State of the art iterative docking with logistic regression and Morgan fingerprints

Lewis J. Martin

The University of Sydney, Brain and Mind Centre, The Lambert Initiative for Cannabinoid Therapeutics, Sydney, NSW, Australia

Abstract:

There is renewed interest in docking campaigns for ligand-discovery since the advent of ultra-large scale virtual libraries. Using brute-force search, the scale of the libraries suggests highly parallelized compute should be used to avoid years-long computations. This paper reports a re-analysis of docking data from an ultra-large docking campaign at the D4 receptor and AmpC beta lactamase, and demonstrates large reductions in computation time to identify the top-ranked ligands. A search of 'baseline' featurizations shows that logistic regression on Morgan fingerprints with pharmacophoric atom invariants can match the reported performance on the same task using message-passing networks. With this approach, an ultra-large docking campaign could be performed in a matter of weeks using consumer-grade CPUs with *RDKit* and *scikit-learn*. All code and figures are available at https://github.com/ljmartin/dockop

1. Introduction:

Molecular docking, and the related technique of molecular shape-matching, are virtual screening techniques that use the charge and 3D shape of molecules to identify new ligands for a target protein (1). Shape matching requires only a known ligand as a starting point and scores query molecules based on their overlapping volume and charge similarity. Docking requires a crystal structure of a protein, and scores molecules using some model of interaction with the binding site. Compared to shape-matching, docking is less biased to the structure of existing ligands, and can potentially identify interactions outside the known ligand binding mode (2). On the other hand, shape-matching is relatively faster (3) and showed good performance in early comparisons (4). The advent of "ultra-large" scale molecule libraries, defined loosely as ~10^8 ligands, and some high-profile successes have driven renewed interest in docking for novel scaffold discovery (5-7).

Practically, a docking campaign might begin by identifying a virtual library of synthesizable molecules. The larger and more diverse the library is, the more likely it contains novel active ligands for the target under investigation. Published ultra-large scale campaigns have taken a brute-force approach, docking ~10^8 - 10^9 ligands to identify the highest-ranking molecules for *in vitro* validation.

Since the chance of success increases with library size, and docking is typically slower than shape-matching, the time required to dock a full ultra-large scale library is a barrier to progress. As an example, the ZINC20 browser lists nearly 700 million protomers for the UCSF DOCK software (8, 9). At 1s per ligand, as estimated by Lyu et al. (6) , docking calculations would require approximately 20 years of compute. This is a lower bound – at the 15s estimated by Gorgulla et al. (7), this requires >300 years of compute. Clearly, then, docking campaigns require high-performance computing to parallelize, slowing iteration time and reducing the pool of researchers able to participate. Faster docking campaigns will broaden participation in virtual screening and novel ligand discovery.

One way to improve speed, without requiring additional resources, is to avoid brute-force search by use of a statistical model that acts as a surrogate for the docking algorithm. Known as 'iterative screening' in virtual screening research (10), or 'active learning' in machine learning research (11), surrogate models can steer the search toward molecules that are more likely to be high-scoring. Substantial reductions in total docking time have been achieved using either classical machine learning (12) or artificial neural networks (13, 14).

One recent example of active learning for docking is Graff et al, who used simulated analyses to show over an order of magnitude reduction in the number of docking calculations using a dataset of ligands docked to AmpC beta lactamase (13). This work is notable not just due to the reduction in docking time, but because they used state of the art technology as the surrogate model – namely, a message-passing network that applies deep learning procedures directly to molecular graphs without featurization, which improved performance compared to classical machine learning. In addition, they were the first to apply this to an ultra-large scale dataset, which was made available by Lyu et al. (6), and which may be one of the drivers for renewed interest in molecular docking.

This manuscript reports the performance of iterative screening using logistic regression on both the AmpC beta lactamase and D4 receptor data from Lyu et al.(6). The model relating docking score to hit-rate is re-framed to use percentiles, rather than absolute docking scores, making it amenable to classification. This framing is used to perform optimization of 'baseline' featurizations such as the Morgan fingerprint, showing larger fingerprint sizes and pharmacophoric atom invariants achieve substantial gains over the *de facto* standard of the 2,048 bit Morgan fingerprint. Comparing directly with the results in Graff et al., this simple approach can achieve better performance than a message-passing network. The AmpC and D4 datasets are also compared, showing performance is target-

dependent. These analyses show that ultra-large docking campaigns can be performed in 1-2 weeks using consumer-grade CPUs, *scikit-learn* and *rdkit*. All analyses are reproducible, and code is available at https://github.com/ljmartin/dockop

2. Results:

2.1 D4: Binarizing docking scores by percentile

Classifiers require a threshold, a.k.a. cut-off, score to binarize the ligands into 'high-scoring' and 'low-scoring' classes. The experimental validation data from Lyu et al. can be used to estimate a threshold score (6). These validation data consist of *in vitro* assays of D4 receptor inhibition by 549 compounds over a wide range of docking scores. They indicate hit-rate varies approximately monotonically with score. In their model, hit-rate was binned over docking scores. Binning can lead to artefacts that depend on the bin size (15). Similarly, the distribution of ligands over the range of docking scores is irregular, and may distort the relationship with hit rate. In addition, the range of scores output by UCSF DOCK can vary depending on the shape of the binding site and the chemistry of the exposed residues, which would require a new model is fit for each new protein.

Figure 1 shows a generalized additive model fit to the validation data with a logit link function, without binning, and using ligand ranks instead of raw docking scores. The ligand ranks were calculated by sorting all 116 million docking scores and, when normalized to the range (0,100), are equivalent to a percentile that is uniformly distributed. This is in line with the intended use-case for docking in ligand discovery as a ranking tool, rather than estimating the distribution of binding affinities directly.

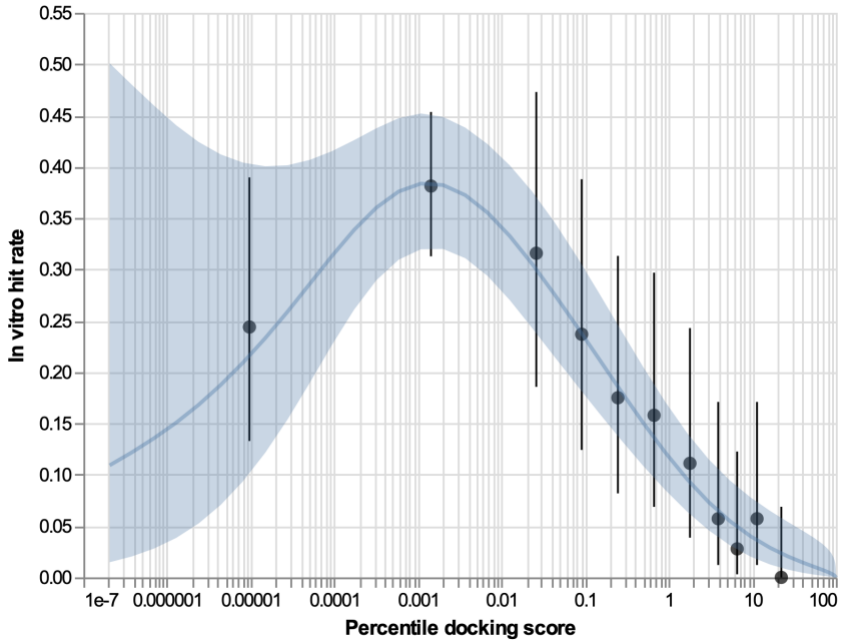

Figure 1. Estimate of *in vitro* hit-rate with percentile docking score for the D4 receptor using data from Lyu et al. (6). A generalized additive model was fit to the inhibition data (blue line, with 95% CIs in shaded area). The hit rates and 95% CIs for the binned data is also shown for visualization purposes (black, scatter points) but only the un-binned data was used in model fitting. This model can be used to determine a cut-off that binarizes docking scores into high- and low-scoring classes. This cut-off was set to the 0.3rd percentile for all experiments.

The use of a generalized additive model grants the flexibility to model the reduction in hit-rate observed at the ~0.0001th percentile and below, which corresponds to the best-scoring 11,600 ligands in the D4 receptor screen. It also helps define a cut-off for binarizing docking scores into 'high-scoring' and 'low-scoring' classes. For the remainder of this work, all logistic regression classifiers have been trained on binarized docking scores with cut-off set to the 0.3rd percentile score in the training data. This level has a half-maximal *in vitro* hit rate, in an attempt to maximize data in the high-scoring class while excluding low-scoring ligands.

2.2 AmpC: Binary fingerprints have a wide range of performance

Multiple fingerprints and fingerprint sizes were analysed to evaluated their performance as surrogate models. Simple, accessible, and fast approaches that handle sparse representation were favoured. To that end, **Figure 2** shows a comparison of fingerprints available in the *RDKit* library (Atom pair (16), Morgan(17), Morgan with pharmacophoric atom invariants (18), Pattern (19), RDK (19), and Topological torsion (20)). Classifiers were trained on the binarized docking scores, using 0.3% as

the percentile cut-off, and compared with average precision, a ranking metric, over 5 Monte Carlo cross-validation repeats using the AmpC dataset (see methods for details).

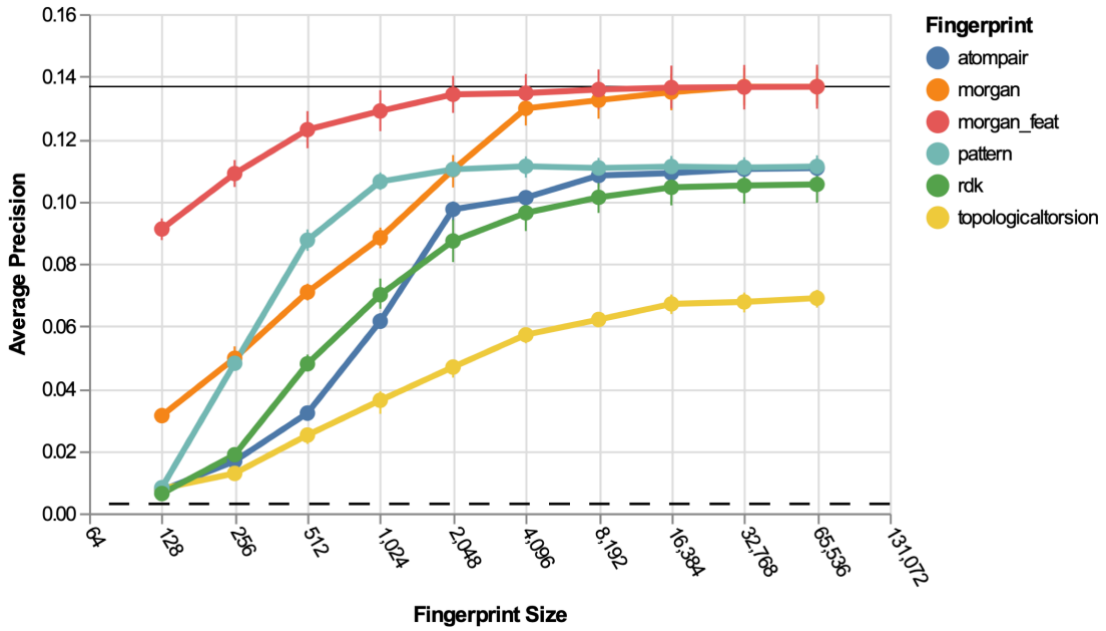

Figure 2. Performance of logistic regression from *scikit-learn* on 5-fold Monte-Carlo cross-validated test sets using the AmpC dataset and molecular fingerprints available in *RDKit*. In general, larger fingerprints perform better, indicating bit-collisions hurt performance. The best-performing fingerprint is the Morgan fingerprint, and the use of pharmacophoric atom-invariants improves performance at lower dimensionality (compare *morgan* and *morgan_feat*). The dotted line indicates performance expected from a random classifier, and the unbroken line indicates best mean performance.

In general, larger fingerprint sizes perform better with performance improving beyond 2,048 bits. Based on these results, whichever fingerprint is used for machine learning of protein-ligand binding it is suggested the size be at least 8,192 rather than the *de facto* standard of 2,048. Using sparse representation, the increase in required storage space for size 8,192 fingerprints is less than 1% over 2,048 fingerprints. Likewise, for logistic regression the additional training time at larger sizes is negligible. The choice of larger fingerprint size thus has substantial benefit but inconsequential downside.

The best-performing featurization was the Morgan fingerprint. Interestingly, including pharmacophoric featurization improved performance at smaller sizes. Since this effect disappears

at higher sizes, it may be a result of bit-collisions affecting the performance of atomic-identity invariants, while pharmacophoric invariants are less affected.

Based on these results, going forward all experiments used Morgan fingerprints with pharmacophoric invariants of size 8,192. Several classifiers and regressors in *sklearn* were investigated (data not shown), but they performed worse than logistic regression with default parameters (inverse shrinkage parameter *C=1*).

2.3 AmpC: Proper sampling requires at least 400,000 ligands

Iterative docking relies on an initial random sample as a training set. Ideally, the training set completely samples the chemical diversity in the high-scoring class such that it resembles the test set. In addition, for classifiers, the training set should have a good estimate of the cut-off percentile. In practice these only occur with large sample sizes - the top 0.3% ligands are in the minority class by definition, making chemical diversity difficult to sample, and the skew of docking scores in this region makes it difficult to estimate the percentile cut-off. Testing the relationship between predictive performance and training set size might gauge the range of best performance and determine the best size for the initial sample in practice.

Figure 3, upper panel, shows that average precision increases with increasing training sample size up to at least 1,500,000 randomly-sampled ligands, although most of the gains were achieved by ~400,000 ligands. This provides a reasonable starting point for the random sample size in iterative docking. At 400,000 ligands, there will be 1,200 ligands in the 'high-scoring' class based on the 0.3rd percentile cut-off, suggesting at least 1,000 ligands are required to widely sample the chemical diversity of the minority class. **Figure 3, lower panel**, shows the estimated cut-off values calculated from the training sets of different size. While lower training set sizes have widely varying estimates, from 400,000 ligands and above the estimated 0.3rd percentile closely tracks the true value.

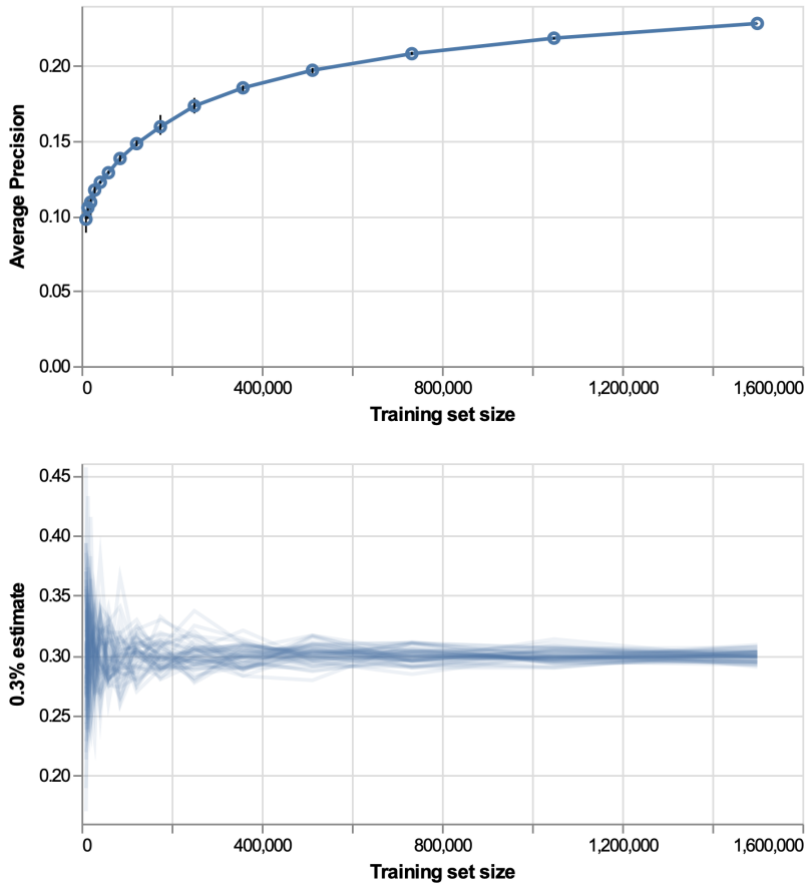

Figure 3. Performance improves with increasing training set size. Since the 'high-scoring' and 'low-scoring' classes are, by definition, imbalanced, complete sampling of chemical diversity requires at least ~400,000 training set sizes, at which point the chemical diversity and estimated cut-off are approximately the same between the training and testing sets.

2.3 AmpC and D4: Single-iteration performance

While larger training sets lead to better classifiers, in practice the training set also takes time to dock. The enrichment (i.e. fold reduction in time compared to random search), and the total time spent docking, are thus a function of model performance *and* training set size. Calculating enrichment and total docking time for a range of training set sizes and goal numbers of high-scoring hits can estimate performance when using just a single iteration of iterative docking. In this case, "high-scoring" is all ligands in the top 0.3rd percentile.

Figure 4 shows the enrichment for the full AmpC (left panel) and D4 (right panel) datasets. In general, enrichment is much better for the AmpC dataset, indicating performance is target-dependent. For both datasets, there is a maximum in the enrichment for any number of desired hits.

After this point, improving the classifier leads to overall worse performance, since the trade-off between time spent docking the training set vs. time saved not docking the test set diminishes. In addition, for increasing number of desired hits, the achievable enrichment gets smaller. For even the largest number of desired high-scorers identified, though, single-iteration docking improves over the brute-force approach by at least an order of magnitude.

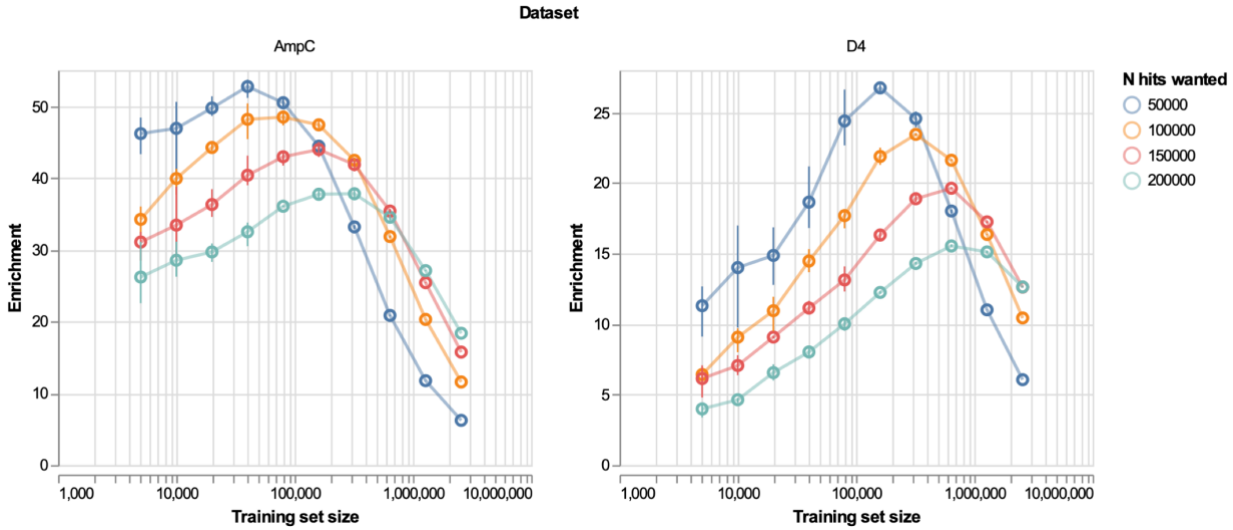

Figure 4 Hit discovery enrichment compared to brute-force search, using a single iteration of iterative docking for the AmpC and D4 datasets. A hit is defined as a ligand in the top 0.3rd percentile of docking scores. In general, the D4 dataset performs worse than the AmpC dataset, indicating performance is target-dependent.

Enrichment is a useful metric in algorithm comparison. In practice, the main concern in a docking campaign is the total computation time (apart from *in vitro* hit-rate). **Figure 5** shows the single-iteration performance in terms of the total number of days spent docking both the training and testing sets, assuming only a single-core CPU is available and docking 1s/ligand as in Lyu et al. (6). As with enrichment, single-iteration docking performs better for AmpC than D4. These results indicate that, for 100,000 docking hits, a docking campaign could be completed in approximately 8 days for AmpC or 17 days for D4 using just a single CPU. Since most modern workstations come with multiple CPUs, and docking is trivially parallelized, this number is an upper range.

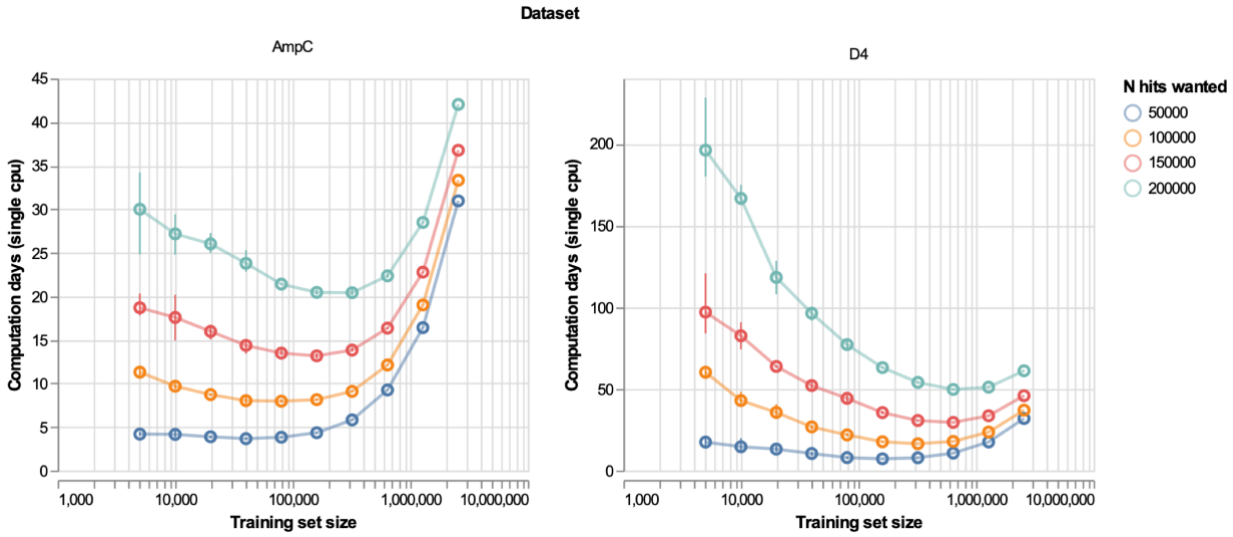

Figure 5 Estimated docking time to discover 'N' hits, where hit is defined as being in the top 0.3th percentile of docking scores. Using brute force search, docking times would be 192 days (50,000 hits), 385 days (100,000 hits), 579 days (150,000 hits), 772 days (200,000 hits).

2.4 Comparison to previous work

An alternative to the single iteration is active learning. Graff et al. demonstrated large performance gains on the AmpC lactamase data with this approach using a random forest, neural network, or a message-passing graph neural network to regress molecule structure against docking score (13). In their approach, a single random sample is taken as an initial training batch, followed by five iterations of batch selection guided by the model (i.e. greedy acquisition) and retraining on the growing batches. The advantage of this approach is that the class imbalance becomes less as the model adds more "hits" to the training set. Replicating this approach exactly can demonstrate a comparison of artificial neural networks and a logistic regression classifier.

To facilitate this comparison, **Figure 6** shows the same metric used in Graff et al., which is the percentage top-k docking scores identified, with k set to 0.05% out of the approximately 100 million ligands in the AmpC lactamase dataset (13). For convenience, k is set to exactly 50,000, as was reported in Graff et al. (13), but to 58,121 for the D4 receptor dataset which has more total ligands. Also shown are the %-top-k ligands reported in Graff et al. (13). Our approach, a combination of logistic regression with Morgan fingerprints and pharmacophoric atom invariants, achieves better performance than the message-passing network when using a greedy acquisition strategy. Similarly to the single-iteration data, performance on the AmpC dataset is better than on the D4 dataset but, since the D4 dataset was not used in Graff et al, we cannot show a direct comparison.

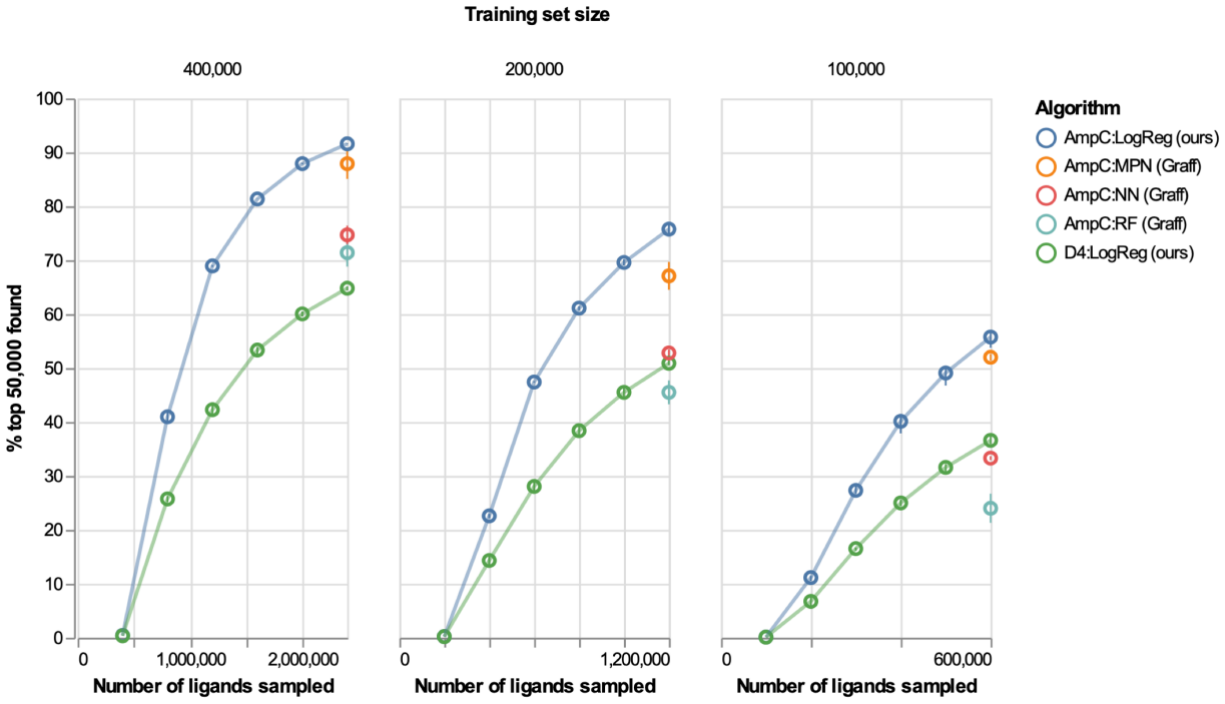

Figure 6 Percentage of the top-k docking scores from the AmpC (k=50,000) or D4 (k=58,121) datasets identified using iterative docking. These data replicate an experiment from Graff et al. (13), whose data are reproduced here, but using logistic regression and Morgan fingerprints instead of graph neural networks. In general, performance on the AmpC dataset is better than on D4. Overall the performance of logistic regression improves on the message-passing network (MPN).

These results are also presented in terms of computation time in **Figure 7**. Based on these calculations, an active learning docking campaign on AmpC would identify over 90% of the top-50,000 ligands in under 28 computation days using just a single-core CPU. Like the single-iteration approach, the docking is trivially parallelized and this is an upper range. In addition, since the LBFGS solver using a single CPU fits a *scikit-learn* logistic regression classifier in ~10s for 400,000 ligands, the contribution from model training is negligible.

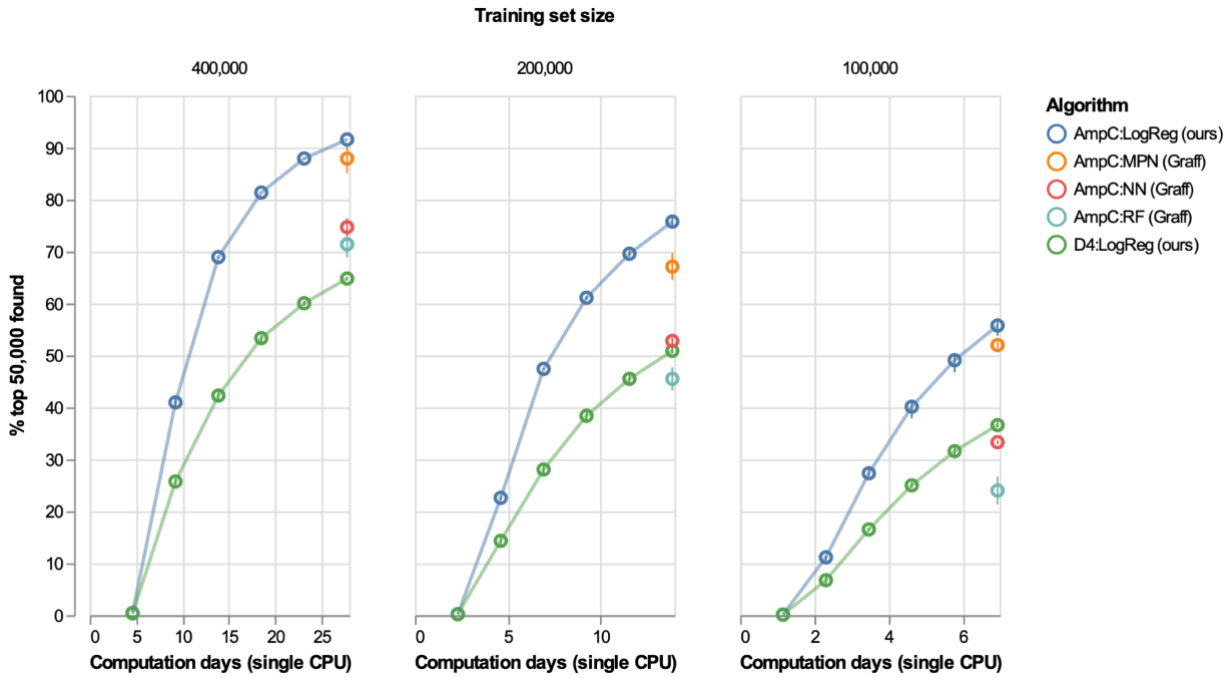

Figure 7 Time spent to identify the top-50,000 ligands from the AmpC and D4 datasets, assuming a single-core CPU and 1/s per ligand. These data are the same as in **Figure 6**, but the x-axis is transformed into docking time.

3. Discussion:

Large-scale docking campaigns have consistently demonstrated the ability to identify *in vitro* hits with novel scaffolds (21-26). Nevertheless, the computation demands for ultra-large campaigns are substantial, typically being associated with high-performance computing clusters(5-7). An alternative to brute-force search is iterative docking, whereby a surrogate model is trained on a random sample to steer the search towards high-scoring ligands. Re-analysing the dataset of Lyu et al.(6) allowed an assessment of how quickly and simply iterative docking could be performed. We compared several featurizations available in the *RDKit* combined with logistic regression from *scikit-learn*. Using the Morgan fingerprint with pharmacophoric features and logistic regression shows an at-least 10-fold enrichment over brute-force search for the most difficult dataset with a single iteration. For active learning, consisting of multiple iterations, this compares favourably against the results previously achieved using a message-passing neural network.

The best-performing featurization overall was the Morgan fingerprints with pharmacophoric atom invariants. Previous work showed that fingerprints motivated by bioactivity outperform the Morgan fingerprint in a ligand-based virtual screening task after unbiasing (27). The default atom invariants in *RDKit,* used to calculate graph substructures, differentiate atoms by proton number. Bioisosteric

atom replacement, such as hydrogen-donor oxygen to hydrogen-donor nitrogen, would thus map to different features despite making the same interactions in the binding site. Pharmacophoric invariants may show better performance by avoiding this feature stratification at lower fingerprint sizes. It may also be useful to test other bioactivity-based fingerprints like CATS(28).

The improved performance at high-dimensionality was a surprising result. Since the fingerprinting algorithm does not change across dimensionality this must arise due to reducing the number of bit-collisions, which become substantial below ~4,096 bits. Accordingly, the best performance was found above this size. This result suggests virtual screening researchers should verify that 2,048-bit fingerprints is an appropriate benchmark, since benchmark performance might be improved by increasing the dimensionality at negligible cost in storage and training time.

Logistic regression requires choosing a cut-off parameter to separate classes, which may seem like a disadvantage compared to non-linear regressors like neural networks or random forests. On the other hand, the USCF DOCK software does not produce docking scores in the same range for every binding site, and it is not clear a regressor will perform equally well across different proteins when the score distribution changes. Indeed, Graff et al. showed how regressors can perform worse when docking scores have lower standard deviation, although it's unclear if different docking algorithms confounded that result (13). Logistic regression, however, applies to the top-ranked scores from any score distribution.

We also re-analysed the *in vitro* data from Lyu et al. Transforming the docking scores to normalized ranks is a natural choice given the expected use-case for docking, in which the magnitude of the raw scores are primarily used to rank molecules. Another advantage of this approach is that a cut-off can be chosen once and used to threshold the docking scores on any desired protein target, whereas the distribution of raw scores will change in each case. Using the 0.3rd percentile as the cut-off yields good performance on these data. Nevertheless, the performance of iterative docking is target-dependent, as shown on the AmpC and D4 datasets, so other targets may perform worse.

These results come at a time when graph-based neural networks are increasingly popular for computational chemistry, although their benefits have also come into question (29). The performance of the *de facto* standard 2,048 bit Morgan fingerprint can be improved simply by using pharmacophoric atom invariants and/or larger fingerprint sizes. For iterative docking, this improvement is more than enough to match the performance of a message-passing network. While not prohibitive, the training time and implementation cost of graph neural networks make it

worthwhile to use faster and simpler alternatives. With logistic regression, the aggregate training and inference time for the surrogate model in a docking campaign would be less than approximately 1 minute.

No statistical tests were used to compare the performance of logistic regression with message-passing networks because the difference in docking computation time is small enough to be functionally equivalent in practical use-cases. The main advance of our work is the simplicity and speed of the implementation while maintaining state-of-the-art performance. As virtual libraries approach 10^9 - 10^{10} ligands, ease of implementation will be a crucial differentiator. These results also suggest that the signal in bioactivity data might not require advanced algorithms to identify, meaning other virtual screening results might be improved simply by a fingerprint search.

Finally, some speculation on why surrogate models work so well to model docking scores. The use of machine learning to model *in vitro* bioactivity has been criticised for poor performance on out-of-distribution data, returning predictions equivalent to a 1-NN classifier (30, 31). Typical pipelines in bioactivity modelling use data from medicinal-chemistry articles via e.g. ChEMBL (32). These data are not randomly sampled, with redundant instances that heavily weight the learning algorithms towards existing scaffolds. In addition, the data may represent multiple binding sites, confusing the labelling. Iterative docking, on the other hand, uses true random sampling from the available library in the first step, and is performed at a single binding site. These two properties mean iterative docking uses training sets with the same distribution in chemical space as the test set, which may account for the superior performance. It will be interesting to determine whether such models can predict bioactivity directly.

Ultra-large scale virtual libraries are a recent advance that have renewed interest in molecular docking for discovery of novel chemotypes. This accessible chemical space is enticing, but the practicalities of docking 10^7 - 10^9 molecules requires highly parallelized computer clusters. Iterative docking, a.k.a. active learning, uses a surrogate model to reduce computation by prioritizing the library based on one or more training samples of docked ligands. This work approached iterative docking with the goal of minimizing implementation cost by using fast featurization and learning algorithms. The approach, which uses logistic regression and Morgan fingerprints, is at least on par with state of the art graph neural networks. This approach can identify over 90% of the top-50,000 ligands in a 10^8-scale virtual library using only a consumer-grade CPU in 1-2 weeks.

4. Methods

All code is available at https://github.com/ljmartin/dockop

4.1 Data

SMILES codes and docking scores were from Lyu et al.(6). Results comparison with a message-passing network used data in tables S6, S7, and S8 of reference (13)

4.1. Software

Molecules were featurized into molecular fingerprints using *RDKit* (19). The model relating hit-rate and docking-rank used *PyGAM* (33). Machine learning algorithms were implemented using *scikit-learn* (34). Fingerprints were stored and processed as sparse arrays, using *numpy* (35) and *scipy* data structures (36). All visualization was performed with *altair (37),* a python API for *vega-lite(38)*.

4.2 Analysis

The model relating hit-rate and docking ranks used a generalized additive model. While this approach may be less interpretable than the sigmoid curve used in Lyu et al.(6), it allowed us to successfully model the reduced hit-rate observed at higher (better) ranks. To resolve the changes occurring between the 0^{th} and 1^{st} percentiles, the normalized ranks were first logit-transformed, and the model fit to the transformed percentiles. The GAM used a binomial distribution to model hits with a logit-link to squash output into the range (0,1) and *n_splines=8*.

The surrogate model, logistic regression, was implemented using *scikit-learn*. The fingerprint comparison was performed on a subset of size 1,000,000 ligands from the AmpC dataset of Lyu et al. (6). Five repeats of Monte-Carlo cross-validation were performed, in which successive test-sets are chosen at random with replacement (39). In each round of cross-validation, 50,000 ligands were selected as training data, with the remaining 950,000 ligands as test data. Evaluation used the average precision metric, which approximates the area under the precision-recall curve (40).

The single-iteration and iterative-docking analyses used the full AmpC and D4 datasets. Error bars are the 95% confidence interval of three repeats, calculated by *altair*.

Supplementary Data

Supplementary Table 1: Raw data for percentage of the top-50,000 docking scores from the AmpC or D4 datasets identified using iterative docking

Training set size	Number of ligands sampled	% top-50,000 ligands identified (mean, %)	95% confidence interval
100,000	100,000	0.1	0.0 – 0.2
	200,000	11.1	9.7 – 12.5
	300,000	27.3	25.4 – 29.2
	400,000	40.1	35.3 – 44.9
	500,000	49.1	44.1 – 54.1
	600,000	55.8	51.4 – 60.1
200,000	200,000	0.2	0.1 – 0.3
	400,000	22.6	21.6 – 23.6
	600,000	47.4	47.1 – 47.7
	800,00	61.1	60.1 – 62.1
	1,000,000	69.6	68.1 – 71.1
	1,200,000	75.8	73.6 – 77.9
400,000	400,000	0.4	0.3 – 0.5
	800,000	41.0	38.1 – 43.8
	1,200,00	68.9	68.6 – 69.3
	1,600,00	81.4	81.3 – 81.4
	2,000,000	87.9	87.7 – 88.2
	2,400,000	91.6	91.4 – 91.8

Acknowledgements

This work was supported by the Lambert Initiative for Cannabinoid Therapeutics, a philanthropic research program based at the Brain and Mind Centre, The University of Sydney.

References:

1. Kuntz ID. Structure-based strategies for drug design and discovery. Science. 1992;257(5073):1078-82.
2. Irwin JJ, Shoichet BK. Docking screens for novel ligands conferring new biology: Miniperspective. J Med Chem. 2016;59(9):4103-20.
3. Grant JA, Gallardo M, Pickup BT. A fast method of molecular shape comparison: A simple application of a Gaussian description of molecular shape. Journal of computational chemistry. 1996;17(14):1653-66.
4. Hawkins PC, Skillman AG, Nicholls A. Comparison of shape-matching and docking as virtual screening tools. J Med Chem. 2007;50(1):74-82.
5. Stein RM, Kang HJ, McCorvy JD, Glatfelter GC, Jones AJ, Che T, et al. Virtual discovery of melatonin receptor ligands to modulate circadian rhythms. Nature. 2020;579(7800):609-14.

6. Lyu J, Wang S, Balius TE, Singh I, Levit A, Moroz YS, et al. Ultra-large library docking for discovering new chemotypes. Nature. 2019;566(7743):224-9.
7. Gorgulla C, Boeszoermenyi A, Wang Z-F, Fischer PD, Coote PW, Das KMP, et al. An open-source drug discovery platform enables ultra-large virtual screens. Nature. 2020;580(7805):663-8.
8. Coleman RG, Carchia M, Sterling T, Irwin JJ, Shoichet BK. Ligand pose and orientational sampling in molecular docking. PLoS One. 2013;8(10):e75992.
9. Irwin JJ, Tang KG, Young J, Dandarchuluun C, Wong BR, Khurelbaatar M, et al. ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. J Chem Inf Model. 2020.
10. Willems H, De Cesco S, Svensson F. Computational Chemistry on a Budget: Supporting Drug Discovery with Limited Resources: Miniperspective. J Med Chem. 2020;63(18):10158-69.
11. Settles B. Active learning. Synthesis lectures on artificial intelligence and machine learning. 2012;6(1):1-114.
12. Svensson F, Norinder U, Bender A. Improving screening efficiency through iterative screening using docking and conformal prediction. J Chem Inf Model. 2017;57(3):439-44.
13. Graff DE, Shakhnovich EI, Coley CW. Accelerating high-throughput virtual screening through molecular pool-based active learning. arXiv preprint arXiv:201207127. 2020.
14. Gentile F, Agrawal V, Hsing M, Ton A-T, Ban F, Norinder U, et al. Deep docking: a deep learning platform for augmentation of structure based drug discovery. ACS central science. 2020;6(6):939-49.
15. Engel J. The multiresolution histogram. Metrika. 1997;46(1):41-57.
16. Carhart RE, Smith DH, Venkataraghavan R. Atom pairs as molecular features in structure-activity studies: definition and applications. J Chem Inf Comput Sci. 1985;25(2):64-73.
17. Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model. 2010;50(5):742-54.
18. Gobbi A, Poppinger D. Genetic optimization of combinatorial libraries. Biotechnology and bioengineering. 1998;61(1):47-54.
19. Landrum G. RDKit: Open-source cheminformatics. 2006.
20. Nilakantan R, Bauman N, Dixon JS, Venkataraghavan R. Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. J Chem Inf Comput Sci. 1987;27(2):82-5.
21. Wang S, Wacker D, Levit A, Che T, Betz RM, McCorvy JD, et al. D4 dopamine receptor high-resolution structures enable the discovery of selective agonists. Science. 2017;358(6361):381-6.
22. Patel N, Huang XP, Grandner JM, Johansson LC, Stauch B, McCorvy JD, et al. Structure-based discovery of potent and selective melatonin receptor agonists. Elife. 2020;9:e53779.
23. Cheng Q, Shah N, Bröer A, Fairweather S, Jiang Y, Schmoll D, et al. Identification of novel inhibitors of the amino acid transporter B0AT1 (SLC6A19), a potential target to induce protein restriction and to treat type 2 diabetes. British journal of pharmacology. 2017;174(6):468-82.
24. Zhao H, Dong J, Lafleur K, Nevado C, Caflisch A. Discovery of a novel chemotype of tyrosine kinase inhibitors by fragment-based docking and molecular dynamics. ACS Med Chem Lett. 2012;3(10):834-8.
25. Berger WT, Ralph BP, Kaczocha M, Sun J, Balius TE, Rizzo RC, et al. Targeting fatty acid binding protein (FABP) anandamide transporters–a novel strategy for development of anti-inflammatory and anti-nociceptive drugs. PLoS One. 2012;7(12):e50968.
26. Lacroix C, Fish I, Torosyan H, Parathaman P, Irwin JJ, Shoichet BK, et al. Identification of novel smoothened ligands using structure-based docking. PLoS One. 2016;11(8):e0160365.
27. Martin LJ, Bowen MT. Comparing fingerprints for ligand-based virtual screening: a fast, scalable approach for unbiased evaluation. J Chem Inf Model. 2020.

28. Reutlinger M, Koch CP, Reker D, Todoroff N, Schneider P, Rodrigues T, et al. Chemically advanced template search (CATS) for scaffold-hopping and prospective target prediction for 'orphan'molecules. Mol Inform. 2013;32(2):133-8.
29. Jiang D, Wu Z, Hsieh C-Y, Chen G, Liao B, Wang Z, et al. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. J Cheminform. 2021;13(1):1-23.
30. Wallach I, Heifets A. Most ligand-based classification benchmarks reward memorization rather than generalization. J Chem Inf Model. 2018;58(5):916-32.
31. Sundar V, Colwell L. The Effect of Debiasing Protein Ligand Binding Data on Generalisation. J Chem Inf Model. 2019.
32. Mendez D, Gaulton A, Bento AP, Chambers J, De Veij M, Félix E, et al. ChEMBL: towards direct deposition of bioassay data. Nucleic acids research. 2018;47(D1):D930-D40.
33. Servén D, Brummitt C. pyGAM: generalized additive models in python. Zenodo DOI. 2018;10.
34. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of machine learning research. 2011;12(Oct):2825-30.
35. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020;585(7825):357-62.
36. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17(3):261-72.
37. VanderPlas J, Granger B, Heer J, Moritz D, Wongsuphasawat K, Satyanarayan A, et al. Altair: Interactive statistical visualizations for python. Journal of open source software. 2018;3(32):1057.
38. Satyanarayan A, Moritz D, Wongsuphasawat K, Heer J. Vega-lite: A grammar of interactive graphics. IEEE transactions on visualization and computer graphics. 2016;23(1):341-50.
39. Kim J-H. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Comput Stat Data Anal. 2009;53(11):3735-45.
40. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (voc) challenge. Int J Comput Vision. 2010;88(2):303-38.