Efficient Multi-Task chemogenomics for drug specificity prediction

Benoit Playe,[*,†,¶] Chloé-Agathe Azencott,[†,¶] and Véronique Stoven[†,¶]

†*Mines ParisTech, PSL Research University, Centre for Computational Biology, 35 Rue Saint-Honoré, F-77305 Fontainebleau Cedex, France*

‡*Institut Curie F-75248 Paris, France*

E-mail: benoit.playe@mines-paristech.fr

Abstract

Adverse drug reactions (also called side effects) often interrupt the long and costly process of drug development. Side effects occur when drugs bind to proteins other than their intended target. As experimentally testing drug specificity against the entire proteome is out of reach, we investigate in this paper the application of chemogenomics approaches.

We formulate the study of drug specificity as a proteome-wide drug-protein interaction prediction problem, and build appropriate data sets on which we evaluate multi-task support vector machines.

Our observations lead us to propose NNMT, an efficient Multi-Task SVM for chemogenomics that is trained on a limited number of data points. Finally, we demonstrate the suitability of NNMT to study the specificity of drug-like molecules in real-life situations by suggesting secondary targets for 36 recently withdrawn drugs. In 9 cases, we identified secondary targets responsible for the withdrawal of the drug.

1 Introduction

1.1 Drug specificity

The current paradigm in rationalized drug design is to identify a small molecular compound that binds to a protein involved in the development of a disease, so as to alter the progression of that disease. Once such a hit ligand has been identified, often by combining *in silico* and *in vitro* approaches, this molecule needs to be optimized in order to meet the ADME (Absorption, Distribution, Metabolism, Elimination), toxicity, and industrial synthesis requirements. Finally, pre-clinical and clinical assays are organized to obtain agreement from the regulatory agencies. When successful, this process often lasts more than ten years, and recent estimates set the cost of drug development in US$2.5 billion in 2013.[1]

This complex, long, and costly process is often interrupted because of adverse drug reactions (ADR, also called side effects) that appear at various stages of drug development, or even after the drug has reached the market. In the USA, ADRs have been estimated to have an annual direct hospital cost of US$1.56 billion.[2] A meta-analysis estimated that, in hospitalized patients, the incidence of severe ADR was $1.9\% - 2.3\%$, while the incidence of fatal ADR was $0.13\% - 0.26\%$. For the year 1994, this means that 2 216 000 patients hospitalized in the US suffered from a serious ADR, and approximately 106 000 died.[3] Finally, a recent review[4] found that between 1950 and 2014, 462 medicinal products were withdrawn from the market in at least one country due to ADR. Of these 462 withdrawn drugs, 114 were associated with deaths.

Side effects frequently occur when drugs lack specificity, which means that they bind to proteins other than their intended target.[5] Therefore, the molecular mechanisms at the source of the therapeutic effect and of the unwanted side effects are of similar nature. However, the complete study of drug specificity is at the moment out of reach of the tools that are currently used to identify and optimize a molecule against a given therapeutic target. Indeed, it requires to evaluate potential interactions between the hit molecule and the entire human

proteome, which would be experimentally very costly. Therefore, there is a strong incentive to develop *in silico* methods that predict specificity early in the drug development process. The goal is to reduce the number of experiments to be performed, identify drug candidates that should be dropped because of their lack of specificity, protect patients from deleterious ADRs, and reduce the expense of time and money for the pharmaceutical industry. In this context, the present work studies the large-scale prediction of protein-ligand interactions.

1.2 Protein-ligand interactions prediction

The study of a drug's specificity mainly boils down to predicting its protein targets in the space of the human proteome, or at least at the scale of "druggable" human proteins, that is to say those that present pockets into which drugs can bind. Various approaches have been developed to predict interactions between a protein and a small molecule. Globally, they can be separated into three categories.

The first one gathers so-called ligand-based approaches such as Quantitative Structure Activity Relationship (QSAR) methods.[6–8] They build a model function that predicts the affinity of any molecule for a given target, based on the affinities of known ligands for this same target. They are efficient to study the affinity of molecules against a given protein target, but they are not suitable to study the specificity of a given molecule against a large panel of proteins. This would indeed require, for each of the considered proteins, that the binding affinities of multiple ligands were available.

The second category is docking,[9–11] also called target-based approaches. Docking is a molecular modeling method that predicts the affinity of a ligand for a protein based on the estimated interaction energy between the two partners. However, it relies on the 3D structure of the proteins, which strongly limits its application on a large scale.

The third category includes a diversity of methods referred to as chemogenomic approaches.[12] They can be viewed as an attempt to fill a large binary interaction matrix where rows are molecules and columns are proteins, partially filled with the known protein-ligand

interaction data available in public databases such as the PubChem database at NCBI.[13] They mainly belong to supervised machine learning (ML) methods, which learn mathematical models from available data, and use these models to make predictions on unknown data. In this context, drug specificity prediction can be formulated as a classification task, where the goal is to separate protein-ligand pairs between those that bind together and those that do not.

Various chemogenomics methods have been proposed in the last decade.[14–22] They all rely on the assumption that "similar" proteins are expected to bind "similar" ligands. These methods can be further divided into two categories, known as Single-Task (ST) and Multi-Task (MT) in the ML literature.[23] In ST methods, one task is the prediction of ligands for a given protein, or the prediction of protein targets for a given molecule. Alternatively, in the MT setting,[12] the prediction for a given protein-ligand interaction is made given a set of known protein-ligand interactions that do, or do not, involve the protein or the ligand of the tested interaction. This is of particular interest when the data available for each task are scarce, as when few interactions are known for a given ligand.

A number of MT chemogenomics methods have been proposed, depending on the ML algorithm that is used to learn the model and make the predictions (for example, Support Vector Machine[15] or Random Forest[24]), on the descriptors used to encode proteins and ligands, and on how similarities between these objects are measured. In most cases, they have been implemented to predict interactions of molecules with a set of proteins belonging to a given family such as kinases or GPCRs.[14,25,26] A few studies have been devoted to larger scales in the protein space, such as[22] which however does not focus on settings relevant to the prediction of drug specificity. Some rely on the 3D structure of the binding pocket,[27,28] which limits the number of proteins that can be considered, others on coarse protein descriptors based on the presence of structural or functional domains.[29]

1.3 Contribution

The goal of this paper is to investigate the application of chemogenomics approaches based on Multi-Task SVMs to the prediction of drug specificity on the entire druggable proteome. To this end, we evaluate Multi-Task SVMs in several key scenarios, to better understand their applicability to orphan settings, and to study the impact on performance of the similarity between the query (protein, molecule) pair and the training data. We also discuss how to generate negative training examples, and we optimize protein and molecule kernels in the spaces of drug-like molecules and druggable proteins.

Our observations lead us to propose *NN-MT*, a Multi-Task SVM for chemogenomics that is trained on a limited number of data points: for a query (protein, molecule) pair (p^*, m^*), the training data is composed of (1) all *intra-task* (protein, ligand) pairs, that is to say, pairs (p, m) with either $p = p^*$ or $m = m^*$; (2) a limited number of *extra-task* (protein, ligand) pairs, that is to say, pairs (p, m) with $p \neq p^*$ and $m \neq m^*$, chosen based on the similarity of p and m to p^* and m^*; and (3) randomly picked negative examples (about ten times more than positive training pairs). While the applicability of Multi-Task approaches can be limited in practice by computational times, our approach only requires training on a dataset of size similar to those used by single-task methods. We demonstrate the suitability of *NN-MT* to study the specificity of drug-like molecules in real-life situations within a drug development pipeline by applying it to suggest secondary targets for 36 recently withdrawn drugs. In 9 cases, we were able to link the suggested secondary targets to the side effect responsible for the withdrawal of the drug.

Finally, the protein-ligand interaction data sets we built are made available at `https://github.com/bplaye/efficient\_MultiTask\_SVM\_for\_chemogenomics/data/datasets/`. The code for our experiments can be found at `https://github.com/bplaye/efficient\_MultiTask\_SVM\_for\_chemogenomics/`.

2 Materials and methods

2.1 Single-Task and Multi-Task algorithms

We consider the problem of predicting interactions between small molecules and the druggable proteome. Our goal is to build a model that predicts whether a new pair (p, m) involving protein p and molecule m is likely to interact. As previously mentioned, Machine Learning approaches provide efficient frameworks to address this problem.

In the context of predicting the specificity of a given molecule m, the principle of a *Single-Task* method is to learn a model function $f_m(p)$ that predicts whether or not molecule m interacts with protein p, based on known protein targets of m. In this setting, a new model function needs to be learned for each molecule m, and one task refers to predicting protein targets for a given molecule. We refer to this setting as *ligand-based ST*. Conversely, a Single-Task method could learn a model function $f_p(m)$ that predicts whether or not protein p interacts with molecule m, based on all ligands known for protein p. We call this the *target-based ST* setting.

When predicting interactions between multiple molecules and multiple targets, it may be more effective to consider the so-called *Multi-Task* methods that train a model function $f(m, p)$ that predicts whether m and p interact based on all known interactions between in set of molecules and a set of proteins. The term *Multi-Task* refers to the fact that the function $f(m, p)$ that predicts whether molecule m interacts with protein p is trained based on not only intra-task examples that would be used in Single-Task approaches (i.e. known protein targets for molecule m and known ligands for protein p), but also based on extra-tasks examples (i.e. any other known interactions). We chose to use kernel methods to build the model function $f(m, p)$ because, as detailed in the next section, they are well-suited to Multi-Task problems and have been successfully used in the past to predict interactions

between proteins and ligands.

2.2 Kernel-based classifiers

We considered two types of kernel-based classifiers in this study: Support Vector Machines (SVM)[30] and kernel ridge regression.[31] SVM is a well-known algorithm that has been applied successfully to many biological problems, including the prediction of protein-ligand interactions.[12,15,26,32–37] Kernel Ridge Regression was first used by Yamanishi et al.[14] for drug-target interaction prediction. All work directly inspired from their pioneering work use this algorithm as a classifier.[16–20,38] In the Supporting Information, we briefly recall the basic principles of these methods.

Many kernels have been proposed for molecules and for proteins, based on various representation vectors $\phi(m)$ for molecules and $\phi(p)$ for proteins and used in Single-Task approaches. In the Multi-Task setting, a kernel needs to be defined between (p, m) pairs. The natural way to build this kernel is to use the Kronecker product of the molecule and protein kernels:[39] Given a molecule kernel $K_{molecule}$ and a protein kernel $K_{protein}$, the Kronecker kernel between two protein-molecule pairs is defined by $K_{pair}((m,p),(m',p')) = K_{molecule}(m,m').K_{protein}(p,p')$. The Kronecker product allows to capture interactions between features of the molecule and features of the protein that govern the interaction between m and p.

In what follows, we review the protein and molecule kernels that we considered for our study.

2.3 Protein kernels

We used sequence-based kernels since they can be used proteome-wide, unlike kernels relying on the 3D structure of the proteins or on binding pocket descriptions. Numerous studies have already been devoted defining relevant descriptor of proteins based on amino-acid se-

quence.[40–44] We considered three sequence-based kernels: the *Profile kernel*, the *SWkernel*, and the *LAkernel*.

The *Profile kernel*, presented in Kuang et al.[44], uses as protein descriptors the set of all possible subsequences of amino acids of a fixed length k, and considers their position-dependent probabilistic mutation neighborhoods computed through iterative heuristic alignment with all recorded proteins. This kernel is available at `http://cbio.mskcc.org/leslielab/software/string_kernels.html`.

We also used two kernels that rely on local alignment scores. The first one is directly based on the the Smith-Waterman (SW) alignment score between two proteins[45] and is called the *SWkernel* in the present paper. SW scores were calculated with the EMBOSS Water tool available at `http://www.ebi.ac.uk/Tools/psa/emboss_water/`. We built a kernel based on the SW score matrix by subtracting its most negative eigenvalue from all diagonal values. We also used the Local Alignment kernel (*LAkernel*) proposed by Saigo et al.[43], which mimics the behavior of the SW score. However, while the SW score only keeps the contribution of the best local alignment between two sequences to quantify their similarity, the LAkernel sums up the contributions of all possible local alignments, which proved to be efficient for detecting remote homology.[43] This kernel is available at `http://members.cbio.mines-paristech.fr/~jvert/software/`.

Kernel hyperparameters values. The Profile kernel has two hyperparameters: the size k of the amino acid subsequences that are searched and compared, and the threshold t used to define the probabilistic mutation neighborhoods. We considered $k \in \{4, 5, 6, 7\}$ and $t \in \{6, 7.5, 9, 10.5\}$. The SWkernel also has two hyperparameters: the penalties for opening a gap (o) and for extending a gap (e). We considered $o \in \{1, 10, 50, 100\}$ and $e \in \{0.01, 0.1, 0.5, 1, 10\}$. The LAkernel has three hyperparameters: the penalties for opening (o) and extending (e) a gap, and the β parameter which controls the importance of the contribution of non-optimal local alignments in the final score. We considered $o \in$

$\{1, 20, 50, 100\}, e \in \{0.01, 0.1, 1, 10\}$, and $\beta \in \{0.01, 0.5, 0.05, 0.1, 1\}$. All kernels hyperparameters were optimized by cross-validation (see Section 2.5).

In the last part of the study, we alternatively considered kernels on proteins based on their family hierarchy. Indeed, most of the classes of drug targets, have been organized into hierarchies which typically describe the functions of the proteins within each family (GPCR,[46] kinases[47] and ion channels[48]). As in Jacob and Vert[15], we define the hierarchy kernel between two targets of the same family as the number of common ancestors in the corresponding hierarchy. More precisely, $K_{hierarchy}(t, t') = \langle \phi(t), \phi(t') \rangle$, where $\phi(t)$ is a binary vector for which each entry corresponds to a node in the hierarchy and is set to 1 if the corresponding node is part of t's hierarchy and 0 otherwise.

All protein kernels were centered and normalized.

2.4 Small molecule kernels

Many descriptors have been proposed for molecules, based on physico-chemical and structural properties.[49–52] To measure the similarity between molecules, we considered two state-of-the-art kernels based on molecular graphs that represent the 2D structure of the molecules, with atoms as vertices and covalent bonds as edges. Both kernels compute similarities between molecules via the comparison of linear fragments found in their molecular graphs. They are available at `http://chemcpp.sourceforge.net/`.

The first one, called the Marginalized kernel,[51] considers infinite sets of random walks over the molecular graphs and prevents the random walks from going backwards. This kernel uses the Morgan Index to increase the specificity of the atom labels, resulting in shorter computational times. The similarity value between two molecules is computed via the inner product of the count vectors averaged over a set of linear fragments.

The second kernel, called the Tanimoto kernel, considers a finite set of linear fragments

of a given length, and has a faster computational speed than the former. These linear fragments are labeled paths built directly from the graph by *depth-first search*, which yields to a list of all the paths of a specific length emanating from one starting node. The vectors describing molecules, whose elements count linear fragments within the molecular graphs, called fingerprints, are compared via the Tanimoto metric.[49]

Kernel hyperparameters values. The Tanimoto kernel has one hyperparameter: the length d of the paths, which we considered in $\{2, 4, 6, 8, 10, 12, 14\}$. The Marginalized kernel has two hyperparameters: the stopping probability (while building a path) q in $\{0.01, 0.05, 0.1, 0.5\}$, and the Morgan Index (MI) in $\{2, 3, 4\}$. For both kernels, hyperparameters were selected by cross-validation (see Section 2.5). All molecule kernels were centered and normalized.

2.5 Evaluation of prediction performance

Prediction performance evaluation is commonly performed via a cross-validation (CV) scheme,[53] which can be split into the following two steps: 1) the dataset is randomly split into K folds 2) the model is run K-times, each run using K-1 folds as the training set and testing the performance on the remaining fold which is the test set. Prediction performance are averaged over all folds. When the model contains hyperparameters that need to be tuned, the CV scheme is repeated for all considered values of the hyperparameters, and those leading to the best prediction performance averaged on all folds are retained. We used K=5, a classical value in CV. We also considered the so-called leave-one-out cross-validation (*LOO CV*) scheme, for which the number of folds is the number of available points in the dataset: the model is trained K times on K-1 points, and tested on the left out point. The LOO CV scheme is particularly convenient when the number of sample is especially small.

The goal of the cross validation scheme is to assess the performance of the model on "new" data which were not used to train or optimize the model. From that perspective,

the CV scheme is not rigorous when optimizing hyperparameters. Indeed, the test data are both used to assess the performance and choosing the set of parameters which led to that best performance. To avoid selecting a possibly over-fitted set of parameters, we also used the so-called nested cross validation (*Nested CV*) scheme which consists in a cross validation (*inner-CV*) nested in a other cross validation (*outer-CV*). At each step of the *outer-CV*, the optimal parameters are found via the *inner-CV* on the train set of the *outer-CV*, and the performance is assessed on the remaining test fold of the outer-CV. Therefore, in *Nested CV*, parameter optimization and performance assessment are performed on different "unseen" data.

We estimated prediction performance using two scores that are classically employed to judge the quality of a classifier in case of drug-target interaction prediction. The first one is the area under the Receiver Operating Characteristic curve,[54] usually referred as the AUC under the ROC curve (ROC-AUC) which is the curve of true positive rate as a function of false positive rate. The intuitive interpretation of the ROC-AUC score is that it represents the probability that if a positive and a negative interaction are each picked at random from the dataset, the positive one will be associated to a higher positive score than the negative one by the classifier. The second one is the area under the Precision-Recall curve,[55] called the AUPR score. The intuitive interpretation is that the AUPR score assesses on average how well the prediction score of true positive interactions is far from predicted scores of false positive interactions. Although we used both the ROC-AUC and AUPR scores, since negative interactions are actually unknown interactions in protein-ligand interaction datasets, the AUPR is considered a more significant quality measure of the prediction method than the ROC-AUC. Indeed, it emphasizes the recovery of the positive samples and penalizes the presence of false positive examples among the best ranked points.

We used the Python library scikit-learn[56] to implement all considered machine learning

algorithms.

2.6 Datasets

Many publicly available databases such as KEGG Drug,[57] DrugBank,[58] or ChEMBL[59] can be used to build a learning dataset of protein-ligand interactions. We chose to build the datasets used in the present study from the DrugBank database v4.3, because it contains information about FDA-approved drugs, or drug candidate molecules studied in various development programs. Drugs occupy only specific regions of the chemical space, since they need to fulfill some requirements. Basically, they must be able to be distributed into the human organism, to interact with human proteins, and they must satisfy security levels like those specified by the FDA. Therefore, it appeared important to train and test our models on molecules similar to those on which they would be applied. In addition, we assumed that the list of human proteins appearing as targets for molecules of DrugBank could represent a "druggable" proteome on which to train models aiming at predicting the specificity of drug-like molecules.

We built a first learning dataset called S, based on Version 4.5 of the DrugBank.[58] We selected all molecules targeting at least one human protein, and having a molecular weight between 100 and 600 $g.mol^{-1}$, a range in which most small molecule marketed drugs are found.[60] This leads to a dataset composed of 3 980 molecules targeting 1 821 proteins, and including 9 536 protein-ligand interactions that correspond to the positive pairs. All other protein-ligand pairs are unlabeled because no interactions were recorded for them in the database. Most of these pairs are expected not to interact, but a small number of them are in fact missing interactions. However, we considered that all unlabeled pairs can be considered negative examples, and the predictions will re-classify some of these pairs as positive examples.

In order to compare the prediction performance of the considered methods on datasets covering a panel of situations that are relevant in the context of large scale prediction in the protein space, we used S to build a set S_0 by extracting all positive pairs in which the protein and the ligand was present in at least two couples, leading to 5 908 positive interactions. We randomly generated four sets of 5 908 negative interactions involving proteins and ligands found in the positive interactions, while ensuring that each protein and each ligand are present in the same number of positive and negative interactions. We assessed performance by computing the mean and standard deviation of the prediction scores over test sets including the positive interactions set and one of the negative interactions sets.

Starting from the S dataset, we also built three datasets that gathered proteins belonging to three families, in order to test the performance of the proposed methods when making prediction within a family of proteins. We extracted all protein-ligand interactions of S_0 that only involved G-Protein Coupled Receptors, ion channels, and kinases respectively and built three sub-datasets accordingly called GPCR, IC and Kinases.

Table 1 gives some statistics about the datasets built and used in the present study, including the distribution of targets per drug and the distribution of ligands per protein.

Table 1: Dataset statistics

	S	S_0	GPCR	IC	Kinases
number of interactions	9 536	5 908	1 735	1 603	847
number of proteins	1 821	788	85	140	143
number of molecules	3 980	1 180	482	295	577
number of targets per drug (mean/median)	5.2/2	7.5/3	20.3/6	5.9/3	11.5/4
number of targets per drug (min – max)	$1-136$	$2-82$	$1-86$	$1-67$	$1-136$
number of ligands per protein (mean/median)	2.4/1	5.0/3	3.6/3	5.4/3	1.5/7
number of ligands per protein (min – max)	$1-70$	$2-48$	$1-31$	$1-26$	$1-18$

3 Results

3.1 Kernel selection and parametrization

The goal of this section is to chose a protein kernel and a molecule kernel that we will use throughout the remainder of this study. We assumed that kernels optimized on a large dataset of interactions between drug-like molecules and druggable human protein would be good default kernels for the prediction of the specificity of drug candidates. We therefore optimized kernels on dataset S, and used the best-performing kernels in the remainder of the paper. Since the goal of this paper is to investigate Multi-Task chemogenomics approaches, we looked for the best-performing Kronecker combination of protein and molecule kernels. The best protein (resp. molecule) kernel we obtained might not be the one that performs best in a *target-based ST* (resp. *ligand-based ST*) approach.

The set of (protein, ligand) pairs in S were randomly 5-folded, and we performed a Nested-5-fold cross validation experiment in order to evaluate the six possible kernel combinations and their best hyperparameters.

Table 2: Best nested 5-fold cross-validated AUPR for each kernel combination, together with optimal hyperparameters.

Kernels	Tanimoto	Marginalized
LAkernel	AUPR $= 0.938 \pm 0.001$ Tanimoto: $d = 14$ LAkernel: $o = 20$, $e = 1$, $\beta = 1$	AUPR $= 0.930 \pm 0.001$ Marginalized: $q = 0.1$, MI=4 LAkernel: $o = 20$, $e = 1$, $\beta = 1$
SWkernel	AUPR $= 0.878 \pm 0.002$ Tanimoto: $d = 6$ SWkernel: $o = 100$, $e = 0.01$	AUPR $= 0.868 \pm 0.002$ Marginalized: $q = 0.1$, MI=2 SWkernel: $o = 50$, $e = 10$
Profile	AUPR $= 0.941 \pm 0.001$ Tanimoto: $d = 8$ Profile: $k = 5$, $t = 7.5$	AUPR $= 0.935 \pm 0.001$ Marginalized: $q = 0.1$, MI=2 Profile: $k = 4$, $t = 6$

Table 2 gives the best prediction performance for the six combinations of protein and molecule kernels, together with the corresponding hyper-parameters. All protein kernels

gave the best AUPR when coupled to the Tanimoto kernel. The Marginalized kernel obtained good performance only when coupled to the Profile kernel. Overall, the Profile kernel ($k = 5, t = 7.5$) associated to the Tanimoto kernel ($d = 8$) gave the best results. Therefore, in what follows, we only consider these two kernels, and call MT the Multi-Task SVM that uses their Kronecker product.

It is worth noting that the SWkernel gave the worst performance, although it is used in many studies.[14,17,20,38] The good performance of multi-task methods we observed on S is consistent with previously reported results.[22,38] However, S was built from the DrugBank, which is mostly fueled by application-specific screens of either related proteins or related small molecules. Therefore, (protein, ligand) pairs of the test sets will usually have close pairs in the train set (i.e. pairs involving the same or similar proteins and ligands), which will facilitate the prediction. The performance we observed are therefore likely to be overinflated compared to what can be expected in real-case prediction of drug specificity, particularly regarding new drugs and therapeutic targets, as already introduced in Pahikkala et al.[25]. The question of interest is now to which extent the proposed MT method is effective to make predictions on more challenging situations that are relevant in the context of drug specificity prediction. In what follows, we will therefore study the evolution of MT's performance in more realistic settings.

3.2 Performance of multi-task approaches in orphan situations

The goal of this section is to evaluate the performance of MT in cases where the queried (protein, molecule) pairs pertain to proteins and/or molecules that are not in the training set, as proposed by Pahikkala et al.[25]. For that purpose, we created four cross-validation data sets, which contain the pairs of S_0 5-folded as follows:

- S_1: random, balanced cross-validation;

- S_2 ("orphan ligand"): (protein, molecule) pairs in one fold only contain molecules that are absent from all other folds;

- S_3 ("orphan protein"): (protein, molecule) pairs in one fold only contain proteins that are absent from all other folds;

- S_4 ("double orphan"): (protein, molecule) pairs in one fold only contain proteins *and* molecules that are both absent from all other folds. The folds of S_4 were built by intersecting those of S_2 and S_3 and S_4 thus contains 25 folds and not 5.

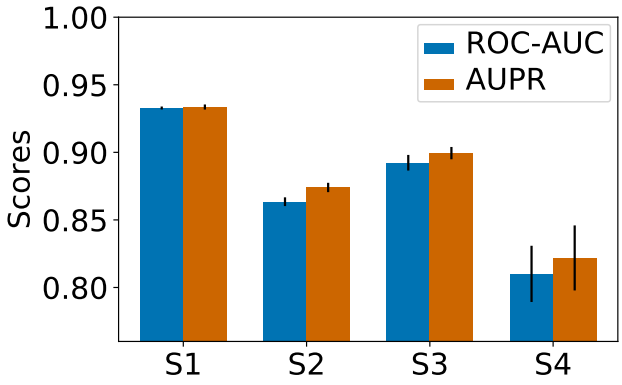


Figure 1: Performance of the MT method on the $S_1 - S_4$ cross-validation datasets. Numerical values can be found in Supporting Information

Figure 1 shows the nested cross-validated AUC and AUPR scores obtained by the MT method on those cross-validation data sets. As expected, the best scores are obtained for S_1, and the worst for S_4, since in S_4 the prediction algorithm cannot leverage information from pairs containing neither the same proteins nor the same molecules as in the test set. The loss of performance between the random and the double orphan settings is about 0.12 both in AUC and AUPR. However, the performance on the S_4 setting remains well above those of a random predictor. These results confirm that MT chemogenomics can make predictions for (protein, ligands) pairs made of unseen proteins and unseen ligands, even in datasets containing very diverse types of proteins. This confirms previous observations on less diverse datasets.[25] It is important to point out that single task approaches would not be able to

provide any prediction in the S_4 setting.

The scores obtained in the S_2 and S_3 settings are intermediate between those observed on S_1 and S_4. This was to be expected, as in these settings the algorithm can rely on training pairs containing either the same proteins (S_2) or the same ligands (S_3) as the test set. The AUC and AUPR scores are both slightly better for S_3 than for S_2, which suggests that specificity is easier to predict for new drug targets than for new targeted compounds, as already noticed in Pahikkala et al.[25]. This may be specific to our dataset, or protein kernels may be better suited than molecule kernels to the prediction of drug-target interactions. The same global behavior was observed with the *5-fold CV* for the $S_1 - S_4$ settings, as shown in Supporting Information.We also observed similar behaviors when replacing the SVM with a kernel ridge regression (see Supporting Information)and hence stopped considering this alternative algorithm in what follows.

All in all, our results suggest that the performance of MT is driven by known (protein, molecule) pairs that are similar to the query pair in the sense that they share either their protein or their molecule. In the next section, we will evaluate how the actual similarity between query and train pairs influences the performance of Multi-Task drug sensitivity prediction algorithms.

3.3 Impact of the similarity of the training examples to the test set

To evaluate the impact on performance of the dissimilarity between training and test pairs, we re-folded the pairs of S_0 following the "clustered cross-validation" approach.[61]

More precisely, we clustered proteins (resp. ligands) into 5 clusters by hierarchical clustering.[62] We used the same kernels as MT, namely the Profile kernel for proteins and the Tanimoto kernel for ligands. We then built four cross-validation datasets, $S'_1 - S'_4$, generated

similarly as $S_1 - S_4$, but with the added constraint that all pairs in a given fold are made of proteins from a single protein cluster and ligands from a single ligand cluster. In this way, test pairs are be more dissimilar from train pairs than when folds are built at random, which might make the problem more difficult.

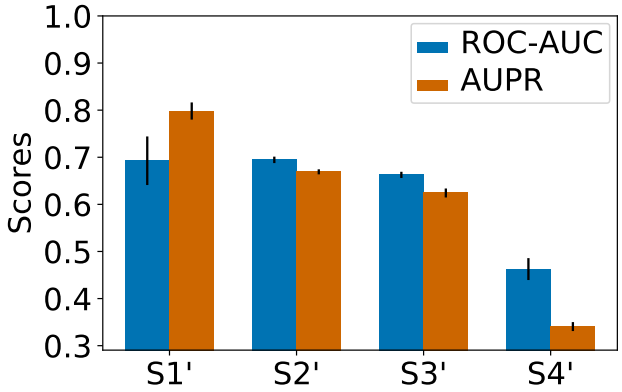


Figure 2: Nested 5-fold cross-validated performance of the MT method on the $S'_1 - S'_4$ datasets. Numerical values can be found in Supporting Information.

Figure 2 shows the prediction performance of MT on these new cross-validation folds. For all the datasets, we observed a strong decrease in prediction scores with respect to those obtained on the corresponding $S_1 - S_4$ settings. The drop in performance is stronger from S_1 to S'_1 than between the two variants of the single-orphan settings, and the ROC-AUC score on S'_1, S'_2 and S'_3 are now quite comparable. However, the AUPR score obtained on S'_1 (0.798 ± 0.018) is significantly larger than on S'_2 (0.669 ± 0.053) or S'_3 (0.624 ± 0.095).

Overall, making predictions on clustered cross-validation data sets ($S'_1 - S'_4$) is more difficult than on a randomly folded double orphan setting (S_4). This suggests that good performance on a query pair (p^*, m^*) is driven by the presence in the training set of pairs made of *both* proteins similar to p^* and molecules similar to m^*. This seems more important than having trained on data containing, for example, p^* itself, but paired only with molecules quite dissimilar to m^*. Finally, performance on S'_4 are close to random, confirming that making predictions for double orphans that are also very dissimilar from the training data is a very difficult task.

These results suggest that the training set pairs that are very dissimilar to the query pair do not help making more accurate predictions. This insight is interesting considering that MT is computationally demanding, as the Kronecker kernel must be evaluated on all (protein, molecule) pairs: can we use it to reduce the number of training samples needed without negatively affecting performance? Furthermore, is it possible to further improve computational efficiency by identifying settings in which Single-Task methods compete with MT in terms of performance? We will endeavor to answer these questions in the next two sections.

3.4 Multi-Task approaches on reduced training sets

Based on the intuition, confirmed in the previous section, that MT prediction is driven by training examples that are close to the query data, rather than by those that are far away, we propose to build training tests of reduced sizes that do not contain such distant training examples. The goal of this section is to compare the prediction performance of the MT method trained on these reduced data sets to that of the simpler and faster Single-Task method. Because this study is motivated by ligand specificity prediction, we chose to focus on comparisons with the *ligand-based ST* method rather than *target-based ST*.

In what follows, when testing a given (protein, ligand) interaction in Multi-Task methods, we will call *intra-task training points* all known interacting pairs involving the same particular protein or the same particular ligand. Similarly, for *ligand-based ST* methods, only interactions involving the query molecule are considered. We refer to these pairs as *intra-task training points*, as defined above in Section 1.3. *Extra-task training points* can only be defined for Multi-Task methods: they correspond to known interacting pairs involving neither the protein nor the ligand of the tested pair. Moreover, n^+ (resp. n^-) will refer to the number of positive (reps. negative) examples in the train set.

In the next section, we used the LOO-CV scheme because intra-task and extra-task pairs can only be defined for each pair separately. In addition, in single task approaches, the size of the training set was relatively small in most cases (see datasets statistics in Section 2), which does not allow to fold the data. We checked that the LOO scheme did not trigger a bias, as sometimes observed[25] (see Supporting Information). Because single task approaches can only be applied to (protein, ligand) pairs involved in at least another interacting pair, we used the S_0 dataset for all following experiments.

3.4.1 Training on intra-task positive examples

The goal of this section is to compare the prediction performance of the *MT* method trained on a reduced data set of size similar to that employed in Single-Task methods. Since the only positive examples *ligand-based ST* can use are intra-task pairs, it is natural when trying to reduce training set size to limit the positive training examples of *MT* to intra-task pairs as well. Note that *MT* still gets more training examples as *ligand-based ST* , as pairs formed with the query protein and a different ligand are also included. In addition, this reduces the training set size, and results in computational times similar to those of the Single-Task method. In the following, we call *MT-intra*this variant of *MT*. For each test ligand, we build the negative training examples by randomly selecting a number n^- of proteins that do not interact with the ligand in S_0. We vary n^- from 1 to $100 \times n^+$.

Figure 3 shows the LOO-CV AUPR of *MT-intra*and *ligand-based ST* on S_0, for increasing values of the n^-/n^+ ratio. For both methods, the AUPR score increases with the number of negative pairs in the train set, before decreasing for large numbers of negative pairs. A good trade-off for both computational and predictive performance seems to be in the range of 10 times more negative points than positive points. We therefore set n^-/n^+ to 10 for the remaining experiments of this section.

The AUPR scores of *MT-intra*outperform those of the *ligand-based ST* method. Interestingly, the performance of the *MT-intra*with a n^-/n^+ ratio of 10 is close to 0.96 which outperforms the AUPR score of 0.93 obtained with *MT* (see Section 3.2). This indicates that including pairs displaying low similarity to the tested pair in the train set degrades both the computational time and the quality of the prediction of *MT*.

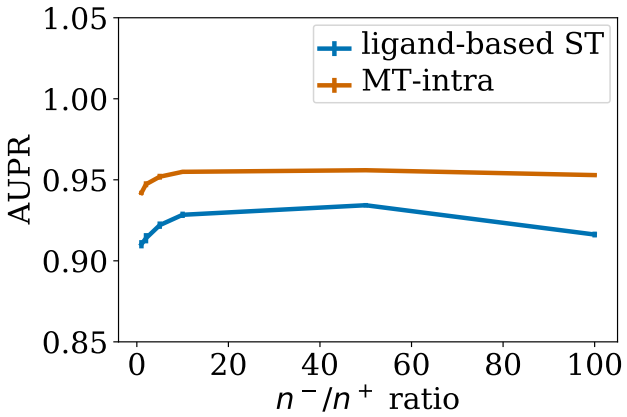


Figure 3: Performance of Single-Task and *MT-intra*as a function of the n^-/n^+ ratio. Numerical values can be found in Supporting Information

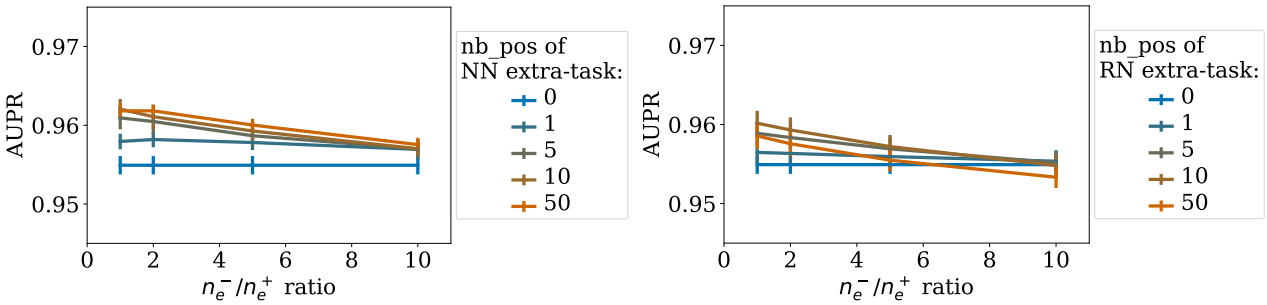
3.4.2 Adding similar extra-task positive examples

The results from Section 3.3 suggest to explore the performance of the *MT-intra*method when trained on various datasets including extra-task pairs close to the tested pair, in addition to the intra-task pairs. Therefore, we built train sets made of:

- the train set of *MT-intra*, that is to say, all the positive intra-task pairs and ten times more randomly chosen negative intra-task pairs (i.e. involving the protein or the ligand of the tested pair).
- n_e^+ closest extra-task positive pairs with respect to the tested pair ($n_e^+ \in \{1, 5, 10, 50\}$).
- n_e^- closest extra-task negative pairs with respect to the tested pair, so that the n_e^-/n_e^+ ratio varies from 1 to 10.

We call *NN-MT* the resulting variant of *MT*. We also considered a similar approach, in which the extra-task pairs were chosen at random rather than according to their similarity to the test pair. We refer to this method as *RN-MT*.

We report the LOO cross-validated performance of *NN-MT* and *RN-MT* on Figure 4. Figure 4a shows that, while adding to the train set 0 to 50 nearest neighbor extra-task positive pairs with respect to the tested pair, the prediction performance of *NN-MT* slightly and monotonously increases. Figure 4b shows that the performance of *RN-MT* also slightly increases (although not monotonously) when random extra-task pairs are added. However, its best performance remains under that of *NN-MT*. Finally, using a high n_e^-/n_e^+ ratio did not improve the performance. This is an interesting observation, since limiting the size of the train set is computationally favorable.



(a) *NN-MT* (b) *RN-MT*

Figure 4: AUPR as a function of the n_e^-/n_e^+ ratio for increasing numbers random extra-task points in the train set. The black horizontal line corresponds to *MT-intra*(which is trained only on intra-task pairs). Numerical values can be found in Supporting Information

Taken together, these results show that the *NN-MT* method not only outperforms *MT-intra*method, but also the more computationally demanding *MT* method trained in the LOO-setting, as in Section 3.2. They also show that the expected AUPR scores for (protein, ligand) pairs involving non-orphan ligands and non-orphan proteins can be expected to be very high (around 0.96).

However, predicting the specificity of a given ligand requires the ability to make predictions for proteins that are far from the known targets. In these cases, the high prediction scores obtained in this section might not hold. Therefore, in the next section, we study the performance of *NN-MT* when the test pairs are far from the train set.

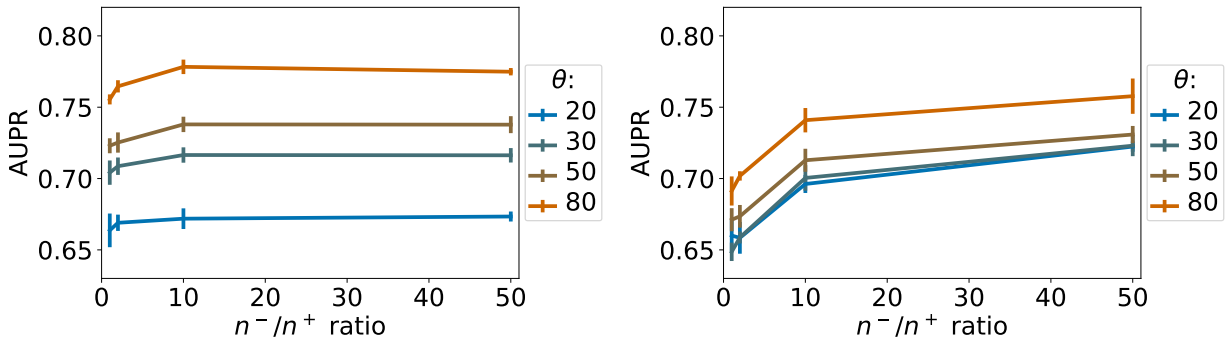
3.5 Impact of the distance of the intra-task examples to the query pair

The goal of this section is to evaluate the performance of the proposed methods when the similarity between the test pair and the training data varies.

3.5.1 Training on dissimilar intra-task positive examples

We first evaluated the performance of *ligand-based ST* and *MT-intra*when the similarity between the test pair and the training data varies. To do so, we computed the percentiles of the molecules (respectively proteins) similarity distribution in S_0. For a test pair (p^*, m^*), we only included in the corresponding training set the positive intra-task pairs (p, m) such that $K_{protein}(p, p^*)$ and $K_{molecule}(m, m^*)$ is lower than a percentile-based threshold θ. We then added n^- random intra-task negative pairs. We did not apply a similarity constraint to negative pairs, as, unlike the positive pairs, they are available in large numbers and at all distances from the tested pairs.

We report on Figure 5 the LOO cross-validated AUPR of *ligand-based ST* and *MT-intra*for varying values of θ ($20^{th}, 30^{th}, 50^{th}$, and 80th percentiles) and of the n^-/n^+ ratio (from 1 to 50).



(a) *MT-intra*method

(b) *ligand-based ST* method

Figure 5: AUPR scores as a function of the n^-/n^+ ratio, for threshold θ ranging from 20% to 80%. Numerical values can be found in Supporting Information

Figure 5a shows that, as expected, the performance of *MT-intra* increases when the similarity of the tested pair to the train set increases from the 20th to the 80th percentiles (AUPR of 0.67 to 0.77). However, the performance is still much lower than when the closest pairs are allowed in the training set (AUPR of 0.96). Figure 5a also suggests that a n^-/n^+ ratio of 10 is again an appropriate choice. We therefore set n^-/n^+ to this value for the remaining experiments of this section.

Figure 5b shows that *ligand-based ST* behaves similarly to *MT-intra*: the AUPR score increases from 0.70 to 0.75 for *ligand-based ST* when threshold θ increases from the 20th to the 80th percentiles. These values again remain much under the AUPR score of 0.93 observed when all intra-task pairs are used. Although modest, the performance of *MT-intra*remains better than that of *ligand-based ST* for all tested thresholds of similarity (except $\theta = 20$) between the tested pair and pairs of the train set.

3.5.2 Adding similar extra-task positive examples

We then explored to which extent adding extra-task (protein, ligand) pairs to the training set of the *MT-intra*method improves the prediction scores. For this purpose, we evaluated *NN-MT* and *RN-MT* using the same percentile-based similarity constraint to the intra-task

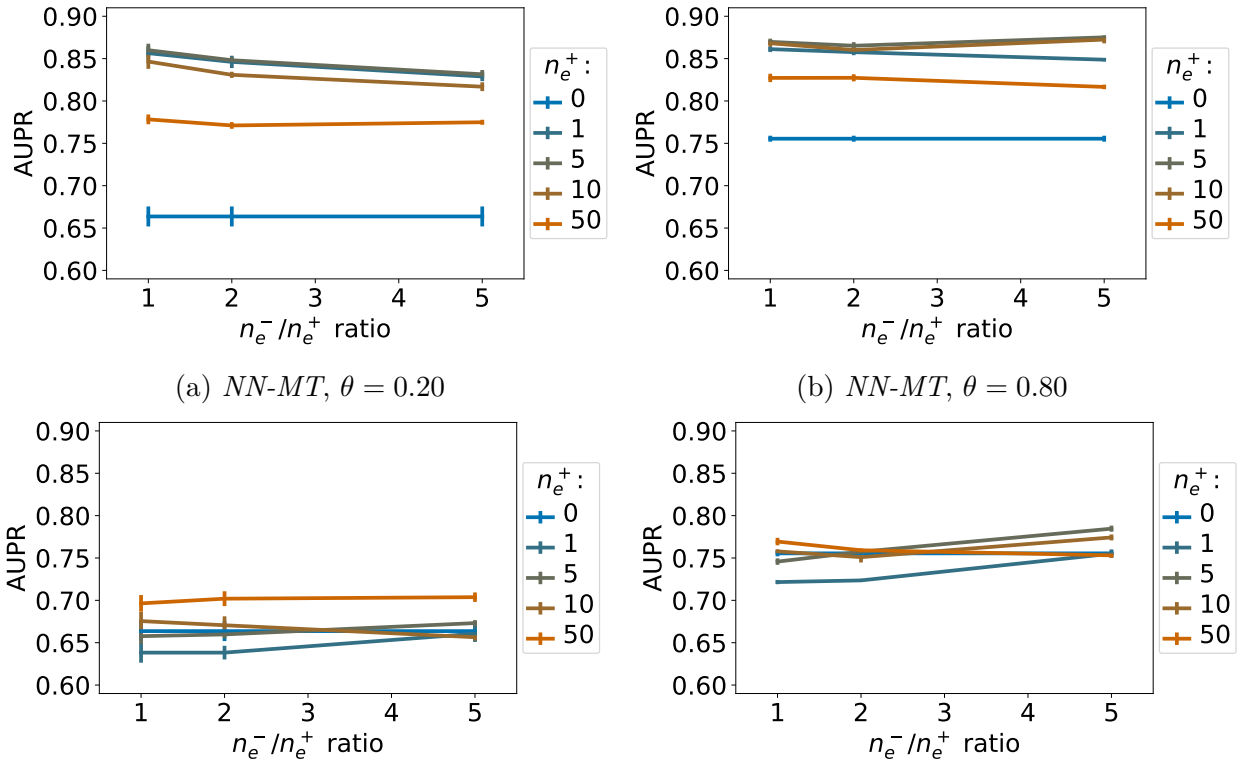
positive pairs of the training set.

We did not apply a similarity constraint to the extra-task pairs, since the principle underlying Multi-Task methods is precisely to learn from extra-task data, which is particularly critical when the intra-task pairs of the train set are far from the tested pair, as illustrated by the poor performance of *ligand-based ST* in the previous section.

We report on Figure 6 the LOO cross-validated AUPR of *MT-intra*for varying values of θ ($20^{th}, 30^{th}, 50^{th}$, and 80th percentiles) and of the n_e^-/n_e^+ ratio (from 1 to 50). Figures 6a and 6b show that adding extra-task pairs to the train set dramatically improves performance. The AUPR score reaches values above 0.85, independently of θ, suggesting than when no close intra-task pairs are available, performance is driven mainly by extra-task training pairs, confirming our observations from Section 3.3. Moreover, when the number of extra-task pairs increases, the performance of *NN-MT* tends towards that of *RN-MT* and degrades when too many dissimilar extra-task pairs are included in the training set. This implies that only a limited number of the closest extra-task pairs is required to reach optimal performance. Unsurprisingly, the best AUPR in the absence of close intra-task pairs (around 0.87 for $\theta = 0.80$) is still lower than when all available intra-task pairs are used (AUPR=0.93, see Section 3.4). Note that, although the performance of *MT-intra* can be biased when considering similarity thresholds of 20th and 80th because the corresponding sizes of the train sets might be different, this is not the case for the *NN-MT* method because the prediction is driven by extra-task pairs.

On the contrary, Figures 6c and 6d show that performance does not improve when the extra-task training pairs are chosen at random and therefore further away from the test pair. It might even degrade when the number of extra-task pairs is large. Finally, Figure 7 shows that for all similarity thresholds the performance of the *MT-intra* and *RN-MT* methods are similar, and far beyond that of the *NN-MT* method.

In conclusion, in settings where (protein, ligand) pairs similar to the test pair are available, our results suggest using *NN-MT* trained with 10 times more negative intra-task pairs than positive ones, 1 to 5 extra-task positive pairs, and the same number of extra-task negative pairs. The computational time will be reasonably comparable to that of *ligand-based ST* , and performance should be high enough (AUPR above 0.85) to guide experimental evaluations for drug specificity prediction.



(a) *NN-MT*, $\theta = 0.20$

(b) *NN-MT*, $\theta = 0.80$

(c) *RN-MT*, $\theta = 0.20$

(d) *RN-MT*, $\theta = 0.80$

Figure 6: AUPR score of *NN-MT* and *RN-MT* as a function of the n_e^-/n_e^+ ratio. Numerical values can be found in Supporting Information

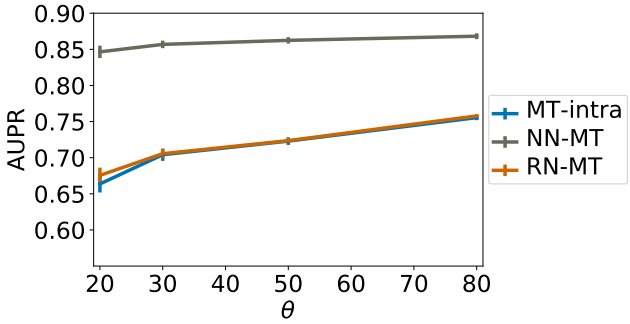
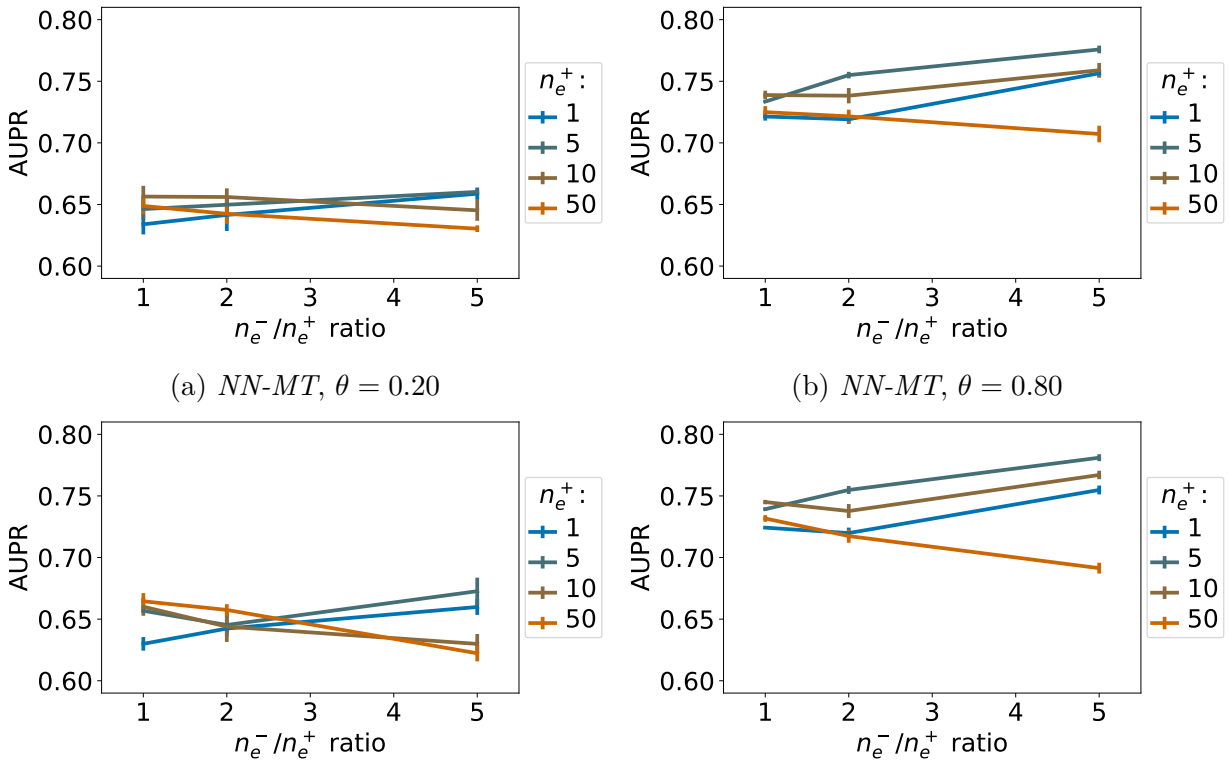


Figure 7: AUPR as a function of percentile-based similarity, for n^-/n^+=10, a number of extra-task positive pairs n_e^+=10 and a ratio of n_e^-/n_e^+=10 for extra-task pairs. Numerical values can be found in Supporting Information

3.5.3 Adding dissimilar extra-task positive examples

While we previously argued that the point of Multi-Task approaches is to leverage similar extra-task data to improve prediction performance, ligand specificity studies can require the prediction of interactions between proteins and ligands for which very little similar extra-task data is available. We therefore repeated the experiments from the previous section, but this time applying the percentile-based similarity constraint to both intra-task and extra-task positive pairs of the train set. We report the corresponding LOO cross-validated AUPR on Figure 8.

We observe that the performance of *NN-MT* remains overall very low (best AUPR score of 0.75 for $\theta = 0.80$). Adding dissimilar extra-task positive pairs fails to improve the scores obtained when only intra-task positive pairs are included in the training set. Hence if no close intra-task nor extra-task positive pairs are available, no method can provide performance good enough for the purpose of drug sensitivity prediction. Such observations were expected given that adding random extra-task training pairs, possibly far from the tested pair, did not improve performance in the previous case Section 3.5.2.



(a) *NN-MT*, $\theta = 0.20$

(b) *NN-MT*, $\theta = 0.80$

(c) *RN-MT*, $\theta = 0.20$

(d) *RN-MT*, $\theta = 0.80$

Figure 8: AUPR scores of the Multi-Task methods trained on dissimilar intra-task and extra-task examples. The black horizontal line corresponds to the *MT-intra*method trained only on intra-task pairs. Exact values can be found in Supporting Information

Taken together, our results show that the *NN-MT* method we proposed is the most appropriate for predicting the specificity of a molecule. Indeed, it outperforms all its comparison partners independently of the number of known (protein, ligand) interacting pairs involving the same or similar ligands or proteins as the query pair. In addition, it requires much fewer training pairs than the classical *MT* approach, and its computational time is therefore close to that of a Single-Task method. Finally, in the most challenging setting where no similar intra-task nor extra-task training data is available, it performs significantly better than random whereas *ligand-based ST* could not make any prediction.

The results we have presented so far pertain to proteome-wide specificity prediction. However, proteins from the same family as the primary target are important secondary target

candidates, and family-wide specificity prediction is an important issue. It corresponds to the setting where similar training pairs are available, as proteins of the same family are similar in terms of sequence (see the clusters obtained when clustering proteins of S with k-means in Supporting Information). In the next section, we assess whether the proposed *NN-MT* method provides good performance for molecule specificity prediction within a family of proteins.

3.6 Specificity prediction within families of proteins

We considered three families of proteins because they gather a wide range of therapeutic targets, and have also been considered in other chemogenomics studies, thus providing reference prediction scores: G-Protein Coupled Receptors (GPCRs), ion channels (IC), and kinases. As presented in Section 2.6, the dataset S was used to build the three corresponding family datasets.

We compared the performance of the *MT-intra*method (trained using only positive pairs involving the protein or the ligand of the tested pair) to those of the *NN-MT* and *RN-MT* methods, in order to evaluate the interest of the Multi-Task approach in family studies. We considered two versions: one in which the Profile protein kernel is used, as in the above sections, and another in which a family-based hierarchy kernel is used (Section 2.3).[15,26] The corresponding methods are called *MT-intra-family*, *NN-MT-family*, and *RN-MT-family*. Indeed, a sequence-based kernel may not be optimal to study the specificity of the molecule within a family of proteins.

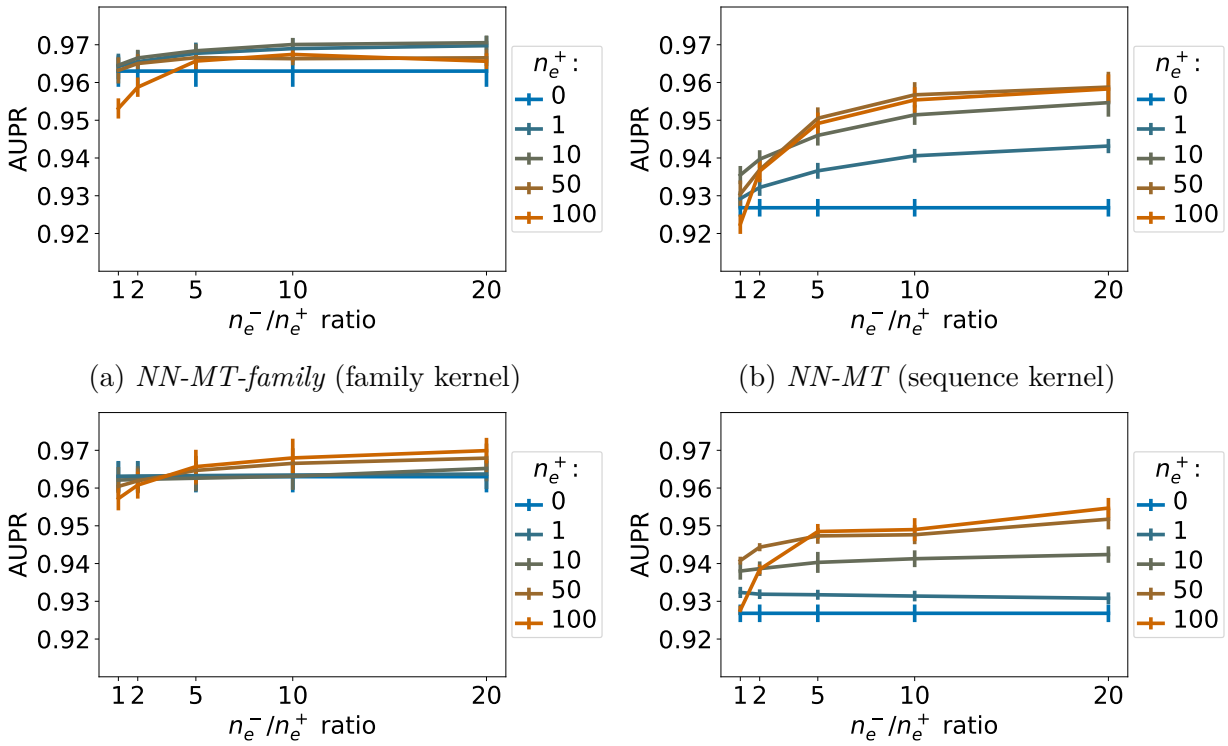
As in the above section, for the two Multi-Task approach, each (protein, ligand) test pair is considered in turn in a LOO scheme, and we used a learning dataset containing: all positive intra-task positive pairs, ten times more random negative intra-task pairs, a varying number of positive extra-task pairs (nearest neighbors for *NN-MT* or *NN-MT-family*, random positive pairs for *RN-MT* or *RN-MT-family*), and a number of negative extra-task pairs so

that the ratio of n_e^-/n_e^+ varies from 1 to 20.

3.6.1 G-Protein Coupled Receptor family

Figures 9a to 9d show that all methods perform very well, with AUPR scores above 0.95. Including extra-task positive pairs in the train set improves the AUPR score, even when added randomly. This indicates that, contrary to studies in larger scales in the protein space, in family studies, extra-task pairs are always close to the tested pair because they belong to the same family. However, the performance reached when adding positive nearest-neighbor extra-task pairs remain above those reached when adding positive random extra-task pairs, as observed in the larger scale studies presented above.

The best AUPR scores of the *NN-MT* and the *NN-MT-family* methods are close (0.96 and 0.97). Although the best scores of the *NN-MT-family* method are slightly above those *NN-MT*, one should note that the family GPCR kernel is based on a GPCR hierarchy that was established using known GPCR ligands. Therefore, the results obtained by the *NN-MT-family* might be biased, which is not the case of those obtained by the *NN-MT*.



(a) *NN-MT-family* (family kernel)

(b) *NN-MT* (sequence kernel)

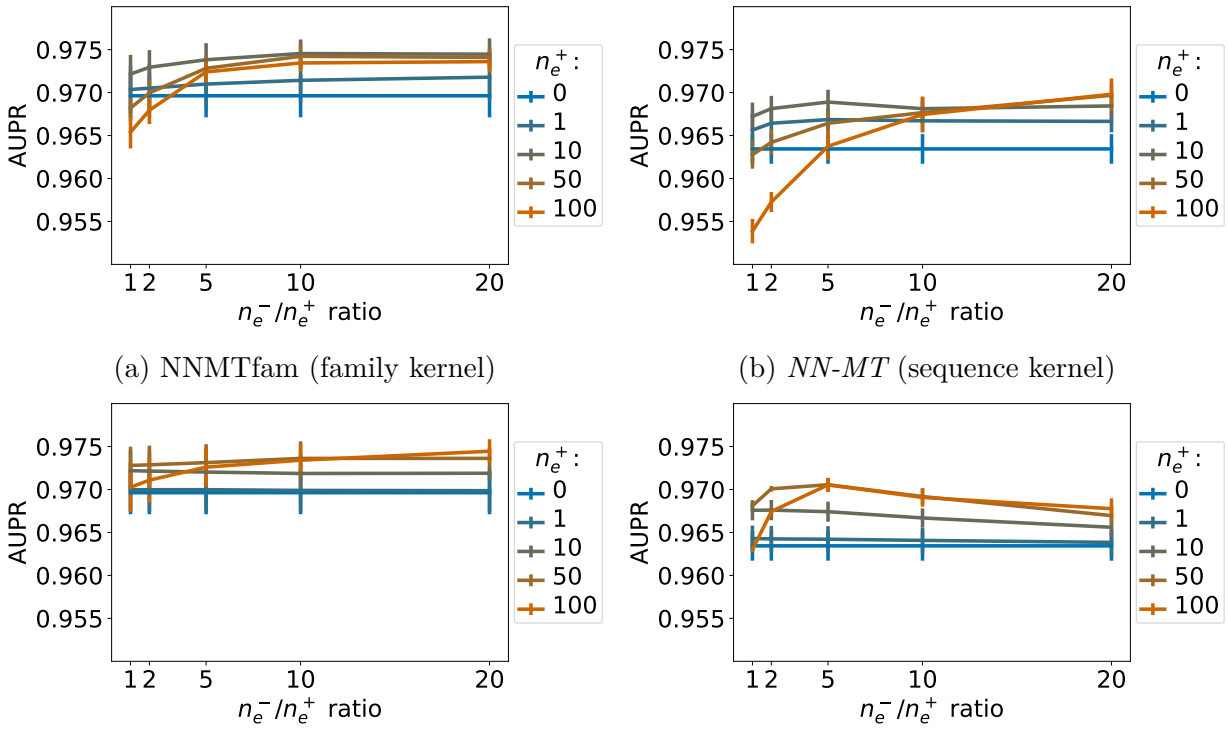
(c) *RN-MT-family*(family kernel)

(d) *RN-MT* (sequence kernel)

Figure 9: AUPR score of the Multi-Task methods on the GPCR family. The black horizontal line corresponds to the *MT-intra*method trained only on intra-task pairs. Numerical values can be found in Supporting Information

3.6.2 Ion Channel family

The conclusions obtained above in the GPCR family also hold in the IC family, as shown in Figures 10a to 10d. Again, all methods perform very well, reaching AUPR scores above 0.97. Adding 10 to 50 extra-task positive pairs to the train set, and around 10 times more random negative extra-task pairs leads to the best performance. These scores are even better when only intra-task pairs are used.



(a) NNMTfam (family kernel)

(b) *NN-MT* (sequence kernel)

(c) *RN-MT-family*(family kernel)

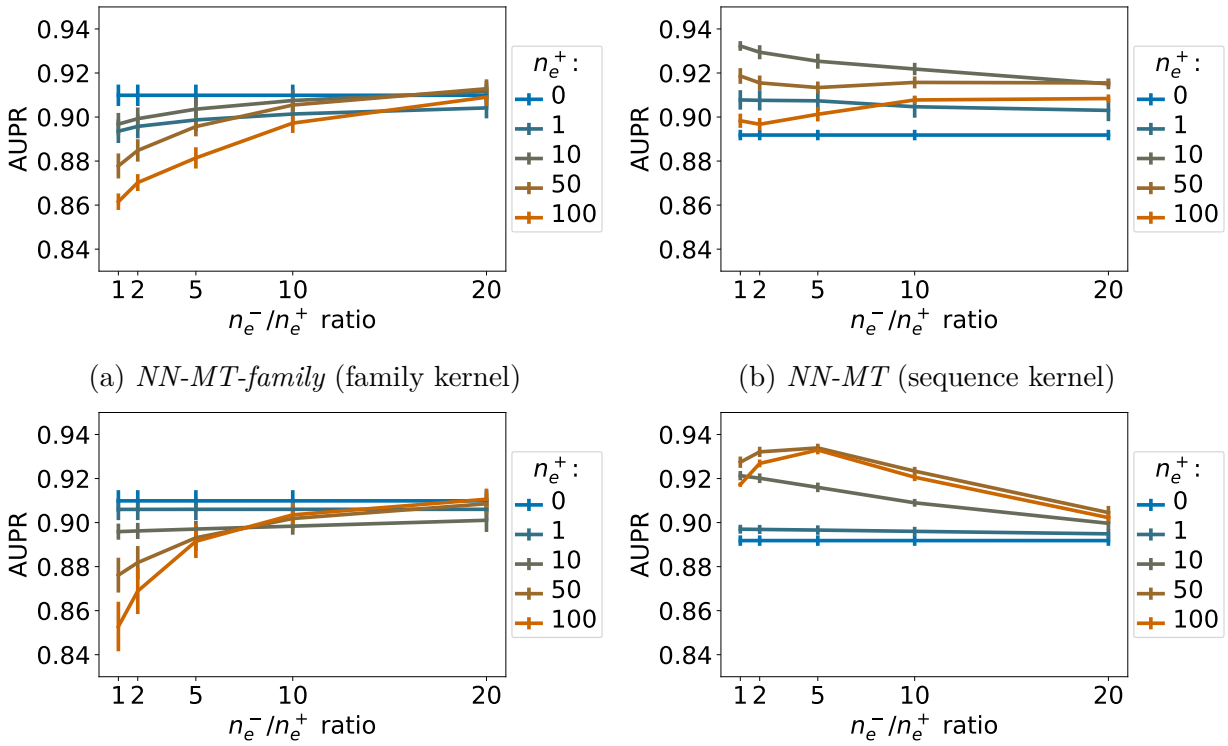
(d) *RN-MT* (sequence kernel)

Figure 10: AUPR of the Multi-Task methods on the IC family. The black horizontal line corresponds to the *MT-intra*method trained only on intra-task pairs. Numerical values can be found in Supporting Information

3.6.3 Kinase family

In the case of the kinase family, the results are somewhat different from those of IC and GPCRs. The *NN-MT* and *RN-MT* methods both outperform the *MT-intra*method that is trained using only intra-task pairs, as shown in Figures 11b and 11d. Again, 10 to 50 extra-task pairs, with a n_e^-/n_e^+ ratio in the range of 1 to 5 leads to the best results, with AUPR scores in the range of 0.93. Unexpectedly, the *NN-MT-family* and *RN-MT-family*methods, which both use the kinase family kernel, tend to perform not as well when extra-task pairs are added to the training set, than when only the intra-task pairs are used, as shown in Figures 11a and 11c. In addition, their best AUPR scores reaches 0.90, which is lower than those of the *NN-MT* and *RN-MT* methods. These observations may reflect the fact that the kinase family gathers proteins that are relatively diverse, which is illustrated by its

organization in several sub-families.[47]



(a) *NN-MT-family* (family kernel)

(b) *NN-MT* (sequence kernel)

(c) *RN-MT-family*(family kernel)

(d) *RN-MT* (sequence kernel)

Figure 11: AUPR score of the Multi-Task methods on the kinase family. The black horizontal line corresponds to the *MT-intra*method trained only on intra-task pairs. Numerical values can be found in Supporting Information

Overall, the above results on the IC, GPCR and kinase families indicate that the proposed *NN-MT* method leads to the best results when the train set includes all positive intra-task pairs, 10 times more random negative intra-task pairs, a small number of nearest neighbors positive extra-task pairs (in the range of 10) and around 5 times more random negative extra-task pairs. These conditions are very similar to those leading to the best prediction scores when ligand specificity is studied on large scale in the protein space. Even if the performance of *NN-MT* on family datasets is better than those reached by other methods on similar datasets,[15,38] they remain in the same order of magnitude.

4 Application to side effect prediction

One important application of the *NN-MT* method is to guide pharmaceutical companies in the choice of the best hit molecules early in the drug development process and to avoid deleterious side effects for patients.

In order to test the interest of the proposed method in this context, we considered a list of 174 drugs that were withdrawn in at least one country between 1960 and 2009[63,64] (list available in Supporting Information). Of these, 64 molecules had a DrugBank ID, among which 45 were present in our dataset S. We analyzed the target predictions made by the *NN-MT* method to determine whether they could be related to the side effects that lead to drugs withdrawal.

In three cases, the protein target responsible for the side effects were known and present in the dataset. In six cases, withdrawal was attributed to major hepatic or renal toxicity (i.e. metabolic toxicity) or severe but aspecific side effects, which are not attributable to unknown targets. We studied in more details the 36 remaining drugs, and considered only their predicted target of highest score, according to the *NN-MT* method. In one case, the best prediction was confirmed, and in eight cases, we found a direct link between the cause of withdrawal and the best predicted target. These nine examples are shortly discussed in the following.

Rimonabant (DrugBank ID DB06155) was an anti-obesity drug acting as an antagonist of the cannabinoid receptor CNR1 (Uniprot ID P21554), withdrawn due to serious psychiatric effects. Its best predicted target is acetylcholinesterase (Uniprot ID P22303). The interaction between Rimonabant and acetylcholinesterase was absent from DrugBank, but this prediction is confirmed in the DrugCentral database (DrugCentral ID 4150), showing that Rimonabant inhibits acetylcholinesterase with an IC50 value of 4.6 μM. Although in this particular case, the known target CNR1 might be, at least in part, responsible for the psychiatric side effects,[65] the fact that the best prediction was confirmed is an interesting result.

Two molecules were withdrawn because of teratogenicity risk: Diethylstilbestrol (DB00255) and Thalidomide (DB01041). This is consistent with the first predicted target DNA polymerase alpha catalytic domain POLA1 (Uniprot ID P09884) for the former, and DNA polymerase epsilon subunit 3 POLE3 (Uniprot ID Q9NRF9) for the latter.

Etretinate (DB00926) was withdrawn because of birth defect risks, while its best predicted target is steryl-sulfatase STS (Uniprot ID P08842), an enzyme that converts sulfated steroid precursors to estrogens during pregnancy.[66]

Four molecules were withdrawn for cardiac toxicity : Dextropropoxyphene (DB00647), Rosiglitazone (DB00412), Prenylamine (DB04825), and Dexfenfluramine (DB01191). The first predicted target for the two former are involved in the metabolism of NO, a key molecule for the fine tuning of cardiac function. For Dextroproxyphene, the predicted SDF2 target (Uniprot ID Q99470) plays a critical role in the activation of NO production,[67] and for Rosiglitazone, the predicted DDAH1 target (Uniprot ID O94760) has a role in NO production regulation.[66] Prenylamine was withdrawn for cardiac arrythmia while its best predicted target is the calcium-activated potassium channel KCNMB3 (Uniprot ID Q9NPA1), which plays a critical role in the the cardiac excitation-contraction coupling.[68] Dexfenfluramine was withdrawn for heart attack and stroke risks while its best predicted target is PI3K (Uniprot ID P48736), a kinase involved in many aspects of heart homeostasis.[69]

Finally, Zimelidine (DB04832) is a serotonin reuptake inhibitor that was used as antidepressant, and withdrawn because of severe central and peripheral neuropathy. The best predicted target is sodium- and chloride-dependant GABA transporter 2 SLC6A14 (Uniprot ID Q9NSD5), involved in the transport of the key neurotransmitter GABA,[70] which could be related to the observed side effects.

To summarize, for 36 withdrawn molecules, side effects could be attributable to unknown targets, and targets prediction was available in our dataset. For these 36 molecules, we considered only the best predicted target acording to the *NN-MT* method, which was a very stringent choice. The best predicted target could be confirmed for one molecule, and a direct

link between the best predicted target and the side effect responsible for drug withdrawal could be found for eight molecules. These observations do not validate the predictions. However, they illustrate that the function of the best predicted target could have helped to foresee potential side effects, and to conceive additional experiments to better evaluate the interest of these drug candidates before entering drug development. For the remaining molecules, no direct link was found between the best predicted target and the side effects that caused drug withdrawal. However, they might have been related to predicted targets of lower ranks, but we drastically restricted the number of predictions that were analyzed. These side effects might also have been indirectly related to the best predicted target, but full analysis of the corresponding pathways was out of the scope of this paper. Such restrictions in the analysis of the results would of course not be employed in real applications, which would help to better foresee potential drug candidate side effects.

5 Conclusion

The present study tackles prediction of ligand specificity, which can be formulated as a problem (protein, ligand) interaction prediction on large scale in the space of proteins.

Our goal was to propose a method to explore the specificity of a molecule with state-of-the-art or better performance over a wide range of prediction situations, at a computational cost close to that of single task approaches. We showed that the *NN-MT* method fulfills these requirements, leading to the best prediction performance for the three tested settings which cover most of the prediction situations that would be encountered in real-case studies.

To summarize the main characteristics of the proposed *NN-MT* method (detailed in Sections 3.1, 3.4 and 3.5), we suggest to predict each (protein, ligand) interaction using the Profile kernel for proteins (with subsequences length k of 5 and threshold equaled to 7.5) and the Tanimoto kernel for molecules (with length of path 8), with a train set including:

- all positive intra-task pairs (i.e. all known interactions involving the protein or the

ligand of the test pair), and around ten times more randomly chosen intra-task negative pairs.

- a small number of the closest positive extra-task pairs (i.e. a number similar to that of intra-task positive pairs), and a similar number of randomly chosen negatives extra-tasks pairs.

This should provide good defaults parameters in order to use the *NN-MT* method in a straightforward manner for users that are not familiar with machine learning approaches.

Overall, the *NN-MT* method could be an interesting tool to guide not only the choice of the best hit molecules early in the drug development process avoiding deleterious side effects for patients, but also to suggest drug repositioning opportunities since a side effect for a patient might be viewed as a therapeutic effect for another.

In order to test the performance the considered algorithms, we used the DrugBank database to build several datasets that illustrate various prediction contexts and are available for the community at `https://github.com/bplaye/efficient_MultiTask_SVM_for_chemogenomics`. Since DrugBank is regularly updated, the *NN-MT* method can be improved continuously, by re-training on larger up-dated train sets. Our datasets included all human proteins that are targeted by at least one drug-like molecule. According to the present knowledge, this set of proteins can be viewed as the druggable proteome. Therefore, these proteins are relevant to train and test computational models for drug specificity prediction.

We formalized (protein, ligand) interaction prediction as a classification problem because they can be solved at a reasonable computational cost on large datasets, as those require for prediction of (protein, ligand) interaction on large scale. Whenever the purpose would be to predict the relative affinities of molecules for a set of proteins, the predicted scores can be used to rank all interactions. However, this question could be also solved using a regression

algorithm when predicting the affinity between pair of molecule and protein.[25] Note that in such an approach, the affinity of all (protein, ligand) pairs in the training data is required, which is rarely available on large scale.

Although the protein-ligand interaction process takes place in the 3D space, we chose to encode the two partners based on features that do not require 3D information. Indeed, the bound conformation of the ligand, and the 3D structure of the protein binding pocket is unknown in most cases, which prevents predictions on large scale. However, we are aware that a method using 3D information to encode the interaction can be of interest on more restricted datasets (i.e. not covering the druggable human proteome) as those available from the PDB database.[27,28]

Since the prediction performance strongly depends on the distance between the predicted interactions and the train set, it could be relevant to apply the multi-kernel learning (MKL) framework.[71] Indeed, different feature spaces will lead to different metrics which could modulate the distance between the test and train sets. This idea was explored in Nascimento et al.[72], but in this work the MKL approach employed L2 regularization between kernels, which did not lead to the improvements that could be expected from an L1 (i.e. sparsity-inducing) regularization term. It is likewise relevant to explore the benefits of deep learning approaches in the context of representation learning.[73] Indeed, learning the featurization of molecules[74] and of proteins[75] on various prediction tasks including drug-target interaction prediction could optimize the featurization for the specificity prediction task.

Acknowledgement

This work was supported by the Mines ParisTech, and from Ministère de l'Économie.

Supporting Information Available

Scores of MT SVM chemogenomics on $S_1 - S_4$ settings obtained with the 5-fold CV scheme (Figure F1). Scores of MT Kernel Ridge regression on S1/2/34 in nested-5foldCV and Scores of MT Kernel Ridge regression on the S_1 depending on CV scheme (Figure F2). Scores of MT SVM method on the S'_1, S'_2, S'_3 and S'_4 datasets with nested-5-fold CV (Figure F3). Scores of the MT method on S_1 depending on CV scheme (Figure F4). Performance values for all displayed performance plots (Tables). List of the 174 considered withdrawn drugs. Basic principles of SVM and Kernel Ridge Regression.

References

(1) DiMasi, J. A.; Grabowski, H. G.; Hansen, R. W. *Journal of health economics* **2016**, *47*, 20–33.

(2) Miguel, A.; Azevedo, L. F.; Araújo, M.; Pereira, A. C. *Pharmacoepidemiology and drug safety* **2012**, *21*, 1139–1154.

(3) Lazarou, J.; Pomeranz, B. H.; Corey, P. N. *Jama* **1998**, *279*, 1200–1205.

(4) Onakpoya, I. J.; Heneghan, C. J.; Aronson, J. K. *BMC medicine* **2016**, *14*, 10.

(5) Scheiber, J.; Chen, B.; Milik, M.; Sukuru, S. C. K.; Bender, A.; Mikhailov, D.; Whitebread, S.; Hamon, J.; Azzaoui, K.; Urban, L.; Glick, M.; Davies, J. W.; Jenkins, J. L. *Journal of chemical information and modeling* **2009**, *49*, 308–317.

(6) Netzeva, T. et al. *ATLA* **2005**, *33*, 155–173.

(7) Tetko, I. V.; Bruneau, P.; Mewes, H.-W.; Rohrer, D. C.; Poda, G. I. *Drug discovery today* **2006**, *11*, 700–707.

(8) Tropsha, A. *Annual reports in computational chemistry* **2006**, *2*, 113–126.

(9) Shoichet, B. K.; Kuntz, I. D.; Bodian, D. L. *Journal of Computational Chemistry* **1992**, *13*, 380–397.

(10) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. *Journal of molecular biology* **1997**, *267*, 727–748.

(11) Morris, G. M.; Goodsell, D. S.; Halliday, R. S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. *Journal of computational chemistry* **1998**, *19*, 1639–1662.

(12) Vert, J.-P.; Jacob, L. *Combinatorial chemistry & high throughput screening* **2008**, *11*, 677–685.

(13) Bolton, E. E.; Wang, Y.; Thiessen, P. A.; Bryant, S. H. *Annual reports in computational chemistry* **2008**, *4*, 217–241.

(14) Yamanishi, Y.; Araki, M.; Gutteridge, A.; Honda, W.; Kanehisa, M. *Bioinformatics* **2008**, *24*, i232–i240.

(15) Jacob, L.; Vert, J.-P. *Bioinformatics* **2008**, *24*, 2149–2156.

(16) Bleakley, K.; Yamanishi, Y. *Bioinformatics* **2009**, *25*, 2397–2403.

(17) van Laarhoven, T.; Nabuurs, S. B.; Marchiori, E. *Bioinformatics* **2011**, *27*, 3036–3043.

(18) Yamanishi, Y.; Kotera, M.; Kanehisa, M.; Goto, S. *Bioinformatics* **2010**, *26*, i246–i254.

(19) Hizukuri, Y.; Sawada, R.; Yamanishi, Y. *BMC medical genomics* **2015**, *8*, 1.

(20) Takarabe, M.; Kotera, M.; Nishimura, Y.; Goto, S.; Yamanishi, Y. *Bioinformatics* **2012**, *28*, i611–i618.

(21) Yamanishi, Y. *Molecular Informatics* **2013**, *32*, 991–999.

(22) Yuan, Q.; Gao, J.; Wu, D.; Zhang, S.; Mamitsuka, H.; Zhu, S. *Bioinformatics* **2016**, *32*, i18–i27.

(23) Caruana, R. *Machine learning* **1997**, *28*, 41–75.

(24) Yu, H.; Chen, J.; Xu, X.; Li, Y.; Zhao, H.; Fang, Y.; Li, X.; Zhou, W.; Wang, W.; Wang, Y. *PloS one* **2012**, *7*, e37608.

(25) Pahikkala, T.; Airola, A.; Pietilä, S.; Shakyawar, S.; Szwajda, A.; Tang, J.; Aittokallio, T. *Briefings in bioinformatics* **2014**, bbu010.

(26) Jacob, L.; Hoffmann, B.; Stoven, V.; Vert, J.-P. *BMC bioinformatics* **2008**, *9*, 363.

(27) Paul, N.; Kellenberger, E.; Bret, G.; Müller, P.; Rognan, D. *Proteins: Structure, Function, and Bioinformatics* **2004**, *54*, 671–680.

(28) Kellenberger, E.; Foata, N.; Rognan, D. *Journal of chemical information and modeling* **2008**, *48*, 1014–1025.

(29) Hue, M.; Riffle, M.; Vert, J.-P.; Noble, W. S. *BMC bioinformatics* **2010**, *11*, 144.

(30) Cortes, C.; Vapnik, V. *Machine learning* **1995**, *20*, 273–297.

(31) Saunders, C.; Gammerman, A.; Vovk, V. **1998**,

(32) Faulon, J.-L.; Misra, M.; Martin, S.; Sale, K.; Sapra, R. *Bioinformatics* **2008**, *24*, 225–233.

(33) Napolitano, F.; Zhao, Y.; Moreira, V. M.; Tagliaferri, R.; Kere, J.; D'Amato, M.; Greco, D. *J. Cheminformatics* **2013**, *5*, 30.

(34) Xia, Z.; Wu, L.-Y.; Zhou, X.; Wong, S. T. *BMC systems biology* **2010**, *4*, S6.

(35) Jorissen, R. N.; Gilson, M. K. *Journal of chemical information and modeling* **2005**, *45*, 549–561.

(36) Geppert, H.; Horváth, T.; Gärtner, T.; Wrobel, S.; Bajorath, J. *Journal of chemical information and modeling* **2008**, *48*, 742–746.

(37) Nagamine, N.; Shirakawa, T.; Minato, Y.; Torii, K.; Kobayashi, H.; Imoto, M.; Sakakibara, Y. *PLoS Comput Biol* **2009**, *5*, e1000397.

(38) Mei, J.-P.; Kwoh, C.-K.; Yang, P.; Li, X.-L.; Zheng, J. *Bioinformatics* **2013**, *29*, 238–245.

(39) Erhan, D.; L'Heureux, P.-J.; Yue, S. Y.; Bengio, Y. *Journal of chemical information and modeling* **2006**, *46*, 626–635.

(40) Jaakkola, T.; Diekhans, M.; Haussler, D. *Journal of computational biology* **2000**, *7*, 95–114.

(41) Leslie, C. S.; Eskin, E.; Noble, W. S. The spectrum kernel: A string kernel for SVM protein classification. Pacific symposium on biocomputing. 2002; pp 566–575.

(42) Eskin, E.; Weston, J.; Noble, W. S.; Leslie, C. S. Mismatch string kernels for SVM protein classification. Advances in neural information processing systems. 2002; pp 1417–1424.

(43) Saigo, H.; Vert, J.-P.; Ueda, N.; Akutsu, T. *Bioinformatics* **2004**, *20*, 1682–1689.

(44) Kuang, R.; Ie, E.; Wang, K.; Wang, K.; Siddiqi, M.; Freund, Y.; Leslie, C. *Journal of bioinformatics and computational biology* **2005**, *3*, 527–550.

(45) Smith, T. F.; Waterman, M. S. *Journal of molecular biology* **1981**, *147*, 195–197.

(46) Okuno, Y.; Tamon, A.; Yabuuchi, H.; Niijima, S.; Minowa, Y.; Tonomura, K.; Kunimoto, R.; Feng, C. *Nucleic acids research* **2007**, *36*, D907–D912.

(47) Manning, G.; Whyte, D. B.; Martinez, R.; Hunter, T.; Sudarsanam, S. *Science* **2002**, *298*, 1912–1934.

(48) Kanehisa, M.; Araki, M.; Goto, S.; Hattori, M.; Hirakawa, M.; Itoh, M.; Katayama, T.; Kawashima, S.; Okuda, S.; Tokimatsu, T.; Yamanishi, Y. *Nucleic acids research* **2007**, *36*, D480–D484.

(49) Swamidass, S. J.; Chen, J.; Bruand, J.; Phung, P.; Ralaivola, L.; Baldi, P. *Bioinformatics* **2005**, *21*, i359–i368.

(50) Kashima, H.; Tsuda, K.; Inokuchi, A. Marginalized kernels between labeled graphs. ICML. 2003; pp 321–328.

(51) Mahé, P.; Ueda, N.; Akutsu, T.; Perret, J.-L.; Vert, J.-P. *Journal of chemical information and modeling* **2005**, *45*, 939–951.

(52) Azencott, C.-A.; Ksikes, A.; Swamidass, S. J.; Chen, J. H.; Ralaivola, L.; Baldi, P. *Journal of chemical information and modeling* **2007**, *47*, 965–974.

(53) Golub, G. H.; Heath, M.; Wahba, G. *Technometrics* **1979**, *21*, 215–223.

(54) Hanley, J. A.; McNeil, B. J. *Radiology* **1982**, *143*, 29–36.

(55) Raghavan, V.; Bollmann, P.; Jung, G. S. *ACM Transactions on Information Systems (TOIS)* **1989**, *7*, 205–229.

(56) Pedregosa, F. et al. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(57) Kanehisa, M.; Goto, S.; Sato, Y.; Furumichi, M.; Tanabe, M. *Nucleic acids research* **2011**, gkr988.

(58) Law, V. et al. *Nucleic acids research* **2013**, *42*, D1091–D1097.

(59) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. *Nucleic acids research* **2012**, *40*, D1100–D1107.

(60) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. *Advanced drug delivery reviews* **1997**, *23*, 3–25.

(61) Kramer, C.; Gedeck, P. *Journal of chemical information and modeling* **2010**, *50*, 1961–1969.

(62) Johnson, S. C. *Psychometrika* **1967**, *32*, 241–254.

(63) Fung, M.; Thornton, A.; Mybeck, K.; Wu, J. H.-h.; Hornbuckle, K.; Muniz, E. *Drug Information Journal* **2001**, *35*, 293–317.

(64) Qureshi, Z. P.; Seoane-Vazquez, E.; Rodriguez-Monguio, R.; Stevenson, K. B.; Szeinbach, S. L. *Pharmacoepidemiology and drug safety* **2011**, *20*, 772–777.

(65) Moreira, F. A. *Best Practice & Research Clinical Endocrinology & Metabolism* **2009**, *23*, 133–144.

(66) Cotton, J.; Kearney, M.; Shah, A. *Heart* **2002**, *88*, 564–566.

(67) Kazakov, A.; Hall, R.; Jagoda, P.; Bachelier, K.; Müller-Best, P.; Semenov, A.; Lammert, F.; Böhm, M.; Laufs, U. *Cardiovascular research* **2013**, *100*, 211–221.

(68) Zhang, X.-D.; Lieu, D. K.; Chiamvimonvat, N. *Heart Rhythm* **2015**, *12*, 1845–1851.

(69) Aoyagi, T.; Matsui, T. *Current pharmaceutical design* **2011**, *17*, 1818–1824.

(70) Christiansen, B.; Meinild, A.-K.; Jensen, A. A.; Braüner-Osborne, H. *Journal of Biological Chemistry* **2007**, *282*, 19331–19341.

(71) Rakotomamonjy, A.; Bach, F. R.; Canu, S.; Grandvalet, Y. *Journal of Machine Learning Research* **2008**, *9*, 2491–2521.

(72) Nascimento, A. C.; Prudêncio, R. B.; Costa, I. G. *BMC bioinformatics* **2016**, *17*, 1.

(73) Bengio, Y.; Courville, A.; Vincent, P. *IEEE transactions on pattern analysis and machine intelligence* **2013**, *35*, 1798–1828.

(74) Lusci, A.; Pollastri, G.; Baldi, P. *Journal of chemical information and modeling* **2013**, *53*, 1563–1575.

(75) Qi, Y.; Oja, M.; Weston, J.; Noble, W. S. *PloS one* **2012**, *7*, e32235.

This material is available free of charge via the Internet at `http://pubs.acs.org/`.