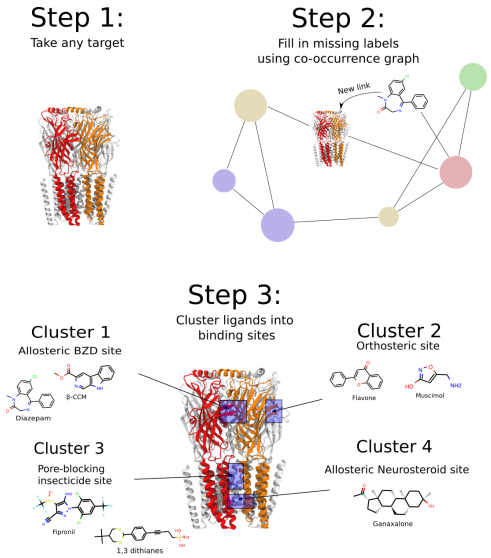
The missing label problem: Addressing false assumptions improves ligand-based virtual screening

Lewis Martin[1,*] and Michael T Bowen[1,*]
[1]The University of Sydney, Brain and Mind Centre, The Lambert Initiative for Cannabinoid Therapeutics, Sydney, NSW, 2006, Australia
*Correspondence: lewis.martin@sydney.edu.au, michael.bowen@sydney.edu.au

Abstract

Ligand-based virtual screening (LBVS) uses machine readable representations of chemicals to learn a mapping function that can predict binding interactions with protein labels. Because it is highly scalable it is increasingly used in drug development in academic and pharmaceutical contexts. We have identified assumptions commonly used in LBVS that are false, which collectively can be described as the missing label problem. Firstly, many of the binding interactions in the bioactivity databases typically used to train LBVS models have never been tested before, but the absence of a label is interpreted by most models as a true negative. Secondly, many proteins have multiple binding sites with unrelated shapes but the associated ligands are grouped together under the one protein label. These assumptions frustrate the ability of the model to learn a correct mapping function. Here we use statistical techniques to predict values for the missing labels and binding sites and show how this improves the ability of LBVS models to rank ligands correctly. In the process we introduce a new technique for removing bias during model evaluation based on data blocking from experimental design theory. All data and code for analysis and generating figures is publicly available on github (https://github.com/ljmartin/Missing_label_problem).



Introduction

Screening of ligands against protein targets is an essential part of drug discovery, target identification, and toxicology. Virtual screening (VS) seeks to predict protein targets using computer modelling, which is both faster and cheaper than *in vitro* approaches, and is thus increasingly integrated in drug discovery and development[1]. VS approaches can be split into two groups: structure-based VS, which requires a crystal structure to which ligands are fit using a scoring function, i.e. docking; and ligand-based VS (LBVS), which uses similarity to known active drugs to determine possible targets[2]. LBVS often uses molecular fingerprints as machine-readable descriptors. A de facto standard molecular fingerprint is the substructure fingerprint, which uses binary categorical variables – i.e. the presence or absence of substructures – to create a vector representation using multi-hot encoding. Similarly, the protein targets bound by a ligand, here referred to as the 'labels', can be considered as vectors where each entry is a binary variable that indicates the presence or absence of a binding interaction. Thus, common LBVS paradigms use statistical techniques called machine learning (such as regression, neural networks, or decision trees) to learn a function that maps the ligand vectors to the label vectors[3]. This process is sometimes called classification. In the prediction stage, untested molecules are first transformed into a molecular fingerprint ligand vector and are then fed through the mapping function to generate the predicted label vector. We have identified some missing label assumptions commonly used in LBVS that are simple to demonstrate as false. Firstly, treating untested interactions in the label vectors as true negatives learns a potentially incorrect mapping from ligand vector to label vector, as in fact the true nature of these interactions are unknown. Secondly, the grouping of ligands that bind different sites on a single protein under the one label is inconsistent with the similar property principle underlying LBVS, and ignores the fact that many proteins have multiple, discrete binding sites with different shapes. We show that by addressing these assumptions we can substantially improve LBVS performance

The LBVS task is closely analogous to 'multi-label learning' and could benefit from some of the approaches used in that field, in particular the use of different scoring functions and addressing missing labels. Multi-label learning is so-called because each instance of the data is associated with multiple labels simultaneously[4]. This contrasts with multi-class learning where each instance is associated with a single label from multiple, mutually exclusive, labels, and with single-label learning where only one label is learned. An extension to multi-label learning, extreme multi-label learning, learns hundreds or thousands of potential labels compared to only tens of labels typically considered in multi-class problems[5]. An example of a multi-label learning problem is scene classification - amongst thousands of potential labels, a given image could be mapped to all of *beaches*, *sunsets*, *parties* and more[6]. Similarly amongst thousands of protein labels any ligand can potentially bind to multiple proteins; for example imatinib, which binds the therapeutic fusion-protein target BCR-ABL but also the off-target C-Abl[7].

One issue shared by these tasks is how best to evaluate predictive performance during the testing stage. Commonly used metrics like precision and recall measure the ability to retrieve the known, true labels. Most training examples, however, are incompletely labelled due to the time-consuming and expensive process of

testing every possible label in advance. Thus, in multi-label settings even perfectly valid, newly predicted labels can be scored as false positives if the label was previously unknown. An alternative comes from the multi-label learning literature, in which models are evaluated only by the ability to rank known positives before unknown labels[8]. Ranking can be considered as a generalization of multi-class classification for the multi-label case[9]. A common ranking metric, the ranking loss, rewards prediction of both known positives and unknown positives as long as the known positives come first[4]. This is well-suited to drug discovery or target identification projects, in which a ranked list of predictions is often tested using *in vitro* assays until some cut-off point, where the predictions are no longer informative. Practically, directly optimizing for ranking thus reduces expense and time in drug development, while theoretically it acknowledges the presence of missing labels.

In the training phase, many machine learning algorithms assume the label vectors used as training data are complete and true[10]. As an example, if a ligand is active at a protein label then that position in the label vector is a 1. Conversely if the ligand-pair has not been tested then that position is a 0, which is treated as an explicit, known negative. In reality, the vast majority of label vectors consist of known positives and *unknown* positives/negatives[6]. The consequence is that, assuming we admit that bioactivity data used for training is incomplete, *all LBVS models are learning an incorrect mapping from ligand vector to label vector*. The gold standard solution would be to perform binding assays on each individual protein-ligand pair used in training, but this is infeasible at present - a recent count of the data in ChEMBL has 3569 human proteins and ~1.8 million distinct compounds, implying ~6.5 billion possible interaction points[11]. An alternative is to fill in the most likely missing points in the label vectors by treating the set of vectors as a network graph and using co-occurrence trends. This changes the multi-label learning task into a link prediction problem, which has been applied to tasks such as search engine and social network classification[12,13] but, to our knowledge, has not been used to solve the missing label problem in LBVS. Assuming the predicted labels are consistent with the ground truth, i.e. they really come from the same population as the true positive ligands, these new labels should help the models to correctly rank test ligands, providing a simple evaluation of the benefit of this approach. Conversely, if the newly predicted labels are spurious then the ranking loss should become worse. As well as improving ranking for test ligands, filling in the missing labels has the additional benefit of predicting new labels even before training a model and can thus be used as a classifier itself.

A related labelling problem that has not yet been addressed in the LBVS literature is the grouping of multiple binding sites for a particular target under the single label for that protein target. In the pharmacology literature the existence of multiple binding sites on a single protein has long been recognised[14], but this has not yet been acknowledged in LBVS. Extreme examples of proteins with multiple sites are the ligand gated ion channel proteins, which can have as many as 15 or more binding sites[15]. Recently, *in silico* and *in vitro* fragment screening results have indicated that GPCRs[16] and enzymes[17] also have multiple binding sites, giving experimental evidence to the suggestion that multiple binding sites at individual proteins is the norm across all dynamic proteins, including the major drug target-types, rather than an exception[18]. Furthermore, even within one single binding

site, some ligands with unrelated structures can bind by accessing different conformations of the site known as binding modes[19].

This grouping of binding sites under a single target label violates the similar property principle underlying LBVS, which posits that ligands with the same label have similar structures[20]. Analogously to missing labels, violating this principle is expected to reduce the ability of statistical models to learn the structural determinants of binding by forcing them to learn commonalities between chemical substructures that, in reality, have no similarity. Determining the true binding site for a ligand usually requires either mutation or crystallization, an unscaleable process that cannot realistically be applied to the number of ligand-target interactions in major bioactivity databases. We propose an *in silico* alternative that uses clustering to group ligands active at a particular target into structurally related subsets. We show that this approach successfully recognises multiple binding sites, recognises scaffold hops, and labels the majority of ligands with their correct site using an example dataset (a $GABA_A$ receptor from ChEMBL) where the binding sites for the major classes of active ligands are known and well characterized. Importantly, applying the clustering algorithm to a larger dataset from the ChEMBL bioactivity databse improves the ability of LBVS models to rank ligands.

The two goals of our work – predicting missing labels and introducing new labels for binding sites – may seem to abstract the label vectors from the ground truth, making evaluation of the new label vectors difficult. To remedy this, we sought an evaluation method that is robust to possible memorization bias, also known as test-train leakage. Recent LBVS literature has rightly pointed out that randomly selecting ligands from the training set to create a test set for evaluation can result in overly optimistic performance estimates that do not align with prospective validation[21-23]. This occurs because the ligands in most bioactivity datasets are not independent and identically distributed – the discovery of one active ligand often leads to many highly similar structural analogues with only a few changed atoms[24], suggesting that the number of independent data points in most LBVS datasets is far fewer than the actual number of ligands. Predicting the activity of these same-scaffold ligands is trivial for machine learning techniques that are able to memorise the ligand vectors used in training, but it does not generalize to so-called 'out-of-sample' data that have unseen scaffolds. A more realistic evaluation uses time-split cross validation - that is, removing blocks of ligands from a contiguous time period to be the test set[25-27]. This approach likely originated in financial time-series analysis where it is known as backtesting[28]. Time-split cross validation shows a reduction in memorisation bias but is still susceptible to memorization since, unlike financial time series, molecular bioactivity datasets are non-linear and have uneven distributions of scaffolds after the discovery date as well as the possibility of train-test splits with zero test ligands[27].

We propose a more robust method to avoid bias by randomly choosing test ligands and explicitly removing all highly-correlated, non-independent, structural analogues from each possible training set. This method inherits from blocked experimental design, and we suggest that all LBVS practitioners should be mindful of this statistical background due to the high correlation inherent in bioactivity datasets. The goal of block design is to group the

experimental units into “blocks” that are as uniform as possible to avoid measurement of effects orthogonal to the conditions of interest[29,30]. To evenly sample all ligands and account for the non-linearity of ligand sets, we repeat the test set evaluation hundreds of times until the performance metric stabilises in a process inspired by bootstrapping. Removing similar ligands from the training set by a distance metric has been proposed before – such as asymmetric validation embedding (AVE)[21] and maximum unbiased validation (MUV)[31] - but these functions search for the global minimum of a bias function rather than rotating all ligands through the test set. As a result the evaluation may be less robust for assessment of out-of-sample data, which is often the intended target of LBVS.

Our approach is motivated by an empirically-derived quantitative definition of correlated structural analogues used as a distance cut-off. It removes a single group of highly similar analogues in each case, akin to leave-one-out sampling[32] but instead leaving out blocks of correlated data. It thus eventually samples all scaffolds while maximising the available training data to more closely approximate out-of-sample data and generalizability estimates for prospective LBVS. This leads to a robust evaluation metric for evaluating manipulations of the label vectors.

In this research article we show improvements to the de facto standard method of LBVS using a common experimental configuration – molecular substructure-based ligand vectors, protein label vectors from the ChEMBL bioactivity database, and a naïve Bayes classifier for predicting label vectors. In Part 1 we introduce the bootstrapping procedure for robust, bias-free evaluation of model performance, then we use this to show how filling in missing labels (Part 2) and clustering ligands into binding sites (Part 3) both substantially improve ranking.

Results

The data

The dataset used in the following experiments is made up of the ligands that bind to 243 proteins from the ChEMBL bioactivity database, which is an online database that automatically annotates bioactivity data published in several medicinal chemistry journals[11]. The proteins cover a range of protein families, are all ‘single protein’ records (which accords with common practice in the field), and each has at least 500 but no more than 5500 associated ligands bringing the total to 252,409 ligands. A schematic in **Figure 1** describes the LBVS process used here. Morgan molecular substructure fingerprints generated by the *rdkit* python library[33] are fed into a trained naïve Bayes classifier from the *sklearn* python library[34], which then predicts a label vector corresponding to the protein targets for that ligand. In each case, the classifier is trained using the pre-labelled dataset from ChEMBL described above. Rather than run a large number of expensive *in vitro* assays to evaluate the predictive performance, portions of the dataset are masked during training to use as a test set, as described in **Part 1**.

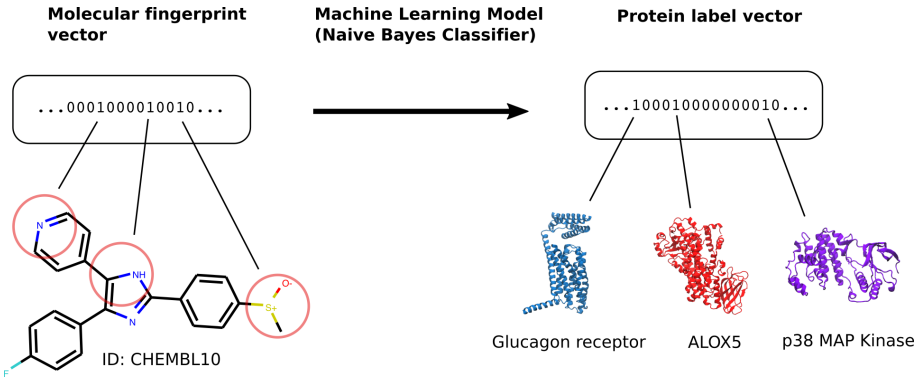


Figure 1. Example of the LBVS paradigm used here. The ligand vectors are multi-hot encoded categorical vectors, where each non-zero position corresponds to the presence of a molecular substructure. The label vectors are also binary vectors, where each non-zero entry corresponds to a binding interaction with a target protein. A machine learning model learns a mapping from ligand vector to label vector, which can then be applied to unseen ligand vectors to predict new label vectors.

Part 1. Evaluation using ranking loss and block bootstrapping

Evaluation of predictive models generally uses k-fold cross-validation, in which the dataset is separated into *k* blocks that are, in turn, each used once as a test set by being separated from the training data and then evaluated on prediction metrics (**Figure 2a**). When some ligands are highly correlated, such as for closely-related molecules resulting from structure-activity relationship analyses[24], this can lead to highly correlated data in both the training and test sets and thus overly optimistic performance metrics. An alternative that respects the non-linearity and non-independence of molecular bioactivity datasets is a blocked design in addition to bootstrapping with replacement, in which test sets are repeatedly removed at random until the performance metric converges (**Figure 2a**). To reduce the bias from highly correlated ligands, in every repeat the test set is made by first choosing a single ligand at random then, in addition to that ligand, selecting the *k* nearest neighbours up to some distance cut-off. We use the Dice distance as the distance metric between ligands, since it has a wider spread of pairwise distances as compared to some other commonly used metrics (see **Figure S1**).

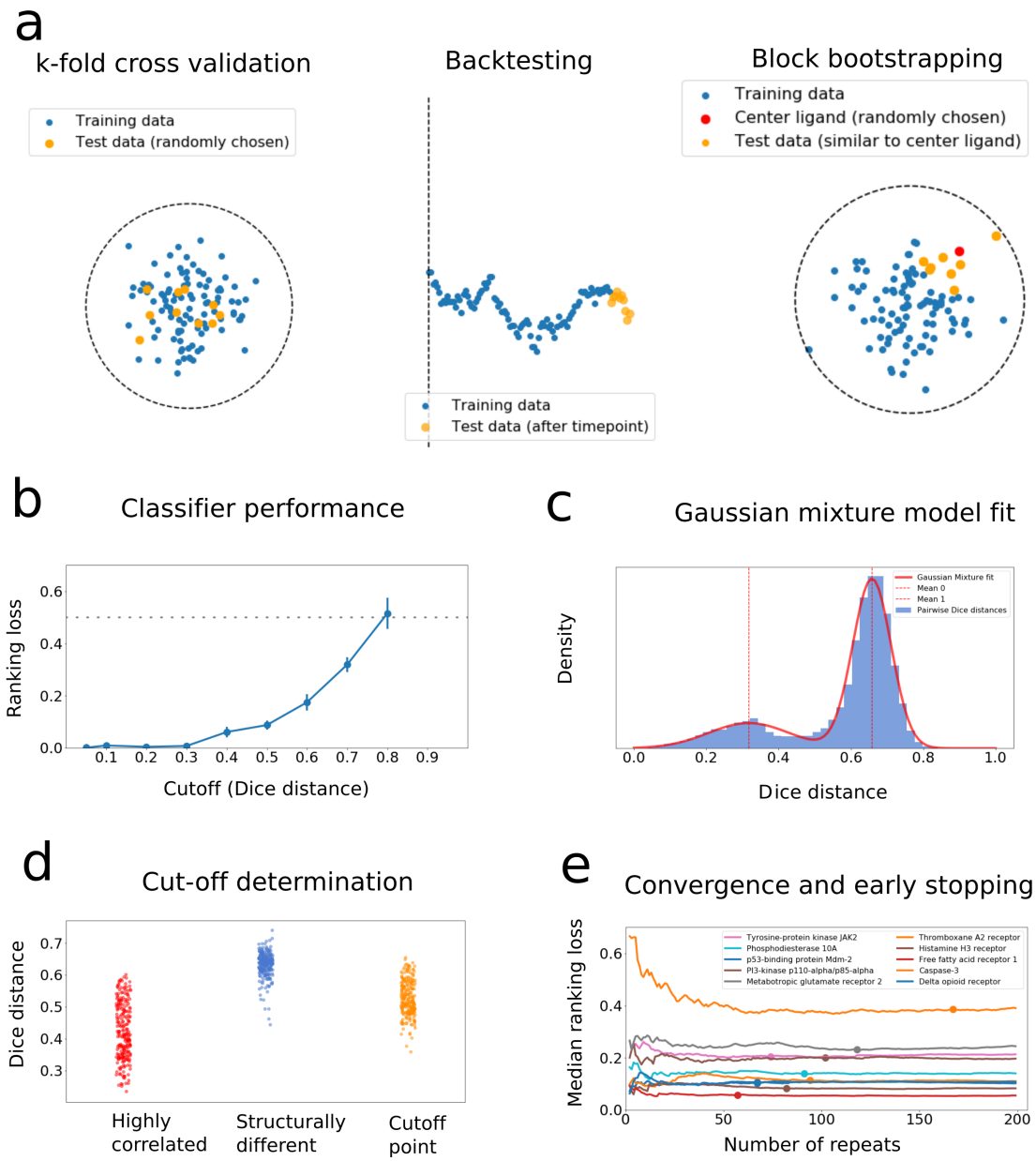


Figure 2. Block bootstrapping can evaluate model performance while also removing bias due to test/train leakage of highly-correlated data points. **a)** A common machine learning approach, k-fold cross validation, splits the data randomly such that each data point is in the test fold only once, but assumes that the data is independent and identically-distributed. Backtesting removes all data from a contiguous timeframe, but requires linearly separable data. For molecular data, these assumption are not true. Our bootstrapping-inspired approach repeats the test-train splits multiple times, ensuring that in every split a central ligand and all highly correlated neighbours are removed as the test fold, minimizing inclusion of structural analogues across both the test and train sets (test/train leakage). **b)** The median ranking loss of 20 randomly-sampled targets at different cut-off values using the bootstrapping sampler. Increasing the cut-off distance used to define the nearest neighbours removes more data from the training set, in turn reducing ability of the classifier to rank correctly. **c)** Histogram of pairwise Dice distances from the 'Protein kinase C theta' label, showing characteristic bimodal distribution (blue), fit using a Gaussian mixture model with two components (red, smooth). Also shown are the positions of the means of each Gaussian component (red, dashed). **d**) The means of the highly correlated (red) or different structure (blue) Gaussian components from all 243 protein labels along with the proposed cut-off points (orange), which are defined as the midpoint between the two components for each target. **e**) Ranking loss from block bootstrapping evaluation of 10 example targets at cut-off

d=0.525, which converges at higher number of repeats (coloured lines). Early-stopping criterion reduces computation by stopping when convergence has been reached (coloured points).

Increasing the distance cut-off leads to the removal of more nearest-neighbours, reducing the amount of information available in the training set to train the classifier (**Figure 2b**). Our goal was to identify the optimal cut-off distance that groups highly-correlated analogues, i.e. dependent data, together while maintaining as much independent data in the training set as possible. To our knowledge all previous methods use an arbitrary cut-off or rely on potentially biased, human-defined, definitions of scaffolds such as Murcko scaffolds. We prefer an empirical approach, starting from the intuition that any given pair of molecules are either a pair of analogues or are independent. To find the cut-off that splits these two types of pairs, we fit the pairwise distance distributions from the ligands in each label using a Gaussian mixture model with *n*=2, reflecting the two possible states for any given pair (analogue or independent), using the *a priori* knowledge that the distributions are made up of both the analogue pair distances and independent pair distances. Consistent with this, pairwise distance distributions of ligand sets have a characteristic bi-modal shape, one example of which is shown in **Figure 2c**. Fitting the Gaussian mixture model to all 243 protein targets, and recording the Gaussian means, estimates the mean of the analogue and independent pairwise distances. The Gaussian centres show a clear separation between the two populations at lower and higher Dice distance (**Figure 2d**).

Setting the cut-off to the value in-between the lower and higher distributions should provide the optimal cut-off point to remove structural analogue pairs from every training set. Averaging over the means of the Gaussian fits, and setting the middle value as the cut-off, led to a 95% confidence interval of separating Dice distance of 0.520 to 0.536 (**Figure 2d**). All of the following evaluations use Dice distance d = 0.525. Visual inspection of 6 ligand pairs, randomly chosen from 6 random targets. with $d < 0.525$ shows that all have large maximum common substructures, while pairs with $d > 0.525$ selected in the same way show more diverse structures (see **Figure S2** for the comparison of ligand pairs). This cut-off will thus strike a good balance between removal of structural analogue pairs and retaining differently structured pairs, which fulfils our goal of identifying an optimal Dice distance cut-off.

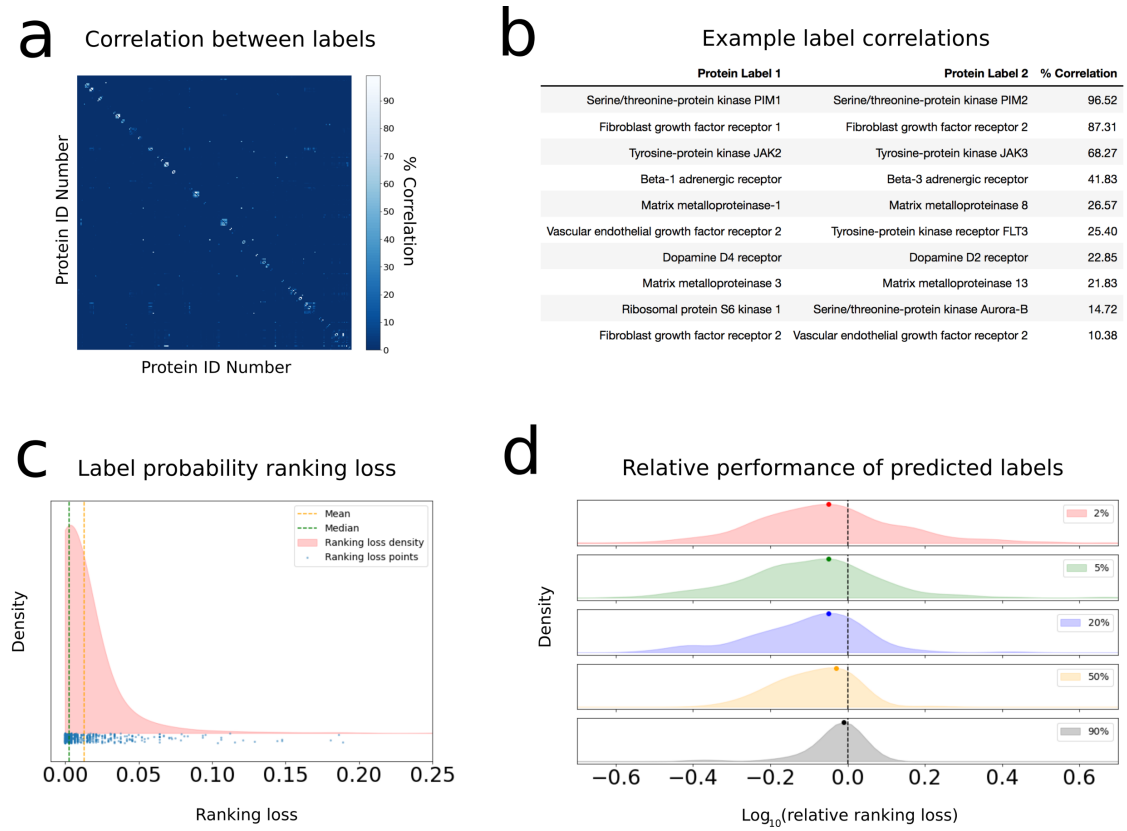
Compared to *k*-fold cross validation the bootstrapping procedure proposed here incurs higher computational costs. This is one of the reasons for using naïve Bayes classifiers, which are highly scalable and thus have short model building times (other reasons are that these classifiers are ideally suited for one-hot encoded binary data[35]). Bootstrap metrics converge on their expected value in the limit of the numbers of repeats[36]. So, to reduce calculation times, we implemented an early stopping criterion that prevents unnecessary repeats after reasonable convergence, which is defined as the maximum percentage change in median ranking loss over the previous 50 trials of lower than 2.5%. This procedure allows more repeats for labels with higher variance, but shorter calculation time for labels with fast convergence (**Figure 2c**). The code used to perform the bootstrapping

experiments is available as a Python module called the 'VirtualScreeningBootstrapper', which takes as input a ligand vectors matrix, a label vectors matrix, and an *sklearn* classifier.

Part 2: Filling in missing labels improves performance

Training most machine learning models requires the input of label vectors that describe the true positive and true negative labels. When the training data is incomplete due to high cost of determining the labels, this process leads to learning an incorrect mapping of the input ligand vectors to the label vectors. A possible solution is using co-occurrence trends in the matrix made up of all the label vectors to predict the missing labels. In this process, the matrix has columns representing each protein target and rows representing the label vectors for each ligand. When a ligand co-occurs at two proteins or more (that is, it has more than one label) it indicates the binding sites at the two proteins have some degree of similarity. Iterating over all ligands with 2 or more labels leads to a label correlation value, calculated as the percentage of ligands for each protein that co-occur at a second label. All proteins in this dataset have at least 500 ligands, which avoids high correlation values occurring by chance from a few shared ligands.

The pairwise label correlations are shown as a heatmap in **Figure 3a.** Clearly some proteins have highly similar binding sites, with correlation values above 80%. A random sample of highly correlated protein pairs is given in **Figure 3b** as an example. Judging only by the names, these pairs align well with a phylogenetic understanding of the protein labels – the most highly correlated pairs are subtypes that split from a common gene, likely having similar protein sequence and thus similar binding sites. This suggests that the ligands with only one label from a correlated pair are likely to also bind to the second protein, but that this interaction either has simply not yet been tested or is not recognised by the automated annotation process used by ChEMBL.



Protein Label 1	Protein Label 2	% Correlation
Serine/threonine-protein kinase PIM1	Serine/threonine-protein kinase PIM2	96.52
Fibroblast growth factor receptor 1	Fibroblast growth factor receptor 2	87.31
Tyrosine-protein kinase JAK2	Tyrosine-protein kinase JAK3	68.27
Beta-1 adrenergic receptor	Beta-3 adrenergic receptor	41.83
Matrix metalloproteinase-1	Matrix metalloproteinase 8	26.57
Vascular endothelial growth factor receptor 2	Tyrosine-protein kinase receptor FLT3	25.40
Dopamine D4 receptor	Dopamine D2 receptor	22.85
Matrix metalloproteinase 3	Matrix metalloproteinase 13	21.83
Ribosomal protein S6 kinase 1	Serine/threonine-protein kinase Aurora-B	14.72
Fibroblast growth factor receptor 2	Vascular endothelial growth factor receptor 2	10.38

Figure 3. Filling in missing labels using a label correlation graph as probabilities improves the ranking performance of LBVS models. **a)** Heatmap representing the label correlation graph. Some pairs of labels show very high percentage of shared ligands, implying high similarity in binding sites. The graph is not symmetrical - if a pair of labels have different amounts of associated ligands, the percent correlation will be different depending on which protein is being compared to the other. **b)** Some example pairs of labels with high ligand correlation. Many such pairs have shared phylogenetic history (based on their names), increasing confidence that they have similar binding sites. **c)** Ranking performance of the label prediction approach on 1000 ligands with multiple labels. The median ranking loss of ~0.012 compares favourably against the LBVS classifiers shown in **Figure 2e**, indicating it is a viable approach to predicting labels. **d)** Ranking performance of LBVS using the new label matrices predicted by the label correlation approach, relative to the original label matrix.

In order to use the correlations between proteins to fill in missing labels, we assumed each protein correlation is independent and calculated a per-protein binding probability for each ligand (see methods). For each ligand with at least two labels, we individually removed that ligands' influence on the correlation matrix and then calculated the probability that each label would be predicted using the label correlation data. Comparing the probability of the known label to the probabilities of the unknown labels in that vector gives the ranking loss. Ligands with only one protein label cannot be used for this evaluation because removing the only label means there are no labels left to calculate a probability. The ranking losses in **Figure 3c** show a distribution with a mean ranking loss of ~0.012 and median of ~0.002, which is approaching perfect ranking. Using the calculated probability values, we filled in the missing labels using different thresholds, which at lower thresholds predicted several hundred thousand new labels (see **Figure S3**).

To determine if the new labels predicted by this thresholding approach are consistent with the existing labels, we evaluated the label matrices using block bootstrapping, removing only the known ligands as the test set but including the newly predicted ligands as positives in the training set. If the labels predicted by the label-correlation-approach are indeed true positives, then their presence in the training set should help the classifier to rank the test set, reducing the overall ranking loss. The relative ranking losses using predicted labels at multiple probability thresholds are shown in **Figure 3e**, where density to the left of the dotted line indicates improved performance as judged by lower ranking loss. Even at the highest threshold of 90% (which has the fewest newly predicted labels) most, but not all, protein labels perform substantially better when including the newly predicted labels. At lower probability thresholds, the majority of labels still perform better but the spread is increased. This has a remarkable implication: At the lowest threshold of 2%, the label matrix has an approximately four-fold increase in the number of labels, equivalent to 1,142,376 new labels. Most labels perform better when including the newly predicted labels, suggesting most of the predictions are true. The scale of this level of predictions dwarfs the number of predictions ordinarily considered in virtual screening, all while subsequently improving performance of the classifiers. Importantly, the technique is highly adjustable depending on the practitioner's level of acceptable confidence, and a middle ground that shows a good balance between a conservative number of newly predicted labels and improved performance is a threshold of 20%. Ultimately, filling in missing labels using the correlation graph as probabilities, with a 20% threshold, shows an improvement in ranking loss of ~20% averaged over all targets in this dataset.

Part 3: Clustering into binding sites improves performance

The lack of knowledge about protein binding sites can be considered another type of missing label problem, which reduces performance by grouping ligands from different binding sites under the same label. Because there is no expectation that the binding sites on a protein have the same structure (excluding the identical sites on homomeric proteins), by the similar property principle one can deduce that the grouped ligands also have unrelated structure. In order to separate grouped ligands into their binding sites without *a priori* knowledge from mutation or crystallization evidence, we first use clustering and then merge clusters based on their relative similarity to two distributions of Dice distances. The distributions, shown in **Figure 4a**, are the pairwise Dice distances from ligand pairs within a single label or between different labels. This serves as a proxy for taking Dice distances from ligand pairs within a single binding site or between different binding sites.

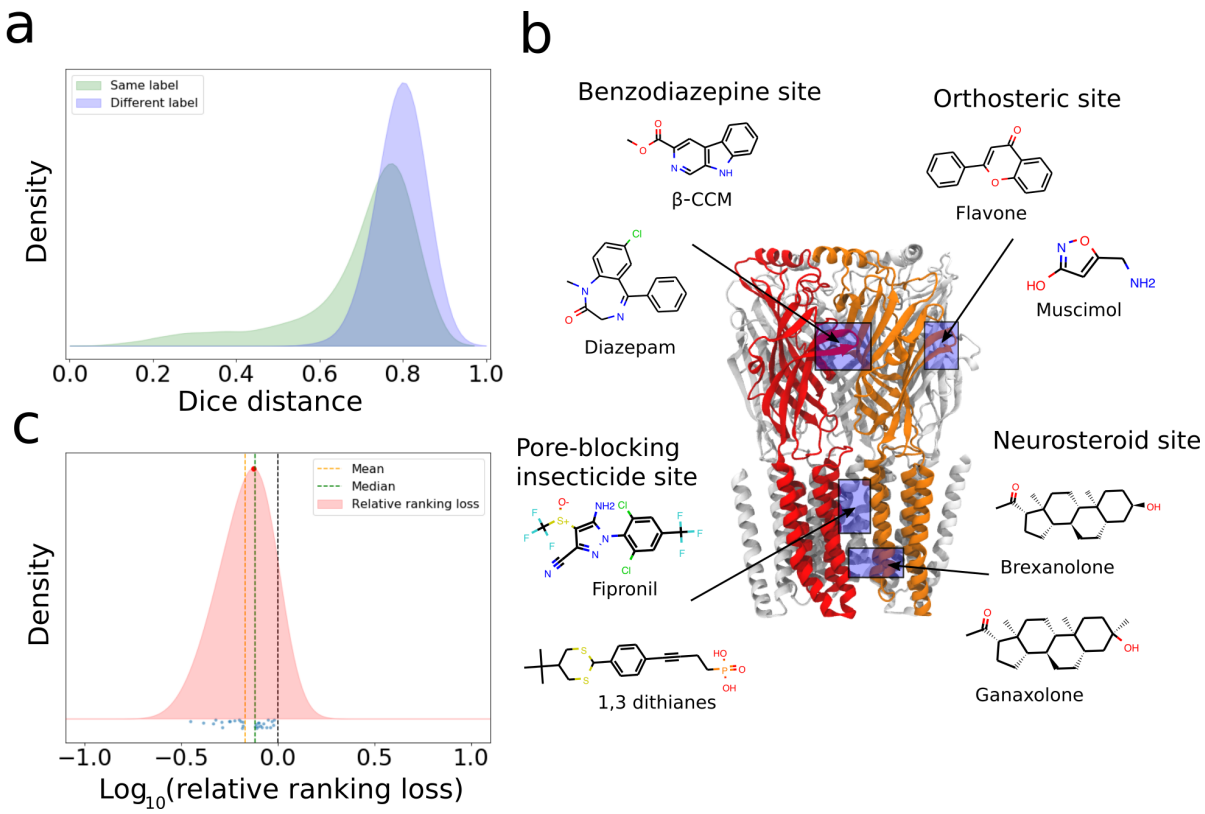


Figure 4. Ligands can be grouped into binding sites using clustering. **a)** Two distributions of pairwise Dice distances taken from within a label (green) or between two labels (blue). This approximates the Dice distance distributions from within a binding site and between different binding sites. Thus, these distributions can be used to merge proposed binding clusters by taking the Kolmogorov-Smirnov distance. **b)** The clustering and merging approach is capable of recognising multiple binding sites for the ligands of a $GABA_A$ receptor. The process does not simply group by scaffold, and correctly groups together different scaffolds that bind to the same site. **c)** Performance of the label vectors that underwent clustering compared to the un-clustered labels. A majority perform better than the un-clustered counterpart.

To separate ligands into binding sites, first the ligands from within a label are clustered using agglomerative clustering. This generates several proposed binding site groups, which can be merged if they resemble the same-label distribution more than the different-label distribution. We used the Kolmogorov-Smirnov statistic to determine the distribution similarity. Applying this process to a label from ChEMBL with multiple known binding sites demonstrates the technique's potential (**Figure 4b**). The protein label used is CHEMBL2093872, which corresponds to a protein complex group of $GABA_A$ receptors with 365 active ligands including neurosteroids, insecticide pore-blockers, benzodiazepine or non-benzodiazepine positive allosteric modulators, and orthosteric site ligands such as GABA. Importantly, these different ligand classes have different and well characterised binding sites at $GABA_A$ receptors. Our combined clustering and merging technique accurately groups two different scaffolds of pore-blockers at a single binding site (demonstrating it is not simply clustering based on a single scaffold), as well as recognising the neurosteroid site binders. The remaining two clusters largely correspond to either the extracellular allosteric site, which includes multiple scaffolds like diazepam and β-carbolines, or the orthosteric site, which includes multiple scaffolds such as the flavones and muscimol. The clustering isn't quite perfect, with the orthosteric ligands pitrazepine and GABA being grouped in the

extracellular allosteric site, but it is a substantial improvement on the original single label. This suggests the technique can be used to improve the label vectors used in LBVS.

The clustering and merging technique was applied to the whole dataset and split 29 targets (approximately 12% of the number of targets) into either two sites or three sites each. The ranking loss of the new target labels is compared to the original, unclustered labels in **Figure 4c**, where density to the left of the dotted line indicates improved ranking (lower ranking loss). To compare the performance of ligand sets that are split into multiple labels (one for each proposed binding site) to that using the single original label, the ranking loss of all clusters was calculated separately and then combined using a weighted average with the weights determined by the number of ligands relative to the single original label. All of the clustered ligand sets perform better than their un-clustered counterpart. This represents a substantial improvement on its own and, as with filling in missing labels, is highly adjustable in that practitioners can choose which targets to keep clustered based on the relative performance. In addition to this, the clustering approach gives access to large numbers of new labels in ChEMBL: Currently the most commonly used protein labels are the 'single protein' records because it is assumed all the ligands will be structurally related since they bind to a single protein but, with clustering into binding sites, other protein type records such as 'protein complex', 'protein complex group', 'chimeric protein', 'protein family', and 'protein-protein interaction' can also be used, increasing the amount of training data and thus the chances of determining the correct label for novel test ligands. Ultimately, filling in missing binding site labels leads to a mean improvement in the ranking loss of approximately 30% for proteins with multiple binding sites.

Discussion

Many LBVS approaches use assumptions that may have arisen out of necessity or simplicity but limit performance. In particular:

- The use of incomplete label vectors to train machine learning models, which treat unknown interactions as true negatives when some are actually positive
- Grouping different binding sites under the same label, which forces models to learn similarities between unrelated chemical substructures

Collectively, these two assumptions can be viewed as missing label problems.

Ultimately, they stem from a single assumption: that the bioactivity data used for training is correct and complete. If this were true, then no more interactions between proteins and ligands present in ChEMBL need be tested. In reality both assumptions are false in many cases. Continued updates of the ChEMBL database is an implicit recognition of this - here we make this recognition explicit, addressing the missing label problem using a more complex but more correct approach to LBVS. As we showed, this improves the performance of the trained models, which suggests that the newly predicted, formerly missing, labels are accurate. The common assumption of independent and identically distributed data has previously been recognised as problematic[21-23,25-27]. So, to show improvements in addressing missing labels, here we also developed a new method of model

evaluation called block bootstrapping that is more computationally intensive but explicitly avoids biased evaluation by removing structural analogues from each and every test/train split.

The key to addressing the simplifying assumptions in LBVS is borrowing techniques from other fields. As a precedent, the proposal for time-splits to replace random-splits for cross validation may have been inspired by financial time-series analysis where it is known as backtesting[28], although we aren't aware of an explicit reference to that field. Our proposal for block bootstrapping improves upon backtesting by handling the non-linearity of ligand bioactivity datasets at the cost of increased computation. While the exact cut-off used to separate highly-correlated pairs from independent pairs will differ by the choice of molecular fingerprint, the use of Gaussian mixture models to determine the cut-off is applicable to any dataset. A previous approach to de-biasing, the AVE bias, initially uses randomly selected training and test sets followed by a genetic algorithm to reduce bias between the sets[21]. This technique appears to effectively reduce bias, but the global minimum of the bias function may be unique and so, by definition, does not including all scaffolds in the test set over the course of evaluation. Our method, using a blocked design, is akin to leave-one-out sampling but instead each test set leaves out a block of highly-correlated analogues. After enough repeats, it thus selects all scaffolds to be in the test set and converges on a performance value that uses all of the available data.

The multi-label learning field has addressed the shift to ranking as a generalization of multi-class learning, which in contrast commonly uses precision and recall as metrics[9]. When some labels are unknown rather than true negatives, as in LBVS, precision penalizes the prediction of new labels that may be true. At the same time, recall may be optimistic because there are unknown positives that aren't counted. An extension of precision to the multi-label case is precision at *k* (p@k)[37-39]. For search engines, as an example, this metric measures the proportion of relevant, i.e. true positive, results returned in the top *k* results, where *k* refers to the number of items an average user might reasonably browse[8]. For LBVS, the value of *k* is impossible to define without biasing against highly selective ligands (using large *k*) or in favour of highly non-selective ligands (using small *k*). To remedy this we used ranking loss, which is equally applicable to both selective and non-selective ligands as well as forgiving when models predict unknown positives as true positives - as long as they are ranked below the known positives.

The approach to filling in missing labels using the label correlation graph was inspired by Tan et al.[40], which is also an extreme multi-label learning setting based on the number of labels. In that work, when items have multiple labels the probability score is the sum of the correlations with each other label. Clearly this can lead to probabilities above 1, which we found unintuitive. Instead, we assumed that each correlation is independent of the others meaning the correlations can be multiplied to generate a final probability. This is akin to the 'naïve' independence assumption used in a naïve Bayes classifier and, while not totally correct, appears also to be an effective assumption[41]. A possible improvement on this technique is to take dependence relations into account when calculating probabilities.

Since clustering into binding sites is a highly task-specific problem, we found no analogy in other fields. Nevertheless, the use of distributions of Dice distances between ligands to merge pairs of clusters is reminiscent of the pioneering similarity ensemble approach (SEA) to LBVS[42]. In SEA, the distance scores of two groups of ligands were fit using an extreme value distribution to generate expectation values. In comparison, our approach uses a non-parametric fit. The good performance on a $GABA_A$ receptor dataset as well as improved ranking loss and simple implementation recommends this approach.

Finally, we comment on the multi-label setting. This work solely uses 'binary relevance', which is a multi-label learning term meaning that a single classifier is fit for every label[43]. This is a convenient problem setting that allows for our block bootstrapping technique, but more advanced problem settings are possible. In particular, random forests, neural networks, and various problem transformation techniques can fit multiple labels at once to take advantage of the similarities between labels and improve performance[43]. Binary relevance was necessarily used here to demonstrate the benefit of addressing missing labels, since the block bootstrapping technique is not applicable to other multi-label problem transformations. Nevertheless, improvements to the label matrix should be classifier- and problem transformation-agnostic and benefit all multi-label learning approaches.

Here we have shown how several simplifying assumptions used in LBVS are false. We provide solutions that address these assumptions and substantially improve the predictive performance of LBVS machine learning models. To do this we developed a new method for evaluating LBVS models based on bootstrapping that is guaranteed to avoid scaffold memorization since it explicitly removes similar structural analogues from every training set. Using this method, we showed how filling in missing labels in the label vectors not only corrects the unsatisfying situation where statistical models are built using knowingly incomplete label vectors that are falsely assumed to be complete, but also improves the ultimate performance of those models. Similarly, the labels of proteins with multiple binding sites have been improved by clustering the ligands into their binding sites, removing the spurious grouping of unrelated ligands under the one label.

Methods

All data and analysis scripts, along with IPython notebooks describing the process, are available at https://github.com/ljmartin/Missing_label_problem

Data

Protein ligand interaction data was downloaded from ChEMBL24[11]. All proteins are 'single protein' records, with activity at a protein defined as a *pchembl* value greater than 5, the equivalent of <10μM. Proteins were kept based on having greater than 500 but less than 5500 active ligands. Ligands were stored and manipulated in python by converting the SMILES strings into molecule objects from the python *rdkit* library[33]. The ligand set was sanitized by removing ions and ligands with molecular mass less than 90amu or greater than 80. All ligands were featurized as Morgan fingerprints with radius=2 and folded to 256-bit binary vectors using the *GetMorganFingerprintAsBitVect* method in the rdkit library. These formed the instance vectors used for model training. The label vectors were generated using the *MultiLabelBinarizer* method available in the *sklearn* python library[34].

Model training

Predictive modelling of protein/ligand interactions was performed using a Bernoulli naïve Bayes classifier as implemented by the *sklearn* python library[34], using default settings i.e. Laplace smoothing parameter of 1. This type of classifier is designed for single-label classification tasks, while the label matrix is inherently multi-label. In order to use a naïve Bayes classifier, model training was performed using the 'binary relevance' problem transformation technique, which splits the labels into multiple single label learning tasks. As a result, each protein target is evaluated individually, and separately from the others. The full ligand set was used as available testing and training data for each single-label learning task.

Model evaluation by block bootstrapping

Model evaluation was performed by calculating a performance metric on test sets made up of ligands that were masked from the model during fitting. After fitting, the model was used to generate predicted probabilities for a label for each of the masked ligands, which were then compared to the ground truth labels for evaluation. The performance metric was label ranking loss as implemented by the *sklearn* library[34]. Due to the binary relevance problem transformation, the ranking loss was calculated using the ranked probabilities of the test ligands binding to a single protein target, thus measuring the ability of the model to rank true positives before unknowns for each single target.

Evaluation used our block bootstrapping approach, which evaluates the model on multiple training and test sets until the median of the performance metric measurements converges. In each iteration of this process the test

positives are selected by first choosing a true positive ligand at random and then masking that ligand as well as all of its nearest neighbours up to some cut-off. Nearest-neighbours were defined by the Dice distance between the molecular fingerprints. For each iteration, the test negatives were chosen by randomly selecting 10% of the ligands that are not true positive. The error of a bootstrap measure converges to zero in the limit[36]. To save computation time, satisfactory convergence was defined by thresholding the maximum change in the median of the performance metric over several previous trials. A threshold of 2.5% was used so that when the performance metric had changed by less than 2.5% in either direction over the previous 50 trials, no further iterations were performed.

Gaussian mixture model fitting

A Gaussian mixture model was fit to the set of pairwise distances of the true positive ligands associated with each target to determine the most likely same-scaffold and different-scaffold distances. Pairwise distances were calculated using ligand vectors and the *pairwise_distances* method available in the *sklearn* Python library[34], with metric='Dice'. Gaussian mixture models were fit using all pairwise distances, except for the distances to self, using the *GaussianMixture* method in *sklearn,* with n_components=2. One mixture model was fit for each target, and the means of the Gaussians were recorded with the lower-valued mean attributed to same-scaffold pairs and the larger-valued mean attributed to different-scaffold pairs. The cut-off separating the means was determined as the value at the midpoint between the two Gaussian means for each target. The confidence interval of the cut-off was calculated using bootstrapping, taking 10,000 samples of the set of cut-offs at random with replacement. The 95% CI is calculated by recording the 250th and 9750th values of the ranked means of these bootstrapped samples.

Correlation graph

Percentage correlations between each pair of targets was calculated using the set of label vectors with 2 or more labels (label vectors with a single entry don't offer any information on the correlation between any pair of labels). The correlation matrix measures the percentage of ligands of each target that are shared with another target. The percentages in this correlation matrix are then used as probabilities to determine new labels. The probably p_i of an unknown label for the *i*th protein target being a positive is calculated as:

$$p_i = 1 - \prod_{j=1}^{n} 1 - c_{ij}$$

where n = the number of true positive labels present in the label vector being considered, and c_{ij} = the percentage correlation of the *i*th target with the *j*th target. In more simple terms and using the rolling of dice as an analogy, this is equivalent to asking "given *n* dice rolls, what is the probability that any one of them is a six?", which is equal to one minus the probability that none of them are six.

Values in the label matrix were set to '1', indicating a true positive, if the probability score was greater than some threshold value. All existing ground truth labels were retained. The new label matrices, using threshold p_i values of 0.02, 0.05, 0.2, 0.5, or 0.9 were then assessed for their effect on ranking using the block bootstrapping approach. To maintain relevance to the ground truth, the test set true positives were chosen only from the original, ground truth, while test set negatives were chosen only from the remaining labels that are not true positives. This means the newly predicted labels are always present in the training set. If the newly predicted ligands have ligand vectors that are consistent with the true positives in the test set, then they should improve ranking of the ground truth true positives, and vice versa. To measure the effect of including the newly predicted labels on ranking, for each target the log10 of the relative ranking loss, compared to the original label vectors, is reported.

Clustering into binding sites

Ligand sets for each target were clustered in order to group them into candidate binding sites. These clusters were subsequently merged based on statistical analysis. Clustering used the *AgglomerativeClustering* method in the *sklearn* library, with a precomputed distance matrix recording the Dice distances as input, n_clusters=None (because the number of binding sites is *a priori* unknown), affinity='precomputed', linkage='average' and distance_threshold=0.8. The distance threshold hyperparameter has not been optimized, but was chosen to slightly overcluster, generating a number of candidate binding sites that can then be subsequently merged.

Merging the clusters involved comparison to two distributions of Dice distances. One distribution was made up of pairwise distances within single target labels, representing Dice distances most likely to be from a single binding site, and another distribution was made up of pairwise distances from between different target labels, representing Dice distances most likely to be from different binding sites. These distributions were generated by 30 repeats of selecting two targets at random, and calculating pairwise Dice distances from ligands within or between the two target labels.

Candidate binding site clusters generated by agglomerative clustering were then compared to these two distributions by calculating a Kolmogorov-Smirnov (KS) statistic using the *ks_2samp* method available in the *scipy* python library[44]. The KS statistic is a measure of the distance between two distributions by comparing their cumulative distribution functions. It was preferred to other non-parametric techniques because it captures the influence of distribution shape, and the same-label distribution has a characteristically different shape to the different-label distribution (refer to **Figure 4A**). Thus, it can measure whether the given two clusters have greater likeness to the same-binding site distribution or the different-binding site distribution. The pair of clusters that have greatest likeness to the same-binding site distribution are merged until all possible pairs of clusters resemble the different-binding site distribution or until there is only a single cluster left.

Acknowledgements

MTB was supported by an NHMRC Doherty Biomedical Research Fellowship (1092046) and LM and MTB were supported by the Lambert Initiative for Cannabinoid Therapeutics, a philanthropically funded research program based at The University of Sydney. We would like to thank our colleagues at the Lambert Initiative who provided valuable insights and support at various stages of this project.

References

1 Schneider, G. Automating drug discovery. *Nature Reviews Drug Discovery* **17**, 97 (2018).
2 Sliwoski, G., Kothiwale, S., Meiler, J. & Lowe, E. W. Computational methods in drug discovery. *Pharmacological reviews* **66**, 334-395 (2014).
3 Lo, Y.-C., Rensi, S. E., Torng, W. & Altman, R. B. Machine learning in chemoinformatics and drug discovery. *Drug discovery today* **23**, 1538-1546 (2018).
4 Zhang, M.-L. & Zhou, Z.-H. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* **26**, 1819-1837 (2013).
5 Yu, H.-F., Jain, P., Kar, P. & Dhillon, I. in *International conference on machine learning.* 593-601.
6 Boutell, M. R., Luo, J., Shen, X. & Brown, C. M. Learning multi-label scene classification. *Pattern recognition* **37**, 1757-1771 (2004).
7 Hopkins, A. L. Network pharmacology: the next paradigm in drug discovery. *Nature chemical biology* **4**, 682 (2008).
8 Croft, W. B., Metzler, D. & Strohman, T. *Search engines: Information retrieval in practice*. Vol. 520 (Addison-Wesley Reading, 2010).
9 Madjarov, G., Kocev, D., Gjorgjevikj, D. & Džeroski, S. An extensive experimental comparison of methods for multi-label learning. *Pattern recognition* **45**, 3084-3104 (2012).
10 Bucak, S. S., Jin, R. & Jain, A. K. in *CVPR 2011.* 2801-2808 (IEEE).
11 Mendez, D. *et al.* ChEMBL: towards direct deposition of bioassay data. *Nucleic acids research* **47**, D930-D940 (2018).
12 Maimon, O. & Rokach, L. Data mining and knowledge discovery handbook. (2005).
13 Liben-Nowell, D. & Kleinberg, J. The link-prediction problem for social networks. *Journal of the American society for information science and technology* **58**, 1019-1031 (2007).
14 Monod, J., Wyman, J. & Changeux, J.-P. On the nature of allosteric transitions: a plausible model. *J Mol Biol* **12**, 88-118 (1965).
15 Puthenkalam, R. *et al.* Structural studies of GABAA receptor binding sites: which experimental structure tells us what? *Frontiers in molecular neuroscience* **9**, 44 (2016).
16 Thal, D. M., Glukhova, A., Sexton, P. M. & Christopoulos, A. Structural insights into G-protein-coupled receptor allostery. *Nature* **559**, 45 (2018).
17 Ludlow, R. F., Verdonk, M. L., Saini, H. K., Tickle, I. J. & Jhoti, H. Detection of secondary binding sites in proteins using fragment screening. *Proceedings of the National Academy of Sciences* **112**, 15910-15915 (2015).
18 Gunasekaran, K., Ma, B. & Nussinov, R. Is allostery an intrinsic property of all dynamic proteins? *Proteins: Structure, Function, and Bioinformatics* **57**, 433-443 (2004).
19 Pottel, J., Levit, A., Korczynska, M., Fischer, M. & Shoichet, B. K. The Recognition of Unrelated Ligands by Identical Proteins. *Acs Chem Biol* **13**, 2522-2533, doi:10.1021/acschembio.8b00443 (2018).
20 Eckert, H. & Bajorath, J. Molecular similarity analysis in virtual screening: foundations, limitations and novel approaches. *Drug discovery today* **12**, 225-233 (2007).
21 Wallach, I. & Heifets, A. Most ligand-based classification benchmarks reward memorization rather than generalization. *Journal of chemical information and modeling* **58**, 916-932 (2018).
22 Sieg, J., Flachsenberg, F. & Rarey, M. In need of bias control: Evaluating chemical data for machine learning in structure-based virtual screening. *Journal of chemical information and modeling* **59**, 947-961 (2019).
23 Chen, L. *et al.* Hidden Bias in the DUD-E Dataset Leads to Misleading Performance of Deep Learning in Structure-Based Virtual Screening. (2019).
24 Hattori, K., Wakabayashi, H. & Tamaki, K. Predicting key example compounds in competitors' patent applications using structural information alone. *Journal of chemical information and modeling* **48**, 135-142 (2008).
25 Sheridan, R. P. Time-split cross-validation as a method for estimating the goodness of prospective prediction. *Journal of chemical information and modeling* **53**, 783-790 (2013).
26 Kearnes, S., Goldman, B. & Pande, V. Modeling industrial ADMET data with multitask networks. *arXiv preprint arXiv:1606.08793* (2016).
27 Ramsundar, B. *et al.* Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072* (2015).

28 Bailey, D. H., Borwein, J., Lopez de Prado, M. & Zhu, Q. J. Pseudo-mathematics and financial charlatanism: The effects of backtest overfitting on out-of-sample performance. *Notices of the American Mathematical Society* **61**, 458-471 (2014).

29 Montgomery, D. C. *Design and analysis of experiments*. (John wiley & sons, 2017).

30 Calinski, T. & Kageyama, S. *Block Designs: A Randomization Approach*. Vol. I: Analysis (Springer New York, 2000).

31 Rohrer, S. G. & Baumann, K. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *Journal of chemical information and modeling* **49**, 169-184 (2009).

32 Wong, T.-T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition* **48**, 2839-2846 (2015).

33 Landrum, G. (2006).

34 Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of machine learning research* **12**, 2825-2830 (2011).

35 McCallum, A. & Nigam, K. in *AAAI-98 workshop on learning for text categorization.* 41-48 (Citeseer).

36 Bickel, P. J. & Freedman, D. A. Some asymptotic theory for the bootstrap. *The annals of statistics* **9**, 1196-1217 (1981).

37 Prabhu, Y. & Varma, M. in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* 263-272 (ACM).

38 Babbar, R. & Schölkopf, B. in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* 721-729 (ACM).

39 Bhatia, K., Jain, H., Kar, P., Varma, M. & Jain, P. in *Advances in neural information processing systems.* 730-738.

40 Tan, Q., Yu, Y., Yu, G. & Wang, J. Semi-supervised multi-label classification using incomplete label information. *Neurocomputing* **260**, 192-202 (2017).

41 Lewis, D. D. in *European conference on machine learning.* 4-15 (Springer).

42 Keiser, M. J. *et al.* Predicting new molecular targets for known drugs. *Nature* **462**, 175 (2009).

43 Tsoumakas, G. & Katakis, I. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)* **3**, 1-13 (2007).

44 Jones, E., Oliphant, T. & Peterson, P. SciPy: Open source scientific tools for Python. (2001).