxin Che, et al. *European journal of medicinal chemistry*, 75:11–20, 2014.

[27] Ata Amini, Paul J Shrimpton, Stephen H Muggleton, and Michael JE Sternberg. *Proteins: Structure, Function, and Bioinformatics*, 69(4):823–831, 2007.

[28] David Zilian and Christoph A Sotriffer. *Journal of chemical information and modeling*, 53(8):1923–1933, 2013.

[29] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N. Jorissen, and Michael K. Gilson. *Nucleic Acids Research*, 35(SUPPL. 1), 2007.

[30] Lingle Wang, Yujie Wu, Yuqing Deng, et al. *Journal of the American Chemical Society*, 137(7):2695–2703, 2015.

[31] David L. Mobley and Michael K. Gilson. *Annual Review of Biophysics*, 46(1):531–558, 2017.

[32] Frederik JR Rombouts, Gary Tresadern, Peter Buijnsters, et al. *ACS Medicinal Chemistry Letters*, 6(3):282–286, 2015.

[33] Peter Buijnsters, Meri De Angelis, Xavier Langlois, et al. *ACS Medicinal Chemistry Letters*, 5(9):1049–1053, 2014.

[34] Frederik J. R. Rombouts, Gary Tresadern, Oscar Delgado, et al. *Journal of Medicinal Chemistry*, 58(20):8216–8235, 2015. PMID: 26378740.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. *Advances In Neural Information Processing Systems*, pages 1–9, 2012.

[36] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, pages 3156–3164, 2015.

[37] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR*, pages 1–16, 2016.

[38] Miha Skalic, José Jiménez Luna, Davide Sabbadin, and Gianni De Fabritiis. *Journal of chemical information and modeling*, 2019.

[39] Bharath Ramsundar, Bowen Liu, Zhenqin Wu, et al. *Journal of Chemical Information and Modeling*, 57(8):2068–2076, 2017.

[40] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, et al. *Chemical Science*, 9(2):513–530, 2018.

[41] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S. Rose, and Gianni De Fabritiis. *Bioinformatics*, 33(19):3036–3042, 2017.

[42] Miha Skalic, Alejandro Varela-Rial, José Jiménez, Gerard Martínez-Rosell, and Gianni De Fabritiis. *Bioinformatics*, 35(2):243–250, 2018.

[43] Miha Skalic, Gerard Martínez-Rosell, José Jiménez, and Gianni De Fabritiis. *Bioinformatics*, 2018.

[44] Christoph Wehmeyer and Frank Noé. *The Journal of chemical physics*, 148(24):241703, 2018.

[45] Georgy Derevyanko, Sergei Grudinin, Yoshua Bengio, and Guillaume Lamoureux. *Bioinformatics*, 34(23):4046–4053, 2018.

[46] Marwin HS Segler, Mike Preuss, and Mark P Waller. *arXiv preprint arXiv:1708.04202*, 2017.

[47] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, et al. *ACS central science*, 4(2):268–276, 2018.

[48] Christian Kramer and Peter Gedeck. *Journal of chemical information and modeling*, 50(11):1961–1969, 2010.

[49] Robert P. Sheridan. *Journal of Chemical Information and Modeling*, 53(4):783–790, 2013.

[50] Richard A. Friesner, Jay L. Banks, Robert B. Murphy, et al. *Journal of Medicinal Chemistry*, 47(7):1739–1749, 2004.

[51] Samuel Genheden and Ulf Ryde. *Expert Opinion on Drug Discovery*, 10(5):449–461, 2015.

[52] Tingjun Hou, Junmei Wang, Youyong Li, and Wei Wang. *Journal of chemical information and modeling*, 51(1):69–82, 2010.

[53] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. *arXiv preprint arXiv:0912.3995*, 2009.

[54] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. *The Journal of Chemical Physics*, 148(24):241722, 2018.

[55] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. *Chemical science*, 8(4):3192–3203, 2017.

[56] Russell Spitzer and Ajay N. Jain. *Journal of Computer-Aided Molecular Design*, 26(6):687–699, 2012.

[57] C Camacho, G Coulouris, V Avagyan, et al. *BMC Bioinformatics*, 10(421):1, 2009.

[58] Greg Landrum. *Online. http://www.rdkit.org.*, 2006.

[59] Christian Kramer and Peter Gedeck. *Journal of Chemical Information and Modeling*, 50(11):1961–1969, 2010.

[60] George Cybenko. *Approximation Theory and its Applications*, 9(3):17–28, 1989.

[61] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *MIT Press*, 2016.

[62] Diederik P. Kingma and Jimmy Lei Ba. *International Conference on Learning Representations 2015*, pages 1–15, 2015.

[63] Adam Paszke, Gregory Chanan, Zeming Lin, et al. *Advances in Neural Information Processing Systems 30*, (Nips):1–4, 2017.

[64] S. Doerr, M. J. Harvey, Frank Noé, and G. De Fabritiis. *Journal of Chemical Theory and Computation*, 12(4):1845–1852, 2016.